

Rescue API Reference Guide



GoTo



Contents

Overview of the Rescue API.....	6
API Quick Starts.....	7
Options for Authenticating with Rescue API.....	7
Using an authcode.....	8
Using the login Method.....	13
Account Information.....	21
login.....	21
Sample Code.....	22
logout.....	24
Sample Code.....	24
getAccount.....	26
Sample Code.....	27
getHierarchy.....	31
Sample Code.....	32
getHierarchy_v2.....	36
Sample Code.....	38
Account Management.....	42
createUser.....	42
Sample Code.....	43
createUsers.....	46
Sample Code.....	49
getUser.....	51
Sample Code.....	52
getUser_v2.....	55
Sample Code.....	56
getUser_v3.....	57
Sample Code.....	58
setUser.....	60
Sample Code.....	62
setUser_v2.....	65
Sample Code.....	68
setUserStatus_v7_1.....	69
Sample Code.....	70
createGroup.....	73
Sample Code.....	74
getGroup.....	76
Sample Code.....	78
getGroup_v2.....	80
Sample Code.....	82
getGroup_v3.....	83
Sample Code.....	85
setGroup.....	86
Sample Code.....	89
setGroup_v2.....	93
Sample Code.....	96
setGroup_v3.....	98
Sample Code.....	101
moveNode.....	103
Sample Code.....	104

deleteNode.....	107
Sample Code.....	107
assignChannel.....	110
Sample Code.....	111
unassignChannel.....	114
Sample Code.....	114
Supporting CRM Systems.....	117
generateEmailText.....	117
Sample Code.....	118
getLoginTicket.....	123
Sample Code.....	123
getDTCLoginTicket.....	124
Sample Code.....	125
requestAuthCode.....	126
Sample Code.....	126
requestAuthCodeSSO.....	129
Sample Code.....	129
requestPINCode.....	132
Sample Code.....	133
requestPINCode - EU (v2).....	137
Sample Code.....	138
requestPINCode_v3.....	142
Sample Code.....	144
requestLensPINCode.....	148
Sample Code.....	149
requestLensPINCode_v2.....	153
Sample Code.....	154
requestLensPINCode_v3.....	158
Sample Code.....	160
requestLensPINCode_v4.....	164
Sample Code.....	166
Handling Pre-Session and Post-Session Data (Post-to-URL).....	168
Session Management.....	170
getSession.....	171
Sample Code.....	173
getSession_v2.....	177
Sample Code.....	179
getSession_v3.....	180
Sample Code.....	182
startSession.....	186
Sample Code.....	186
holdSession.....	189
Sample Code.....	190
transferSession.....	193
Sample Code.....	194
closeSession.....	197
Sample Code.....	197
cancelAction.....	200
Sample Code.....	201
isAnyTechAvailableOnChannel.....	203
Sample Code.....	205
Reporting.....	207
getChat.....	207
Sample Code.....	208

getNote.....	211
Sample Code.....	212
getReport.....	215
Sample Code.....	217
Structure of Generated Reports.....	223
Columns of Generated Reports.....	224
getReport_v2.....	232
Sample Code.....	234
getReportArea.....	237
Sample Code.....	238
getReportArea_v2.....	241
Sample Code.....	242
getReportArea_v3.....	246
Sample Code.....	247
getReportArea_v5.....	250
Sample Code.....	251
getReportArea_v6.....	254
Sample Code.....	255
getReportArea_v7.....	259
Sample Code.....	260
getReportArea_v8.....	263
Sample Code.....	264
setReportArea.....	268
Sample Code.....	268
setReportArea_v2.....	272
Sample Code.....	273
setReportArea_v3.....	276
Sample Code.....	277
setReportArea_v5.....	281
Sample Code.....	282
setReportArea_v6.....	285
Sample Code.....	286
setReportArea_v7.....	290
Sample Code.....	291
setReportArea_v8.....	294
Sample Code.....	295
getDateFormat.....	299
Sample Code.....	299
setDateFormat.....	301
Sample Code.....	301
getDelimiter.....	303
Sample Code.....	303
setDelimiter.....	307
Sample Code.....	307
getOutput.....	311
Sample Code.....	311
setOutput.....	315
Sample Code.....	315
getReportDate.....	319
Sample Code.....	319
getReportDate_v2.....	323
Sample Code.....	323
setReportDate.....	327
Sample Code.....	327
setReportDate_v2.....	331

Sample Code.....	332
getReportTime.....	336
Sample Code.....	336
setReportTime.....	340
Sample Code.....	340
getTimeZone.....	344
Sample Code.....	345
setTimeZone.....	348
Sample Code.....	349
getReportType.....	352
Sample Code.....	353
setReportType.....	356
Sample Code.....	357

Index..... 361

Overview of the Rescue API

The Rescue API provides an interface to third parties for communicating with an application. Use this API to customize how Rescue integrates with your other support applications.

The Rescue API is a tool for retrieving information related to your Rescue account from the LogMeIn environment and based on that information, recreate most of the functionality that the Rescue Administration Center offers. This document describes the Rescue methods that you can use to interact with Rescue from your environment. For information about the functionality that these methods invoke, see the [Rescue Administration Center User Guide](#).

Why would I use Rescue API?

The primary examples of why you would do this are:

- Make the Administration Center functionality available from your own website
- CRM Integration
- Custom Reporting
- Billing Integration
- Customize the Administration Center

Who should read this document?

The pages related to the API are written for a technical audience. Since utilization of the API is custom in nature, the following must be considered:

- Support for using the API is very limited since its usage is custom in nature
- Knowledge of API, Web Service usage, and programming are required

What technology does the Rescue API support?

The Rescue API supports three methods for communication that C# and PHP programmers can use:

Communication Method	Description
HTTP GET	<p>This protocol creates a query string of the name and value pairs and then appends the query string to the URL of the script on the server that handles the request. For example, if you want to retrieve information about a technician with ID 123456, you must use the URL of the <code>getUser</code> API method and append the input parameter (<code>node</code>) at the end of the URL:</p> <pre>https://secure.logmeinrescue.com/API/getUser.aspx?node=123456</pre>
HTTP POST	<p>This protocol passes the name and value pairs in the body of the HTTP request message. For example, if you want to retrieve information about the technician with ID 123456, you must define API method you want to call (<code>getUser</code>) and then the name and value of the input parameter (<code>node</code>):</p> <pre><form method="post" action="https://secure.logmeinrescue.com/API/getUser.aspx"> <input name=node value=123456> </form></pre>

Communication Method	Description
SOAP	This protocol is used for exchanging XML messages between clients and servers. The response message can be easily parsed into a third-party system as the structure of the message is based on well-defined standards.

What samples do I get?

All PHP examples use PHP5's built-in SOAP extension.

Examples in C# are compiled for ASP.NET 2.0 compatibility, and use `HttpRequest`, which in turn uses the HTTP POST variables listed in the examples for each API method.

Where can I learn more about API programming?

There are numerous tutorials for HTML, XML, C#, and PHP programming available on the Internet that help you understand and reproduce the examples given in this document. Visit the following websites to find out more about the API programming code:

- For a C# tutorial, visit <http://www.csharp-station.com/Tutorial>
- For HTML, XML, and PHP tutorials, visit <http://www.w3schools.com>

API Quick Starts

API quick starts are simple tutorials including code snippets on how to use the Rescue API for accessing and managing a Rescue Administration Center account without logging in on the Rescue website.

Options for Authenticating with Rescue API

Rescue API lets a Rescue account holder use the Rescue Administration Center's features without logging in to an account.

Rescue API offers two methods to authenticate users:

- The `login` method is used with applications that can handle session cookies over the HTTP protocol. It imitates the log in process of the Rescue website and session cookies are used to store and send user information. An account holder who is logged in with the `login` method must log in again after 20 minutes of inactivity.
- The `requestAuthCode` method is used with applications that cannot handle session cookies over the HTTP protocol. This method generates a random string of characters (authcode) that authenticates the Rescue account holder. The account holder uses this authcode to invoke API methods without actually logging in to the Rescue account. With a valid authcode, the account holder can invoke any number of API methods. An authcode does not time out; it remains valid until the account holder requests a new one.

The Rescue API quick starts provide examples for both methods.

Using an authcode

The following sections provide examples for using the `requestAuthCode` method to authenticate an account holder with Rescue API.

How to List Sessions of a Channel

Find a channel that has an available technician and transfer a session to that technician.

Why would you do this? You can see the details of one or more of your technicians' sessions and also see technicians' current and past workload.

1. Generate an account holder authentication code (authcode).

Account holders generate authcodes by using the [requestAuthCode](#) on page 126 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/requestAuthCode.aspx?
email=some@email.com&
pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
requestAuthCode.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
```

Response: OK AUTHCODE:4ahx...80u0

2. Request the list of technician sessions on a specific channel.

Use the [getSession_v3](#) on page 180 API method to view the list.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/getSession_v3.aspx?node=1902861082
&noderef=CHANNEL&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getSession_v3.aspx">
  <input name="node" value="1902861082">
  <input name="noderef" value="CHANNEL">
  <input name="authcode" value="4ahx...80u0">
</form>
```

Response: An OK message followed by the name of columns containing session information. The actual session information is displayed as a tab-separated list after the column names.

For example: 1571322|Connecting|0||337366|John Doe|10/14/2011 10:12 AM|105|
Customer1|||en|||yes|| 1571323|Connecting|0||337366|John Doe|10/14/2011
10:13 AM|13|Customer2|||en|||yes||

How to Assign or Unassign a Technician to a Channel

You can assign a technician to another channel if the original channel is too busy.

Why would you do this? You can transfer technicians between channels to set the workload of your technicians.

1. Generate an account holder authentication code (authcode).

Account holders generate authcodes by using the [requestAuthCode](#) on page 126 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/requestAuthCode.aspx?  
email=some@email.com&  
pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
requestAuthCode.aspx">  
  <input name="email" value="some@email.com">  
  <input name="pwd" value="secretPassword">  
</form>
```

Response: OK AUTHCODE:4ahx...80u0

2. Assign the technician to a channel.

Use the [assignChannel](#) on page 110 or [unassignChannel](#) on page 114 API method to assign or unassign a technician to a channel.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/assignChannel.aspx?  
node=337366&entry=1902861082  
&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
assignChannel.aspx">  
  <input name="node" value="337366">  
  <input name="entry" value="1902861082">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```



Note: The [unassignChannel](#) on page 114 API method requires the same parameters as [assignChannel](#) on page 110.

Response: OK

The technician is assigned to another channel.

How to View Technician Activity

Report on technician activity for a specified period of time.

Why would you do this? You can ensure your technician has a balanced workload.

1. Generate an account holder authentication code (authcode).

Account holders generate authcodes by using the [requestAuthCode](#) on page 126 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/requestAuthCode.aspx?  
email=some@email.com&  
pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
requestAuthCode.aspx">  
  <input name="email" value="some@email.com">  
  <input name="pwd" value="secretPassword">  
</form>
```

Response: OK AUTHCODE:4ahx...80u0

2. Set the start and end date of the report.

Use the [setReportDate_v2](#) on page 331 API method to set the dates of the report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/setReportDate_V2.aspx?  
bdate=12/1/2011  
&edate=12/31/2011&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
setReportDate_v2.aspx">  
  <input name="bdate" value="12/1/2011">  
  <input name="edate" value="12/31/2011">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

Response: OK

3. Optionally, set the start and end time of the report. For example, you may want to see technician activity only during office hours.

Use the [setReportTime](#) on page 340 API method to set the timespan of the report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/setReportTime.aspx?btime=9:00:00  
&etime=17:00:00&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
setReportTime.aspx">  
  <input name="btime" value="9:00:00">  
  <input name="etime" value="17:00:00">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

Response: OK

4. Optionally, if you are not looking for a technician's session information, you can change the type of your report. For example, you may want to view a report on sessions that the technician missed.

Use the [setReportArea](#) on page 268 API method to change the type of your report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/setReportArea.aspx?
area=3&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea.aspx">
  <input name="area" value="3">
  <input name="authcode" value="4ahx...80u0">
</form>
```

Response: OK

5. View the report on technician activity.

Use the [getReport](#) on page 215 API method to generate a report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/getReport.aspx?
node=337366&noderef=NODE
&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReport.aspx">
  <input name="node" value="337366">
  <input name="noderef" value="NODE">
  <input name="authcode" value="4ahx...80u0">
</form>
```

Response: An OK message and a detailed list of technician activity history.

For example, OK 12/30/2011 1:35:39 PM|8/31/2010 1:38:44 PM|00:03:05|859436|
Applet On LAN|Closed by waiting customer|JDOE|192.168.1.6|John Doe|-

For information about the available session information, see [Columns of Generated Reports](#) on page 224.

How to Change a Technician's Password

Change the password of a technician.

Why would you do this? In case a technician forgets his password you can easily change that with Rescue API.

1. Generate an account holder authentication code (authcode).

Account holders generate authcodes by using the [requestAuthCode](#) on page 126 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/requestAuthCode.aspx?
email=some@email.com&
pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
requestAuthCode.aspx">
  <input name="email" value="some@email.com">
```

```
<input name="pwd" value="secretPassword">
</form>
```

Response: OK AUTHCODE:4ahx...80u0

2. Change the password of the technician.

Use the [setUser_v2](#) on page 65 API method to generate a report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/setUser.aspx?node=337366&name=John
Doe
&nick=JDoe&email=jdoe@company.com&ssoid=123456&pwd=secretPassword
&confpwd=secretPassword&oldpwd=oldPassword&status=enabled&description=Admin
&addons=true&rpataddon=false&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setUser.aspx">
  <input name="node" value="337366">
  <input name="name" value="John Doe">
  <input name="nick" value="JDoe">
  <input name="email" value="jdoe@company.com">
  <input name="ssoid" value="123456">
  <input name="pwd" value="secretPassword">
  <input name="confpwd" value="secretPassword">
  <input name="oldpwd" value="oldPassword">
  <input name="status" value="enabled">
  <input name="description" value="Admin">
  <input name="addons" value="true">
  <input name="rpataddon" value="false">
  <input name="authcode" value="4ahx...80u0">
</form>
```

Response: OK

The password of the technician has been changed from `oldPassword` to `secretPassword`.

How to Transfer a Session to an Available Channel

Find a channel that has an available technician and transfer a session to that technician.

Why would you do this? You can reduce customers' waiting time by transferring sessions in Waiting status to a channel that has an available technician.

1. Generate an account holder authentication code (authcode).

Account holders generate authcodes by using the [requestAuthCode](#) on page 126 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/requestAuthCode.aspx?
email=some@email.com&
pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
requestAuthCode.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
```

Response: OK AUTHCODE:4ahx...80u0

2. Request the list of available technicians on a channel.

Use the [isAnyTechAvailableOnChannel](#) on page 203 API method to view the list.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/isAnyTechAvailableOnChannel.aspx?
ChannelID=
1902861082&ExcludingAwayAndBusy=true&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
isAnyTechAvailableOnChannel.aspx">
  <input name="ChannelID" value="1902861082">
  <input name="ExcludingAwayAndBusy" value="true">
  <input name="authcode" value="4ahx...80u0">
</form>
```

Response: TRUE



Note: You may have to repeat this step until you find an available channel.

3. Transfer your selected session to the available channel.

Use the [transferSession](#) on page 193 API method to transfer a session.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/transferSession.aspx?
session=123456&
node=1902861082&noderef=CHANNEL&authcode=4ahx...80u0
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
transferSession.aspx">
  <input name="session" value="123456">
  <input name="node" value="1902861082">
  <input name="noderef" value="CHANNEL">
  <input name="authcode" value="4ahx...80u0">
</form>
```

Response: OK

The selected session is transferred to a channel with an available technician.

Using the logIn Method

The following sections provide examples for using the `logIn` method to authenticate an account holder with Rescue API.

How to List Sessions of a Channel

Find a channel that has an available technician and transfer a session to that technician.

Why would you do this? You can see the details of one or more of your technicians' sessions and also see technicians' current and past workload.

- 1.** Log in if you have not done that already.

Account holders log in by using the *login* on page 21 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
```

Response: OK

2. Request the list of technician sessions on a specific channel.

Use the *getSession_v3* on page 180 API method to view the list.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/getSession_v3.aspx?node=1902861082
&noderef=CHANNEL
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getSession_v3.aspx">
  <input name="node" value="1902861082">
  <input name="noderef" value="CHANNEL">
</form>
```

Response: An OK message followed by the name of columns containing session information. The actual session information is displayed as a tab-separated list after the column names.

For example: 1571322|Connecting|0||337366|John Doe|10/14/2011 10:12 AM|105|
Customer1|||en|||yes|| 1571323|Connecting|0||337366|John Doe|10/14/2011
10:13 AM|13|Customer2|||en|||yes||

Here is a complete example for checking technician availability:

- **With HTTP GET**

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
https://secure.logmeinrescue.com/API/getSession_v3.aspx?node=1902861082
&noderef=CHANNEL
```

- **With HTTP POST**

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
getSession_v3.aspx">
  <input name="node" value="1902861082">
  <input name="noderef" value="CHANNEL">
</form>
```

How to Assign or Unassign a Technician to a Channel

You can assign a technician to another channel if the original channel is too busy.

Why would you do this? You can transfer technicians between channels to set the workload of your technicians.

1. Log in if you have not done that already.

Account holders log in by using the [login](#) on page 21 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
```

Response: OK

2. Assign the technician to a channel.

Use the [assignChannel](#) on page 110 or [unassignChannel](#) on page 114 API method to assign or unassign a technician to a channel.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/assignChannel.aspx?
node=337366&entry=1902861082
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
assignChannel.aspx">
  <input name="node" value="337366">
  <input name="entry" value="1902861082">
</form>
```



Note: The [unassignChannel](#) on page 114 API method requires the same parameters as [assignChannel](#) on page 110.

Response: OK

The technician is assigned to another channel.

Here is a complete example for checking technician availability:

- With HTTP GET

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
https://secure.logmeinrescue.com/API/assignChannel.aspx?
node=337366&entry=1902861082
```

- With HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
```

```
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
assignChannel.aspx">
  <input name="node" value="337366">
  <input name="entry" value="1902861082">
</form>
```

How to View Technician Activity

Report on technician activity for a specified period of time.

Why would you do this? You can ensure your technician has a balanced workload.

1. Log in if you have not done that already.

Account holders log in by using the [login](#) on page 21 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
```

Response: OK

2. Set the start and end date of the report.

Use the [setReportDate_v2](#) on page 331 API method to set the dates of the report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/setReportDate_v2.aspx?
bdate=12/1/2011
&edate=12/31/2011
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportDate_v2.aspx">
  <input name="bdate" value="12/1/2011">
  <input name="edate" value="12/31/2011">
</form>
```

Response: OK

3. Optionally, set the daily start and end time of the report. For example, you may want to see technician activity only during office hours.

Use the [setReportTime](#) on page 340 API method to set the timespan of the report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/setReportTime.aspx?btime=9:00:00
&etime=17:00:00
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportTime.aspx">
```



```
<input name="btime" value="9:00:00">
<input name="etime" value="17:00:00">
</form>
```

Response: OK

4. Optionally, if you are not looking for a technician's session information, you can change the type of your report. For example, you may want to view a report on sessions that the technician missed.

Use the [setReportArea](#) on page 268 API method to change the type of your report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/setReportArea.aspx?area=3
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea.aspx">
  <input name="area" value="3">
</form>
```

Response: OK

5. View the report on technician activity.

Use the [getReport](#) on page 215 API method to generate a report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/getReport.aspx?
node=337366&noderef=NODE
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReport.aspx">
  <input name="node" value="337366">
  <input name="noderef" value="NODE">
</form>
```

Response: An OK message and a detailed list of technician activity history.

For example, OK 12/30/2011 1:35:39 PM|8/31/2010 1:38:44 PM|00:03:05|859436|
Applet On LAN|Closed by waiting customer|JDOE|192.168.1.6|John Doe|-

For information about the available session information, see [Columns of Generated Reports](#) on page 224.



Note:

Here is a complete example for checking technician availability:

- With HTTP GET

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
https://secure.logmeinrescue.com/API/setReportDate_v2.aspx?bdate=12/1/2011
&edate=12/31/2011
https://secure.logmeinrescue.com/API/setReportTime.aspx?btime=9:00:00
&etime=17:00:00
https://secure.logmeinrescue.com/API/setReportArea.aspx?area=3
```

```
https://secure.logmeinrescue.com/API/getReport.aspx?
node=337366&noderef=NODE
```

- **With HTTP POST**

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportDate_v2.aspx">
  <input name="bdate" value="12/1/2011">
  <input name="edate" value="12/31/2011">
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportTime.aspx">
  <input name="btime" value="9:00:00">
  <input name="etime" value="17:00:00">
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea.aspx">
  <input name="area" value="3">
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
getReport.aspx">
  <input name="node" value="337366">
  <input name="noderef" value="NODE">
</form>
```

How to Change a Technician's Password

Change the password of a technician.

Why would you do this? In case a technician forgets his password you can easily change that with Rescue API.

1. Log in if you have not done that already.

Account holders log in by using the [login](#) on page 21 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
```

Response: OK

2. Change the password of the technician.

Use the [setUser_v2](#) on page 65 API method to generate a report.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/setUser.aspx?node=337366&name=John
Doe
&nick=JDoe&email=jdoe@company.com&ssoid=123456&pwd=secretPassword
```

```
&confpwd=secretPassword&oldpwd=oldPassword&status=enabled&description=Admin
&addons=true&rpataddon=false
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setUser.aspx">
  <input name="node" value="337366">
  <input name="name" value="John Doe">
  <input name="nick" value="JDoe">
  <input name="email" value="jdoe@company.com">
  <input name="ssoid" value="123456">
  <input name="pwd" value="secretPassword">
  <input name="confpwd" value="secretPassword">
  <input name="oldpwd" value="oldPassword">
  <input name="status" value="enabled">
  <input name="description" value="Admin">
  <input name="addons" value="true">
  <input name="rpataddon" value="false">
</form>
```

Response: OK

The password of the technician has been changed from oldPassword to secretPassword.

Here is a complete example for checking technician availability:

- With HTTP GET

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
https://secure.logmeinrescue.com/API/setUser.aspx?node=337366&name=John
Doe
&nick=JDoe
&email=jdoe@company.com&ssoid=123456&pwd=secretPassword&confpwd=secretPassword
&oldpwd=oldPassword&status=enabled&description=Admin&addons=true&rpataddon=false
```

- With HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
setUser.aspx">
  <input name="node" value="337366">
  <input name="name" value="John Doe">
  <input name="nick" value="JDoe">
  <input name="email" value="jdoe@company.com">
  <input name="ssoid" value="123456">
  <input name="pwd" value="secretPassword">
  <input name="confpwd" value="secretPassword">
  <input name="oldpwd" value="oldPassword">
  <input name="status" value="enabled">
  <input name="description" value="Admin">
  <input name="addons" value="true">
  <input name="rpataddon" value="false">
</form>
```

How to Transfer a Session to an Available Channel

Find a channel that has an available technician and transfer a session to that technician.

Why would you do this? You can reduce customers' waiting time by transferring sessions in Waiting status to a channel that has an available technician.

1. Log in if you have not done that already.

Account holders log in by using the [login](#) on page 21 API method.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com&pwd=secretPassword
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
```

Response: OK

2. Request the list of available technicians on a channel.

Use the [isAnyTechAvailableOnChannel](#) on page 203 API method to view the list.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/isAnyTechAvailableOnChannel.aspx?ChannelID=1902861082&ExcludingAwayAndBusy=true
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/isAnyTechAvailableOnChannel.aspx">
  <input name="ChannelID" value="1902861082">
  <input name="ExcludingAwayAndBusy" value="true">
</form>
```

Response: TRUE



Note: You may have to repeat this step until you find an available channel.

3. Transfer your selected session to the available channel.

Use the [transferSession](#) on page 193 API method to transfer a session.

- Using HTTP GET request:

```
https://secure.logmeinrescue.com/API/transferSession.aspx?session=123456&node=1902861082&noderef=CHANNEL
```

- Using HTTP POST request:

```
<form method="post" action="https://secure.logmeinrescue.com/API/transferSession.aspx">
  <input name="session" value="123456">
  <input name="node" value="1902861082">
  <input name="noderef" value="CHANNEL">
```

```
</form>
```

Response: OK

The selected session is transferred to a channel with an available technician.

Here is a complete example for checking technician availability:

- **With HTTP GET**

```
https://secure.logmeinrescue.com/API/login.aspx?email=some@email.com
&pwd=secretPassword
https://secure.logmeinrescue.com/API/isAnyTechAvailableOnChannel.aspx?
ChannelID=
1902861082&ExcludingAwayAndBusy=true
https://secure.logmeinrescue.com/API/transferSession.aspx?session=123456&
node=1902861082&noderef=CHANNEL
```

- **With HTTP POST**

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
isAnyTechAvailableOnChannel.aspx">
  <input name="ChannelID" value="1902861082">
  <input name="ExcludingAwayAndBusy" value="true">
</form>
<form method="post" action="https://secure.logmeinrescue.com/API/
transferSession.aspx">
  <input name="session" value="123456">
  <input name="node" value="1902861082">
  <input name="noderef" value="CHANNEL">
</form>
```

Account Information

These methods provide basic information on an account and allow the user to log in and out of an account.

login

This method authenticates and logs in the user. It reproduces the login process to your Rescue account.




Note: For information about the different authentication methods, see [Options for Authenticating with Rescue API](#) on page 7.

Input Parameters

Element	Description
email	The email address of the Rescue account that you want to log in. Required.

Element	Description
pwd	The password of the corresponding Rescue account. Required.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred.
OK	The user successfully logged in.
INVALID	Login failed because at least one of the input values is incorrect.
LIMITREACHED	The <code>logIn</code> method was called more than ten times in three minutes.  Note: If you want to log in more often, use the requestAuthCode on page 126 API method. Once you are logged in, the authcode prevents you from logging out if there is a timeout.

Sample Code

The following are examples for using the `logIn` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/login.aspx?
email=some@email.com&pwd=secretPassword
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
login.aspx">
  <input name="email" value="some@email.com">
  <input name="pwd" value="secretPassword">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=login>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
```

```
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/login.aspx";
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";

    System.Net.HttpWebRequest oReq =
        (System.Net.HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
+ "?email="
+ sEmail + "&pwd=" + sPwd);
    System.Net.HttpWebResponse oResp =
        (System.Net.HttpWebResponse) oReq.GetResponse();

    string sResp = new
        System.IO.StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write(sResp); //You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API Login Test</title>
</head>
<body>

</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>

<%@ Import Namespace="System"%>
<%@ Import Namespace="System.Net"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">
```

```

</script>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Rescue API Login Example</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <%
        //create a proxy for the SOAP requests
        WebServiceClients.API proxy = new WebServiceClients.API();

        //set up variables
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";

        //Initiate an API login
        WebServiceClients.loginRet loginResult =
proxy.login(sEmail,sPwd);
        Response.Write(loginResult);

        %>
      </div>
    </form>
  </body>
</html>

```

logOut

This method logs out the user who is currently logged in. It reproduces the logout process from your Rescue account.

Input Parameters

There are no input parameters for this method.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred.
OK	The user successfully logged out.
NOTLOGGEDIN	Logout failed because the user is no longer logged in.

Sample Code

The following are examples for using the `logOut` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/logout.aspx
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/logout.aspx">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=logout>.

PHP with SOAP

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

//log out
$logouResult = $soapclient->logout();

//print the result
echo "<b>Logout full response.</b><br />"; //formatting
print_r($logouResult);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

```
<%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/logout.aspx";

    System.Net.HttpWebRequest oReq = (System.Net.HttpWebRequest)
System.Net.WebRequest.Create(sEndpoint);
System.Net.HttpWebResponse oResp =
    (System.Net.HttpWebResponse)oReq.GetResponse();

string sResp = new
    System.IO.StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write(sResp); //You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue Single Sign On Login Test</title>
</head>
<body>

</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using APIexamples.WebServiceClients;

namespace APIexamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            WebServiceClients.API proxy = new WebServiceClients.API();
            WebServiceClients.logoutRet oLogout = proxy.logout();
            Response.Write(oLogout.ToString());
        }
    }
}
```

getAccount


Retrieves the account information of the user who is currently logged in. You can see this information in the Administration Center by hovering the mouse over a user in the organization tree.


Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK AccountID:560961 Organization:Company AdminID:337365 TechID:337366
Email:jdoe@company.com
```

Element	Description
AccountID	Rescue account ID
Organization	Name of the account holder's organization
AdminID	Administrator ID of the current user
	 Note: If the current user is not an Administrator, AdminID contains 0.

Element	Description
TechID	Technician ID of the current user  Note: If the current user has no technician seat (or his technician seat is disabled), TechID contains 0.
Email	Email address of the account holder

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	The properties of the currently logged-in user's account were successfully retrieved.
NOTLOGGEDIN	Retrieving information failed because the user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getAccount` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getAccount.aspx?authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/getAccount.aspx">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getAccount>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

//define parameters
```

```

$loginparams = array (
    'sEmail' => 'some@email.com',
    'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//get account info
$accountParams = array("");
$accountResult = $soapclient->getAccount($accountParams);

//print out the hierarchy full response
print_r("<b>getAccount full response.</b><br />");
print_r($accountResult);
echo "<br /><br />"; //formatting

//show the account info
$accountID = $accountResult["iAccountID"];
$organization = $accountResult["sOrganization"];
$adminID = $accountResult["iAdminID"];
$techID = $accountResult["iTechID"];
$email = $accountResult["sEmail"];

print_r("<b>Account Information.</b>");
print_r("<table border = \"0\">");
print_r("<tr><td>" . "Account ID: </td><td>" . $accountID . "</td></tr>");
print_r("<tr><td>" . "Organization: </td><td>" . $organization . "</td></tr>");
print_r("<tr><td>" . "Admin ID: </td><td>" . $adminID . "</td></tr>");
print_r("<tr><td>" . "Tech ID: </td><td>" . $techID . "</td></tr>");
print_r("<tr><td>" . "Email: </td><td>" . $email . "</td></tr>");
print_r("</table>");

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/";
//add actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    //sAuthCode is the return value of the requestAuthCode API call
    string sAuthCode = "";

//set up the request

```

```

    HttpRequest oReq =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
    + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpResponse oResp = (HttpResponse) oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value +
"<br />");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");

    Response.Write("When issued: " + cook.TimeStamp + "<br />");
    Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
    Response.Write("Don't save: " + cook.Discard + "<br />");
    Response.Write("Comment: " + cook.Comment + "<br />");
    Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
    Response.Write("Version: RFC " + cook.Version + "<br />");

    // Show the string representation of the cookie.
    Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the account info
HttpRequest oReqAccount =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
    + "getAccount.aspx?authcode=");
oReqAccount.CookieContainer = sessioncookie;

HttpResponse oRespAccount = (HttpResponse) oReqAccount.GetResponse();
string sRespAccount = new
    StreamReader(oRespAccount.GetResponseStream()).ReadToEnd();
Response.Write("getAccount result: " + sRespAccount + "<br />"); //You can
customize
the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getAccount Test</title>
</head>

```

```
<body>
</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using APIexamples.WebServiceClients;

namespace APIexamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string sEmail = "your@email.com"; //Rescue account email
            address
            string sPwd = "secretpassword"; //Rescue account password
            //Set up empty data for getAccount response
            int iAccount;
            string sOrg;
            int iAdmin;
            int iTech;
            string sAcctEmail;

            WebServiceClients.API proxy = new WebServiceClients.API();
            proxy.CookieContainer = new CookieContainer();
            WebServiceClients.loginRet oLogin = proxy.login(sEmail, sPwd);
            Response.Write("Login: " + oLogin + "<br />");
            WebServiceClients.getAccountRet oGetAccount =
            proxy.getAccount(out iAccount,
            out sOrg, out iAdmin, out iTech, out sEmail1);
            Response.Write(oGetAccount.ToString());
            Response.Write("Account ID: " + iAccount + "<br />" +
            "Organization ID: "
            + sOrg + "<br />" + "Admin ID: " + iAdmin + "<br />" + "Technician ID: " +
            iTech + "<br />"
            + "Email: " + sAcctEmail);
        }
    }
}
```

getHierarchy

Retrieves the hierarchy of a specified Rescue account. Unlike the `getUser` and `getGroup` methods, `getHierarchy` retrieves information about any type of Rescue node.

Input Parameters

Element	Description
<code>isnode</code>	The returned set is a node or the complete hierarchy. Optional. The possible value are as follows: <ul style="list-style-type: none">• <code>true</code>• <code>false</code>
<code>node</code>	Contains the ID of the node if the value of the <code>isnode</code> parameter is <code>true</code> . Optional.
<code>authcode</code>	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK NodeID:671835 ParentID:337364 Name:Technician  
Email:jdoe@company.com Description: Status:Offline Type:Technician
```

NodeID: ID of the Rescue node
ParentID: ID of the node's parent node
Name: Name of the node
Email: If available, the email address of the node
Description: Description of the node
Status: Status of the node as listed below
Type: Type of the node as listed below

The following node statuses are available:

- Disabled
- Offline
- Online
- Busy
- Away

The following node types are available:

- Root
- Masteradministrators
- Administrators
- Technicians
- AdministratorGroup
- TechnicianGroup
- Masteradministrator
- Administrator
- Technician

- AdministratorLink
- AdministratorGroupLink
- Channel



Note: A Rescue account holder can have multiple node IDs depending on their associated roles.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the properties of the currently logged-in user's account succeeded.
NOTLOGGEDIN	Retrieving information failed because the user is no longer logged in.
INVALIDPARAM_ISNODE	The <code>isnode</code> parameter is incorrect.
INVALIDPARAM_NODE	The <code>isnode</code> parameter is set to <code>true</code> and the <code>node</code> parameter is incorrect.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.
HAS_NO_RIGHT	The specified user (node) does not have sufficient rights to retrieve the hierarchy of the Rescue account.

Sample Code

The following are examples for using the `getHierarchy` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getHierarchy.aspx?
isnode=true&node=337364
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getHierarchy.aspx">
  <input name="isnode" value="true">
  <input name="node" value="337364">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getHierarchy>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

$hierparams = array("" => "");

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//getHierarchy
$hierarchyResult = $soapclient->getHierarchy($hierparams);

//print out the hierarchy full response
echo "<b>Hierarchy full response.</b><br />";
print_r($hierarchyResult);
echo "<br /><br />"; //formatting

//hierarchy array
$hierarchy = $hierarchyResult['aHierarchy'];

//nodes of the hierarchy
$nodes = $hierarchy["HIERARCHY"];
$numberofnodes = count($nodes);

echo "<b>Hierarchy on it's own.</b><br />";
print_r($hierarchy["HIERARCHY"]);
echo "<br /><br />"; //formatting

//now format the hierarchy
print_r("<b>" . "Formatted selection of hierarchy." . "</b>");
echo "<br />"; //formatting

echo "<table border = \"0\" cellspacing = \"5\">";
for ($iNodes = 0; $iNodes < $numberofnodes; $iNodes += 1)
{
    print_r("<tr>");
    print_r("<td>" . "Name: " . $nodes[$iNodes]["sName"] . "<br /></td>");
    print_r("<td>" . "Email: <a href=\"mailto:\" . $nodes[$iNodes]["sEmail"] .
"\>"
. $nodes[$iNodes]["sEmail"] . "</a><br /></td>");
    print_r("<td>" . "Role: " . $nodes[$iNodes]["eType"]) . "</td>";
}
```

```

    print_r("</td>");
}
print_r("</table>");
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for
each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    //sAuthCode is the return value of the requestAuthCode API call
    string sAuthCode = "";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint +
    "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

```

```

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

    //get the Hierarchy info
HttpRequest oReqHierarchy =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
+ "getHierarchy.aspx?isnode=true&node=337364&authcode=");
oReqHierarchy.CookieContainer = sessioncookie;

HttpResponse oRespHierarchy =
    (HttpResponse) oReqHierarchy.GetResponse();
string sRespHierarchy = new
    StreamReader(oRespHierarchy.GetResponseStream()).ReadToEnd();
Response.Write("getHierarchy result: " + sRespHierarchy + "<br />"); //You
    can customize
the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API Login Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using APIexamples.WebServiceClients;

namespace APIexamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string sEmail = "some@email.com"; //Rescue account email
            address
            string sPwd = "secretPassword"; //Rescue account password
            int iNode = 337364; //Node ID of which to getHierarchy
            HIERARCHY[] aHierarchy; //Setup empty array for getHierarchy
            response
            DataTable dt = new DataTable(); //Setup DataTable for display

```

```

WebServiceClients.API proxy = new WebServiceClients.API();
proxy.CookieContainer = new CookieContainer();
//Login
WebServiceClients.loginRet oLogin = proxy.login(sEmail, sPwd);
Response.Write("Login: " + oLogin + "<br />");
//getHierarchy
WebServiceClients.getHierarchyRet oGetHierarchy =
proxy.getHierarchy(false,
iNode, out aHierarchy);
Response.Write(oGetHierarchy.ToString());
//Bind return data to GridView
gridView1.DataSource = aHierarchy;
gridView1.DataBind();
    }
}
}

```

getHierarchy_v2


Retrieves the hierarchy of a specified Rescue account. Unlike the `getUser` and `getGroup` methods, the `getHierarchy_v2` method retrieves information about any type of Rescue node.

Changes in Version 2 of getHierarchy

The following has been introduced in the `getHierarchy_v2` method:

- `filteroffline` has been added as new input parameter.

Input Parameters

Element	Description
<code>isnode</code>	<p>The returned set is a node or the complete hierarchy. Optional.</p> <p>The possible value are as follows:</p> <ul style="list-style-type: none"> • <code>true</code> • <code>false</code>
<code>node</code>	<p>Contains the ID of the node if the value of the <code>isnode</code> parameter is true. Optional.</p>
<code>filteroffline</code>	<p>Filters offline technicians from the hierarchy. Optional.</p> <p>The possible value are as follows:</p> <ul style="list-style-type: none"> • <code>true</code> • <code>false</code> <p> Note: This parameter does not filter offline nodes that are not technicians.</p>
<code>authcode</code>	<p>The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.</p>

Output

```
OK NodeID:337364 ParentID:337363 Name:Technician Group1 Email:  
Description:  
Status:Offline Type:TechnicianGroup
```

NodeID: ID of the Rescue node
ParentID: ID of the node's parent node
Name: Name of the node
Email: If available, the email address of the node
Description: Description of the node
Status: Status of the node as listed below
Type: Type of the node as listed below

The following node statuses are available:

- Disabled
- Offline
- Online
- Busy
- Away

The following node types are available:

- Root
- Masteradministrators
- Administrators
- Technicians
- AdministratorGroup
- TechnicianGroup
- Masteradministrator
- Administrator
- Technician
- AdministratorLink
- AdministratorGroupLink
- Channel



Note: A Rescue account holder can have multiple node IDs depending on their associated roles.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the properties of the currently logged-in user's account succeeded.
NOTLOGGEDIN	Retrieving information failed because the user is no longer logged in.
INVALIDPARAM_ISNODE	The <code>isnode</code> parameter is incorrect.
INVALIDPARAM_NODE	The <code>isnode</code> parameter is set to <code>true</code> and the <code>node</code> parameter is incorrect.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.

Displayed Return Value	Description
USER_DELETED_OR_DISABLED	The user is deleted or disabled.
HAS_NO_RIGHT	The specified user (node) does not have sufficient rights to retrieve the hierarchy of the Rescue account.

Sample Code

The following are examples for using the `getHierarchy_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getHierarchy_v2.aspx?
isnode=true&node=337364
&filteroffline=true&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getHierarchy_v2.aspx">
  <input name="isnode" value="true">
  <input name="node" value="337364">
  <input name="filteroffline" value="true">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getHierarchy_v2.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

$hierparams = array("" => "");

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
```

```

echo "<br /><br />"; //formatting

//getHierarchy
$hierarchyResult = $soapclient->getHierarchy_v2($hierparams);

//print out the hierarchy full response
echo "<b>Hierarchy full response.</b><br />";
print_r($hierarchyResult);
echo "<br /><br />"; //formatting

//hierarchy array
$hierarchy = $hierarchyResult['aHierarchy'];

//nodes of the hierarchy
$nodes = $hierarchy["HIERARCHY"];
$numberofnodes = count($nodes);

echo "<b>Hierarchy on it's own.</b><br />";
print_r($hierarchy["HIERARCHY"]);
echo "<br /><br />"; //formatting

//now format the hierarchy
print_r("<b>" . "Formatted selection of hierarchy." . "</b>");
echo "<br />"; //formatting

echo "<table border =\"0\" cellpadding = \"5\">";
for ($iNodes = 0; $iNodes < $numberofnodes; $iNodes += 1)
{
    print_r("<tr>");
    print_r("<td>" . "Name: " . $nodes[$iNodes]["sName"] . "<br /></td>");
    print_r("<td>" . "Email: <a href=\"mailto:" . $nodes[$iNodes]["sEmail"] .
        "\">"
        . $nodes[$iNodes]["sEmail"] . "</a><br /></td>");
    print_r("<td>" . "Role: " . $nodes[$iNodes]["eType"]) . "</td>");
    print_r("</td>");
}
print_r("</table>");
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for
    each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    //sAuthCode is the return value of the requestAuthCode API call
    string sAuthCode = "";

    //set up the request

```

```

    HttpWebRequest oReq =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint +
    "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");

    Response.Write("When issued: " + cook.TimeStamp + "<br />");
    Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
    Response.Write("Don't save: " + cook.Discard + "<br />");
    Response.Write("Comment: " + cook.Comment + "<br />");
    Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
    Response.Write("Version: RFC " + cook.Version + "<br />");

    // Show the string representation of the cookie.
    Response.Write("String: " + cook.ToString());
    }*/

    //add cookies to cookie container
    CookieContainer sessioncookie = oReq.CookieContainer;

    //get the Hierarchy info
    HttpWebRequest oReqHierarchy =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
    + "getHierarchy_v2.aspx?isnode=true&node=337364&authcode=");
    oReqHierarchy.CookieContainer = sessioncookie;

    HttpWebResponse oRespHierarchy =
    (HttpWebResponse) oReqHierarchy.GetResponse();
    string sRespHierarchy = new
    StreamReader(oRespHierarchy.GetResponseStream()).ReadToEnd();
    Response.Write("getHierarchy_v2 result: " + sRespHierarchy + "<br />"); //
    You can customize
    the response

    %>

    <html xmlns="http://www.w3.org/1999/xhtml" >
    <head id="Head1" runat="server">
        <title>Rescue API Login Test</title>
    </head>

```



```
<body>
</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using APIexamples.WebServiceClients;

namespace APIexamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string sEmail = "your@email.com"; //Rescue account email
            address
            string sPwd = "secretpassword"; //Rescue account password
            int iNode = 12345678; //Node ID of which to getHierarchy
            HIERARCHY[] aHierarchy; //Setup empty array for getHierarchy_v2
            response
            DataTable dt = new DataTable(); //Setup DataTable for display

            WebServiceClients.API proxy = new WebServiceClients.API();
            proxy.CookieContainer = new CookieContainer();
            //Login
            WebServiceClients.loginRet oLogin = proxy.login(sEmail, sPwd);
            Response.Write("Login: " + oLogin + "<br />");
            //getHierarchy
            WebServiceClients.getHierarchy_v2Ret ogetHierarchy_v2 =
            proxy.getHierarchy_v2(false,
            iNode, out aHierarchy);
            Response.Write(ogetHierarchy_v2.ToString());
            //Bind return data to GridView
            gridView1.DataSource = aHierarchy;
            gridView1.DataBind();
        }
    }
}
```

Account Management

With the account management methods, you can create new Rescue users and groups, and assign Rescue users to channels.



Note: The `authcode` parameter enables you to use these methods without logging in to Rescue.

createUser

Creates a new user node (technician, administrator, or master administrator) in the company hierarchy (Organization Tree). For information about how to add users in the Rescue Administration Center, see *Setting up Your Organization* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
<code>parent</code>	The ID of the parent node that must be Master Administrator, Administrator, or Technician Group. Required.
<code>authcode</code>	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
<code>ERROR</code>	An unspecified error occurred, such as timeout.
<code>OK [new person nodeid]</code>	Creating a new Rescue user with the displayed ID succeeded.
<code>NOTLOGGEDIN</code>	Creating a new Rescue user failed because the current user is no longer logged in.
<code>INVALIDPARAM_PARENT</code>	The ID of the parent group is incorrect either because it does not exist or because it is an incorrect group type. For example, a technician can only be created in a Technician Group, but not in a channel.
<code>INVALID_SECRETAUTHCODE</code>	The secret authentication code for the user is invalid.
<code>USER_DELETED_OR_DISABLED</code>	The user is deleted or disabled.

Sample Code

The following are examples for using the `createUser` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/createUser.aspx?parent=337364&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/createUser.aspx">
  <input name="parent" value="337364">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=createUser>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

$loginparams = array (
  'sEmail' => 'some@email.com',
  'sPassword' => 'secretPassword'
);

$createuserparams = array (
  'iParentID' => 337364,
  'sAuthCode' => ''
);

$loginResult = $soapclient->login($loginparams);
print_r($loginResult);

$createUserResult = $soapclient->createUser($createuserparams);

$newUser = $createUserResult['iNewNodeID'];

print_r($createUserResult . " New user node: " . $newUser);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;

namespace createUser
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "[some@email.com]";
                string sPwd = "secretpassword";
                string sAuthCode = "";

                //Authenticate Rescue credentials
                HttpWebRequest oReq =
                    (HttpWebRequest)WebRequest.Create(sEndpoint +
"login.aspx" + "?email="
+ sEmail + "&pwd=" + sPwd);

                //Create session cookie to carry login credentials
                oReq.CookieContainer = new CookieContainer();

                //Get response
                HttpWebResponse oResp =
                    (HttpWebResponse)oReq.GetResponse();

                string sResp = new
                StreamReader(oResp.GetResponseStream()).ReadToEnd();
                Response.Write(sResp + "<br />");

                //Add cookie to CookieContainer
                CookieContainer sessioncookie = oReq.CookieContainer;

                //Define ID of parent node
                string pNode = [ID of parent node];

                //Create user
                HttpWebRequest oReq1 =
                    (HttpWebRequest)WebRequest.Create(sEndpoint +
"createUser.aspx"
+ "?parent=" + pNode + "&authcode=" sAuthCode);
                oReq1.CookieContainer = sessioncookie;

                HttpWebResponse oResp1 =
                    (HttpWebResponse)oReq1.GetResponse();

                string sResp1 = new
                System.IO.StreamReader(oResp1.GetResponseStream())
                .ReadToEnd();
            }
        }
    }
}
```

```

        Response.Write(sRespl + "<br />");
    }
    catch (Exception ex)
    {
        defaultError.Text = ex.Message;
    }
}
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using API.WebServiceClients;

namespace API
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "";

                //Login
                WebServiceClients.loginRet loginResult =
                proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                int iParentID = [ID of parent node];
                int iNewNodeID;

                //Create user
                WebServiceClients.createUserRet createResult =
                proxy.createUser(iParentID,
                sAuthCode, out iNewNodeID);
                Response.Write(iNewNodeID);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}

```






createUsers

Creates a number of Rescue users in an existing group by reading and validating the details of the new users. For information about how to add individual users in the Rescue Administration Center, see *Setting up Your Organization* in the [Rescue Administration Center User Guide](#).



Important: Only Master Administrators can use this API method. The maximum number of new users that can be created largely depends on the length of the optional parameters. For best results, it is not recommended to create more than 1000 users simultaneously - in one array.

Input Parameters

Element		Description
USERDATA[]		An array containing the parameters of the new users. These parameters are described in details below.  Note: You cannot add external technicians.
	uniqueid	The temporary ID of the new user. Required.  Note: Rescue automatically assigns a permanent node ID to each new user after the userdata array is validated.
	parentid	The ID of the parent group. Required. Depending on what type of users you want to create, you must define the ID of their parent group: <ul style="list-style-type: none">• To create Master Administrators, use the ID of the root-level category Master Administrators• To create Administrators, use the ID of an Administrator Group or the root-level category Administrators• To create Technicians, use the ID of a Technician Group. You cannot create users in the root-level category Technicians.  Note: If you want to add users to a new user group, first you must create that group as described in createGroup on page 73.
	name	The name of the user. Required.  Note: For special characters, use URL encoding.
	nick	The nickname of the user. Optional.
	email	The email address of the user. Required.  Note: You cannot use the email address of an existing Administrator or Master Administrator to create a Technician.
	ssoid	The Single Sign On ID of the user. Optional.
	pwd	The password of the user. Required.

Element		Description
	confpwd	The confirmation password. Required.
	status	The status of the user. Optional. The possible values are as follows: <ul style="list-style-type: none"> • Enabled (default, if not specifically defined) • Disabled
	description	The description of the user. Optional.
	addons	Specifies whether the user has mobile addon. Required. The possible values are as follows: <ul style="list-style-type: none"> • True • False
authcode		The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

The return values contain the result of the user creation process itself in the `state` field and the details of creating the individual users in the other fields.

A user is created in Rescue if the creation process succeeds and a `nodeID` is created for the new user. If there is an error in the `userStatus` field, then the user is still created and you should update the details of the user to ensure that it is properly set up.

Displayed Return Value		Description
state		The status of processing the new users. Possible values are described below.
	OK	Adding all users in the provided list succeeded.
	PARTIALLY_PROCESSED	Not all users were successfully created.
	ERROR	No users were created due to an unspecified error, such as timeout. The details of failure are listed in the <code>userState</code> field for the individual users.
	NOTLOGGEDIN	Adding users failed because the current user is no longer logged in.
	INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
	USER_DELETED_OR_DISABLED	The administrator who tried to add users is deleted or disabled.
	UNIQUE_ID_NOT_UNIQUE	At least one of the specified user IDs (<code>uniqueid</code>) is already taken.
	INVALIDPARAM_EMAIL_ALREADY_IN_USE	At least one of the specified email addresses is not unique.

Displayed Return Value		Description
	SSO_ID_ALREADY_EXISTS	The specified SSOID is already assigned to another newly created user.
	UNKNOWN_ERROR	Some or all users were not created due to an unspecified error.

The following information is returned about the individual users:

Displayed Return Value		Description
nodeId		The ID of the new user. If creating the user failed, then its <code>nodeId</code> is 0.
uniqueId		The temporary ID of the new user that was used during the import process.
userState		The status of processing a specific user. Possible values are described below.
	OK	Adding the user succeeded.
	ERROR	The user was not created due to an unspecified error, such as timeout.
	INVALIDPARAM_PARENTID	The ID of the parent group is incorrect either because it does not exist or because it is an incorrect group type. For example, a technician can only be created in a Technician Group, but not in a channel.
	ACCESS_DENIED	You do not have permission to create users in the specified group.
	INVALIDPARAM_PARENTID_IS_ZERO_OR_LESS	The specified parent group ID is zero or a negative number.
	INVALIDPARAM_NAME	The name is more than 128 characters long or the field is empty.
	INVALIDPARAM_NICK	The name is more than 128 characters long.
	INVALIDPARAM_DESCRIPTION	The description is more than 1024 characters long.
	INVALIDPARAM_EMAIL	The email address of the user has an incorrect format or it is more than 128 characters long.
	INVALIDPARAM_EMAIL_ALREADY_IN_USE	The specified email address is already in use.
	INVALIDPARAM_PASSWORD_CANNOT_CONTAIN_WHITESPACE_CHARACTERS	One of the password fields contains an extra space.
	INVALIDPARAM_PASSWORD_HAS_TO_BE_AT_LEAST_8_CHARACTERS_LONG	The new password must be at least 8 characters in length.

Displayed Return Value	Description
INVALIDPARAM_PASSWORD_HAS_TO_CONTAIN_AT_LEAST_TWO_UPPERCASE_LETTERS_OR_SPECIAL_CHARACTERS	The new password must contain at least 2 characters that are either uppercase or special characters.
INVALIDPARAM_PASSWORD_DOES_NOT_MEET_THE_MINIMUM_PASSWORD_STRENGTH_REQUIREMENTS	The Password Strength meter must meet the pre-defined minimum requirements.
INVALIDPARAM_NEW_PASSWORDS_DO_NOT_MATCH	The new and the confirmation passwords do not match.
INVALIDPARAM_CURRENT_PASSWORD_IS_WRONG	The current password of the user is incorrect.
INVALIDPARAM_MISSING_CONFIRMATION_PASSWORD	The confirmation password field cannot be empty.
NOT_ENOUGH_SEAT_LICENSES_AVAILABLE	There are not enough standard or mobile licenses available.
NODE_CANNOT_BE_ENABLED_BECAUSE_ITS_PARENT_NODE_IS_DISABLED	The status of the user cannot be set to enabled because the parent node is disabled.
USER_DELETED_OR_DISABLED	The administrator who tried to add users is deleted or disabled.
CREATED_BUT_NOTIFICATION_EMAIL_NOT_SENT	Notification email was not sent to the Administrator who created the users.
SSO_ID_ALREADY_EXISTS	The specified SSOID is already taken by an existing Rescue user.
INVALIDPARAM_UNIQUEID_IS_ZERO_OR_LESS	The specified user ID is zero or a negative number.
INVALIDPARAM_PASSWORD_MISSING	One of the passwords is not defined.
INVALIDPARAM_EMAIL_MISSING	The email address of the user is not defined.

Sample Code

The following are examples for using the `createUsers` method that you can call in your environment.

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=createUsers>.

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using createUsers.APIServiceReference;

namespace Rescue.API
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                APIServiceReference.APISoapClient proxy = new APIServiceReference.
                APISoapClient("APISoap12");
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";

                string sAuthCode = string.Empty;

                APIServiceReference.requestAuthCodeRet authCodeRet =
                proxy.requestAuthCode(out
                sAuthCode, sEmail, sPwd);

                if (authCodeRet !=
                APIServiceReference.requestAuthCodeRet.requestAuthCode_OK)
                {
                    Response.Write("Auth code missing: " + authCodeRet + "<br />");
                    return;
                }

                int iParent = [ID of the parent node];

                int iUIId = 0;

                APIServiceReference.USERDATA userDataFirst = new
                APIServiceReference.USERDATA
                {
                    iUniqueId = ++iUIId,
                    iParentID = iParent,
                    sName = "John Doe",
                    sNick = "JDoe",
                    sEmail = "jdoe@company.com",
                    sSSOID = "123456",
                    sPwd = "secretPassword",
                    sConfPwd = "secretPassword",
                    eStatus = APIServiceReference.nodeStatus.Enabled,
                    sDescription = "Admin",
                    eAddOns = APIServiceReference.hasMobileAddon.False
                };
            }
        }
    }
}
```

```

        APIServiceReference.USERDATA userDataSecond = new
        APIServiceReference.USERDATA
        {
            iUniqueId = ++iUIId,
            iParentID = iParent,
            sName = "Jane Doe",
            sNick = "JaneDoe",
            sEmail = "janedoe@company.com",
            sSSOID = "654321",
            sPwd = "secretPassword",
            sConfPwd = "secretPassword",
            eStatus = APIServiceReference.nodeStatus.Enabled,
            sDescription = "Admin",
            eAddOns = APIServiceReference.hasMobileAddon.False
        };

        APIServiceReference.CREATEUSERSINFO ret = sAPI.createUsers(new[]
        { userDataFirst,
        userDataSecond }, sAuthCode);

        Response.Write(ret.state.ToString());

        if (ret.createUserInfos.Length > 0)
        {
            Response.Write(ret.createUserInfos[0].userState + "<br />" +
            ret.createUserInfos[0].iNodeId);
        }
        catch (Exception ex)
        {
            lblError.Text = ex.Message;
        }
    }
}
}
}

```

getUser

Retrieves information about a specific user. You can see this information in the Rescue Administration Center by selecting a user in the organization tree.

Input Parameters

Element	Description
node	The ID of the user whose information you want to retrieve. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK NODEID: 337366 NAME: John Doe NICK: EMAIL: jdoe@company.com SSOID:  
DESCRIPTION:  
TYPE: Technician HASMOBILEADDON: False ISACCONTHOLDER: True STATUS:  
Online
```

NODEID: ID of the user whose information you want to retrieve
NAME: Username
NICK: Nickname of the user
EMAIL: User's email
SSOID: Single Sign On ID of the user
DESCRIPTION: User's description
TYPE: Type of the user, which is either Technician, Administrator, or
MasterAdministrator
HASMOBILEADDON: Displays whether the user has mobile addon
ISACCONTHOLDER: Displays whether the user is an account holder
STATUS: Status of the user

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving information about a user succeeded.
NOTLOGGEDIN	Retrieving information about a user failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing user.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getUser` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getUser.aspx?  
node=337366&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
getUser.aspx">  
  <input name="node" value="337366">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getUser>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
    $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
    api.asmx?wsdl");

    $loginparams = array (
        'sEmail' => 'some@email.com',
        'sPassword' => 'secretPassword'
    );

        $iNodeID = 337366;

    $getuserparams = array (
        'iNodeID' => $iNodeID,
        'sAuthCode' => '4ahx...80u0'
    );

    $loginResult = $soapclient->login($loginparams);

    print_r($loginResult);

    $getUserResult = $soapclient->getUser($getuserparams);

    print_r($getUserResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;

namespace getUser
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string sEndpoint = "https://secure.logmeinrescue.com/api/";
            string sEmail = "some@email.com";
            string sPwd = "secretPassword";
            int iNode = 337366;
            string sAuthCode = "4ahx...80u0";

            HttpWebRequest oLogin =
            (HttpWebRequest)WebRequest.Create(sEndpoint
```

```

+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
    oLogin.CookieContainer = new CookieContainer();

    HttpWebResponse oLoginResp = (HttpWebResponse)oLogin.GetResponse();
    string sLoginResp = new
StreamReader(oLoginResp.GetResponseStream()).ReadToEnd();
    Response.Write(sLoginResp);

    CookieContainer sessioncookie = oLogin.CookieContainer;

    HttpWebRequest oReq = (HttpWebRequest)WebRequest.Create(sEndpoint
+ "getUser.aspx" + "?node=" + iNode + "&authcode=" + sAuthCode);
    oReq.CookieContainer = sessioncookie;

    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();
    string sResp = new
StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write(sResp);
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using API.WebServiceClients;

namespace getUser
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";

                WebServiceClients.loginRet loginResult =
proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                int iNodeID = [ID of node in which information is being
retrieved];
                User oUser = new User();

                WebServiceClients.getUserRet getResult =
proxy.getUser(iNodeID, sAuthCode,
out oUser);
            }
            catch { }
        }
    }
}

```

```

        Response.Write(getResult + oUser.sName + oUser.sEmail
[etc...]);
    }
    catch (Exception ex)
    {
        lblError.Text = ex.Message;
    }
}
}
}

```

getUser_v2

Retrieves information about the specified user. You can see this information in the Rescue Administration Center by selecting a user in the organization tree.

Changes in Version 2 of getUser

The following has been introduced in the `getUser_v2` method:

- HASRPATADDON has been added as new output parameter.

Input Parameters

Element	Description
node	The ID of the user whose information you want to retrieve. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```

OK NODEID: 337366 NAME: John Doe NICK: EMAIL: jdoe@company.com SSOID:
DESCRIPTION:
TYPE: Technician HASMOBILEADDON: False HASRPATADDON: True ISACCOUNTHOLDER:
True STATUS: Online

```

```

NODEID: ID of the user whose information you want to retrieve
NAME: Username
NICK: Nickname of the user
EMAIL: User's email
SSOID: Single Sign On ID of the user
DESCRIPTION: User's description
TYPE: Type of the user, which is either Technician, Administrator, or
MasterAdministrator
HASMOBILEADDON: Displays whether the user has mobile addon
HASRPATADDON: Displays whether the user has RPAT addon
ISACCOUNTHOLDER: Displays whether the user is an account holder
STATUS: Status of the user

```

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving information about a user succeeded.
NOTLOGGEDIN	Retrieving information about a user failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing user.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getUser_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getUser_v2.aspx?
node=337366&authcode=4ahx...80u0
```


HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getUser_v2.aspx">
  <input name="node" value="337366">
  <input name="authcode" value="4ahx...80u0">
</form>
```

C# with SOAP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
```


Element	Description
email	The email address of the user whose details you want to retrieve. Required.  Important: You must use either the <code>node</code> or the <code>email</code> parameter to retrieve user information.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK NODEID: 337366 NAME: John Doe NICK: EMAIL: jdoe@company.com SSOID:
DESCRIPTION:
TYPE: Technician HASMOBILEADDON: False HASRPATADDON: True ISACCOUNTHOLDER:
True STATUS: Online
```

```
NODEID: ID of the user whose information you want to retrieve
NAME: Username
NICK: Nickname of the user
EMAIL: User's email
SSOID: Single Sign On ID of the user
DESCRIPTION: User's description
TYPE: Type of the user, which is either Technician, Administrator, or
MasterAdministrator
HASMOBILEADDON: Displays whether the user has mobile addon
HASRPATADDON: Displays whether the user has RPAT addon
ISACCOUNTHOLDER: Displays whether the user is an account holder
STATUS: Status of the user
```

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving information about a user succeeded.
NOTLOGGEDIN	Retrieving information about a user failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing user.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getUser_v3` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getUser_v3.aspx?email=jdoe@company.com
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getUser_v3.aspx">
  <input name="email" value="jdoe@company.com">
  <input name="authcode" value="4ahx...80u0">
</form>
```

C# with SOAP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";

                //Login
                WebServiceClients.loginRet loginResult =
                proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                int iNodeID = [ID of node in which information is being
                retrieved];
                User_V2 oUserV2 = new User_V2();

                //Get user data
                WebServiceClients.getUserRet_v2 createResult =
                proxy.getUser_v3(iNodeID, sAuthCode, out oUserV2);
                Response.Write(createResult + oUserV2.sName +
                oUserV2.sEmail [etc...]);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}
```

```
}  
  
}
```

setUser

Updates the properties of a user in the company hierarchy. In the Rescue Administration Center, select a user to modify its properties. For information about how to modify user details in the Rescue Administration Center, see *Setting up Your Organization* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
node	The ID of the parent node. Required.
name	The name of the user. Required.
nick	The nick of the user. Optional.
email	The email address of the user. Required.
ssoid	The Single Sign On ID of the user. Optional.
pwd	The password of the user. Optional.
confpwd	The confirmation password. Optional.
oldpwd	The old password of the user. Optional.
status	The status of the user. Required. The possible values are as follows: <ul style="list-style-type: none">• Enabled• Disabled
description	The description of the user. Optional.
addons	Specifies whether the user has mobile addon. Required. The possible values are as follows: <ul style="list-style-type: none">• True• False
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.

Displayed Return Value	Description
OK	Updating user properties succeeded.
NOTLOGGEDIN	Updating user properties failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is either not the ID of a user node or does not exist.
INVALIDPARAM_NAME	The name is too long or the field is empty.
INVALIDPARAM_STATUS	The specified status is incorrect or does not exist.
INVALIDPARAM_MOBILEADDON	The mobile addon of the user is incorrectly specified.
INVALIDPARAM_EMAIL	The email address of the use has an incorrect format.
INVALIDPARAM_EMAIL_ALREADY_IN_USE	The specified email address is already in use.
INVALIDPARAM_MISSING_CURRENT_PASSWORD	The current password is not defined. This error only occurs when Master administrators update their own user properties.
INVALIDPARAM_PASSWORD_CANNOT_CONTAIN_WHITESPACE_CHARACTERS	One of the password fields contain an extra space.
INVALIDPARAM_PASSWORD_HAS_TO_BE_AT_LEAST_8_CHARACTERS_LONG	The new password must be at least eight characters in length.
INVALIDPARAM_PASSWORD_HAS_TO_CONTAIN_AT_LEAST_TWO_UPPERCASE_LETTERS_OR_SPECIAL_CHARACTERS	The new password must contain at least two characters that are either uppercase or special characters.
INVALIDPARAM_PASSWORD_LENGTH_HAS_TO_BE_BIGGER_THAN_ZERO	The password fields cannot be empty.
INVALIDPARAM_PASSWORD_DOES_NOT_MEET_THE_MINIMUM_PASSWORD_STRENGTH_REQUIREMENTS	The Password Strength meter must meet the pre-defined minimum requirements.
INVALIDPARAM_NEW_PASSWORDS_DO_NOT_MATCH	The new and the confirmation passwords do not match.
INVALIDPARAM_CURRENT_AND_NEW_PASSWORD_MATCH	The current and the new passwords cannot match.
INVALIDPARAM_CURRENT_PASSWORD_IS_WRONG	The current password of the user is incorrect.
INVALIDPARAM_MISSING_CONFIRMATION_PASSWORD	The confirmation password field cannot be empty.
NOT_ENOUGH_SEAT_LICENSES_AVAILABLE	There are not enough standard or mobile licenses available.
NODE_CANNOT_BE_ENABLED_BECAUSE_ITS_PARENT_NODE_IS_DISABLED	The status of the user cannot be set to enabled because the parent node is disabled.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.

Displayed Return Value	Description
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setUser` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setUser.aspx?node=337366&name=John
Doe&nick=JDoe
&email=jdoe@company.com&ssoid=123456&pwd=secretPassword&confpwd=secretPassword
&oldpwd=oldPassword&status=enabled&description=Admin&addons=true&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setUser.aspx">
  <input name="node" value="337366">
  <input name="name" value="John Doe">
  <input name="nick" value="JDoe">
  <input name="email" value="jdoe@company.com">
  <input name="ssoid" value="123456">
  <input name="pwd" value="secretPassword">
  <input name="confpwd" value="secretPassword">
  <input name="oldpwd" value="oldPassword">
  <input name="status" value="enabled">
  <input name="description" value="Admin">
  <input name="addons" value="true">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setUser>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

$loginparams = array (
  'sEmail' => 'some@email.com',
  'sPassword' => 'secretPassword'
);

$iNodeID = 123456;
$sName = "John Doe";
$sNick = "JDoe";
$sEmail = "jdoe@company.com";
$sSSOID = "123456";
```

```

$sPwd = "secretPassword";
$sConfPwd = "secretPassword";
$sOldPwd = "oldPassword";
$eStatus = Enabled;
$sDescription = "Admin";
$eAddons = true;
$sAuthCode = "4ahx...80u0";

$setuserparams = array (
    'iNodeID' => $iNodeID,
    'sName' => $sName,
    'sNick' => $sNick,
    'sEmail' => $sEmail,
    'sSSOID' => $sSSOID,
    'sPwd' => $sPwd,
    'sConfPwd' => $sConfPwd,
    'sOldPwd' => $sOldPwd,
    'eStatus' => $eStatus,
    'sDescription' => $sDescription,
    'eAddons' => $eAddons,
    'sAuthCode' => $sAuthCode
);

$loginResult = $soapclient->login($loginparams);

print_r($loginResult);

$setUserResult = $soapclient->setUser($setuserparams);

print_r($setUserResult);
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;

namespace setUser
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sNode = "ID of node being updated";
                string sName = "Name of user";
                string sNick = "Nick name of user";
                string sTechEmail = "Email address of user";
                string sSSOID = "Single Sign On ID of user";
                string sTechPwd = "New password";
            }
        }
    }
}

```

```

        string sConfPwd = "Password confirmation";
        string sOldPwd = "Old password";
        string sStatus = "Status of node (Enabled | Disabled)";
        string sDescr = "Description of user";
        string sAddons = "Has mobile addons (True | False)";
        string sAuthCode = "";

        HttpRequest oReqLogin =
        (HttpRequest)WebRequest.Create(sEndpoint +
        "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
        oReqLogin.CookieContainer = new CookieContainer();

        HttpResponse oRespLogin =
        (HttpResponse)oReqLogin.GetResponse();
        string sRespLogin = new
        StreamReader(oRespLogin.GetResponseStream()).ReadToEnd();
        Response.Write(sRespLogin + "<br />");

        CookieContainer sessioncookie = oReqLogin.CookieContainer;

        HttpRequest oReqSetUser =
        (HttpRequest)WebRequest.Create(sEndpoint +
        "setuser.aspx" + "?node=" + sNode + "&name=" + sName + "&nick=" + sNick +
        "&email="
        + sTechEmail + "&ssoid=" + sSSOID + "&pwd=" + sTechPwd + "&confpwd=" +
        sConfPwd + "&oldpwd="
        + sOldPwd + "&status=" + sStatus + "&description=" + sDescr + "&addons=" +
        sAddons
        + "&authcode=" + sAuthCode);
        oReqSetUser.CookieContainer = sessioncookie;

        HttpResponse oRespSetUser =
        (HttpResponse)oReqSetUser.GetResponse();
        string sRespSetUser = new
        StreamReader(oRespSetUser.GetResponseStream())
        .ReadToEnd();
        Response.Write(sRespSetUser + "<br />");
    }
    catch (Exception ex)
    {
        lblError.Text = ex.Message;
    }
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using setUser.WebServiceClients;

namespace setUser
{

```



```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            string sEndpoint = "https://secure.logmeinrescue.com/api/";
            string sEmail = "some@email.com";
            string sPwd = "secretPassword";
            int iNode = Node ID of user being modified;
            string sName = "Name of user";
            string sNick = "Nick name of user";
            string sTechEmail = "Email address of user";
            string sSSOID = "Single Sign On ID of user";
            string sTechPwd = "New password";
            string sConfPwd = "Password confirmation";
            string sOldPwd = "Old password";
            string sDescr = "Description of user";
            string sAuthCode = "";

            WebServiceClients.API proxy = new WebServiceClients.API();

            WebServiceClients.loginRet loginResult =
proxy.login(sEmail, sPwd);
            Response.Write(loginResult + "<br />");

            WebServiceClients.setUserRet setUserReturn =
proxy.setUser(iNode, sName,
sNick, sTechEmail, sSSOID, sTechPwd, sConfPwd, sOldPwd,
nodeStatus.Enabled, sDescr,
hasMobileAddon.True, sAuthCode);
            Response.Write(setUserReturn + "<br />");
        }
        catch (Exception ex)
        {
            lblError.Text = ex.Message;
        }
    }
}
}

```

setUser_v2

Updates the properties of a user in the company hierarchy. In the Rescue Administration Center, select a user to modify its properties. For information about how to modify user details in the Rescue Administration Center, see *Setting up Your Organization* in the [Rescue Administration Center User Guide](#).

Changes in Version 2 of setUser

The following have been introduced in the `setUser_v2` method:

- `rpataddon` has been added as new input parameter.
- `INVALIDPARAM_RPATADDON`, `INVALID_SECRETAUTHCODE`, and `USER_DELETED_OR_DISABLED` have been added as new return values.

Input Parameters

Element	Description
node	The ID of the parent user. Required.
name	The name of the user. Required.
nick	The nick of the user. Optional.
email	The email address of the user. Required.
ssoid	The Single Sign On ID of the user. Optional.
pwd	The password of the user. Optional.
confpwd	The confirmation password. Optional.
oldpwd	The old password of the user. Optional.
status	The status of the user. Required. The possible values are as follows: <ul style="list-style-type: none">• Enabled• Disabled
description	The description of the user. Optional.
mobileaddon	Specifies whether the user has mobile addon. Required. The possible values are as follows: <ul style="list-style-type: none">• True• False
rpataddon	Specifies whether the user has RPAT addon. Required. The possible values are as follows: <ul style="list-style-type: none">• True• False
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Updating user properties succeeded.
NOTLOGGEDIN	Updating user properties failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is either not the ID of a user node or does not exist.
INVALIDPARAM_NAME	The name is too long or the field is empty.

Displayed Return Value	Description
INVALIDPARAM_STATUS	The specified status is incorrect or does not exist.
INVALIDPARAM_MOBILEADDON	The mobile addon of the user is incorrectly specified.
INVALIDPARAM_EMAIL	The email address of the use has an incorrect format.
INVALIDPARAM_EMAIL_ALREADY_IN_USE	The specified email address is already in use.
INVALIDPARAM_MISSING_CURRENT_PASSWORD	The current password is not defined. This error only occurs if a Master administrator updates his own user properties.
INVALIDPARAM_PASSWORD_CANNOT_CONTAIN_WHITESPACE_CHARACTERS	One of the password fields contain an extra space.
INVALIDPARAM_PASSWORD_HAS_TO_BE_AT_LEAST_8_CHARACTERS_LONG	The new password must be at least 8 characters in length.
INVALIDPARAM_PASSWORD_HAS_TO_CONTAIN_AT_LEAST_TWO_UPPERCASE_LETTERS_OR_SPECIAL_CHARACTERS	The new password must contain at least 2 characters that are either uppercase or special characters.
INVALIDPARAM_PASSWORD_LENGTH_HAS_TO_BE_BIGGER_THAN_ZERO	The password fields cannot be empty.
INVALIDPARAM_PASSWORD_DOES_NOT_MEET_THE_MINIMUM_PASSWORD_STRENGHT_REQUIREMENTS	The Password Strength meter must meet the pre-defined minimum requirements.
INVALIDPARAM_NEW_PASSWORDS_DO_NOT_MATCH	The new and the confirmation passwords do not match.
INVALIDPARAM_CURRENT_AND_NEW_PASSWORD_MATCH	The current and the new passwords cannot match.
INVALIDPARAM_CURRENT_PASSWORD_IS_WRONG	The current password of the user is incorrect.
INVALIDPARAM_MISSING_CONFIRMATION_PASSWORD	The confirmation password field cannot be empty.
NOT_ENOUGH_SEAT_LICENSES_AVAILABLE	There are not enough standard, mobile, or RPAT licenses available.
NODE_CANNOT_BE_ENABLED_BECAUSE_ITS_PARENT_NODE_IS_DISABLED	The status of the user cannot be set to enabled because the parent node is disabled.
INVALIDPARAM_RPATADDON	The RPAT addon of the user is incorrectly specified.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setUser_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setUser_v2.aspx?node=337366&name=John
Doe&nick=JDoe
&email=jdoe@company.com&ssoid=123456&pwd=secretPassword&confpwd=secretPassword
&oldpwd=oldPassword&status=enabled&description=Admin&addons=true&rpataddon=false
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setUser_v2.aspx">
  <input name="node" value="337366">
  <input name="name" value="John Doe">
  <input name="nick" value="JDoe">
  <input name="email" value="jdoe@company.com">
  <input name="ssoid" value="123456">
  <input name="pwd" value="secretPassword">
  <input name="confpwd" value="secretPassword">
  <input name="oldpwd" value="oldPassword">
  <input name="status" value="enabled">
  <input name="description" value="Admin">
  <input name="addons" value="true">
  <input name="rpataddon" value="false">
  <input name="authcode" value="4ahx...80u0">
</form>
```

C# with SOAP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iNode = [Node ID of user being modified];
            }
        }
    }
}
```


Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred.
OK	The status of the Rescue user was successfully changed.
NOTLOGGEDIN	Changing the Rescue user's status failed because the current user is no longer logged in.
INVALIDPARAM_STATUS	The specified status does not exist.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing Rescue user.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user who is trying to change the status of another Rescue user is deleted or disabled.

Sample Code

The following are examples for using the `setUserStatus_v7_1` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setUserStatus_v7_1.aspx?
node=337366&status=disabled
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setUserStatus_v7_1.aspx">
  <input name="node" value="337366">
  <input name="status" value="disabled">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=setUserStatus_v7_1.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
    $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
    api.asmx?wsdl");

    $loginparams = array (
        'sEmail' => 'some@email.com',
        'sPassword' => 'secretPassword'
    );

    $iNodeID = 123456;
    $eStatus = Enabled;
    $sAuthCode = "4ahx...80u0";

    $setuserparams = array (
        'iNodeID' => $iNodeID,
        'eStatus' => $eStatus,
        'sAuthCode' => $sAuthCode
    );

    $loginResult = $soapclient->login($loginparams);

    print_r($loginResult);

    $setUserResult = $soapclient->setUserStatus_v7_1($setuserparams);

    print_r($setUserResult);

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;

namespace setUser
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sNode = "ID of node being updated";
                string sStatus = "Status of node (Enabled | Disabled)";
                string sAuthCode = "";

                HttpWebRequest oReqLogin =
                (HttpWebRequest) WebRequest.Create(sEndpoint +
```

```

"login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
oReqLogin.CookieContainer = new CookieContainer();

    HttpResponseMessage oRespLogin =
(HttpWebResponse) oReqLogin.GetResponse();
    string sRespLogin = new
StreamReader(oRespLogin.GetResponseStream()).ReadToEnd();
    Response.Write(sRespLogin + "<br />");

    CookieContainer sessioncookie = oReqLogin.CookieContainer;

    HttpWebRequest oReqSetUser =
(HttpWebRequest) WebRequest.Create(sEndpoint +
"setuserstatus_v7_1.aspx" + "?node=" + sNode + "&status=" + sStatus +
"&authcode=" + sAuthCode);
    oReqSetUser.CookieContainer = sessioncookie;

    HttpResponseMessage oRespSetUser =
(HttpWebResponse) oReqSetUser.GetResponse();
    string sRespSetUser = new
StreamReader(oRespSetUser.GetResponseStream())
.ReadToEnd();
    Response.Write(sRespSetUser + "<br />");
    }
    catch (Exception ex)
    {
        lblError.Text = ex.Message;
    }
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using setUser.WebServiceClients;

namespace setUser
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iNode = Node ID of user being modified;
                string sAuthCode = "";

                WebServiceClients.API proxy = new WebServiceClients.API();
            }
        }
    }
}

```



```

        WebserviceClients.loginRet loginResult =
proxy.login(sEmail, sPwd);
        Response.Write(loginResult + "<br />");

        WebserviceClients.setUserRet setUserReturn =
proxy.setUserStatus_v7_1(iNode,
nodeStatus.Enabled, sAuthCode);
        Response.Write(setUserReturn + "<br />");
    }
    catch (Exception ex)
    {
        lblError.Text = ex.Message;
    }
}
}
}

```

createGroup

Creates a new user group in the company hierarchy (Organization Tree). For information about how to add user groups in the Rescue Administration Center, see [Setting up Your Organization](#) in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
parent	The ID of the parent node, which can be either the Administrators or Technicians organizational group in the Administration Center. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK [new group nodeid]	Creating a new Rescue user group with the displayed ID succeeded.
NOTLOGGEDIN	Creating a new Rescue user group failed because the current user is no longer logged in.
INVALIDPARAM_PARENT	The ID of the parent group is incorrect either because it does not exist or because it is an incorrect group type. For example, a Technician Group can only be created in another Technician Group, but not in an Administrator Group.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.

Displayed Return Value	Description
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `createGroup` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/createGroup.aspx?
parent=337364&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
createGroup.aspx">
  <input name="parent" value="337364">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=createGroup>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
  $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

  $loginparams = array (
    'sEmail' => 'some@email.com',
    'sPassword' => 'secretPassword'
  );

  $creategroupparams = array (
    'iParentID' => 337364,
    'sAuthCode' => '4ahx...80u0'
  );

  $loginResult = $soapclient->login($loginparams);

  print_r($loginResult);

  $createGroupResult = $soapclient->createGroup($creategroupparams);

  $newGroup = $createGroupResult['iNewNodeID'];

  print_r($createGroupResult . " New group node: " . $newGroup);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using createGroup.WebServiceClients;

namespace createGroup
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@mail.com";
                string sPwd = "secretpassword";
                int iParent = [ID of parent node];
                int iNewNodeID;
                string sAuthCode = "";

                HttpWebRequest oLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint +
                "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
                oLogin.CookieContainer = new CookieContainer();

                HttpWebResponse oLoginResp =
                (HttpWebResponse)oLogin.GetResponse();
                string sLoginResp = new
                StreamReader(oLoginResp.GetResponseStream())
                .ReadToEnd();
                Response.Write(sLoginResp + "<br />");

                CookieContainer sessioncookie = oLogin.CookieContainer;

                HttpWebRequest oCreateGroup =
                (HttpWebRequest)WebRequest.Create(sEndpoint +
                "createGroup.aspx" + "?parent=" + iParent + "&authcode=" + sAuthCode);
                oCreateGroup.CookieContainer = sessioncookie;

                HttpWebResponse oCreateGroupResp =
                (HttpWebResponse)oCreateGroup.
                GetResponse();
                string sCreateGroupResp = new StreamReader
                (oCreateGroupResp.GetResponseStream()).ReadToEnd();
                Response.Write(sCreateGroupResp);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using createGroup.WebServiceClients;

namespace createGroup
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iParent = [ID of the parent node];
                int iNewNodeID;
                string sAuthCode = "";

                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                WebServiceClients.loginRet loginResult =
                proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                WebServiceClients.createGroupRet cGroup =
                proxy.createGroup(iParent,
                sAuthCode, out iNewNodeID);
                Response.Write(cGroup + "<br />" + iNewNodeID);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}
```

getGroup

Retrieves information about a user group. You can view the same information in the Rescue Administration Center by selecting a user group. For information about group permissions, see

How to Create a Technician Group and Assign Permissions in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
node	The ID of the user group of which information you want to retrieve. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK NODEID: 337364 NAME: TechGroup DESCRIPTION: TechGroup1 TYPE:
TechnicianGroup PERMISSIONS:
RemoteControl, RemoteView, SendFile, SendURL, ViewSystemInformation,
RebootClientComputer,
ReceiveFile, SessionRecording, PrivateSessions, FileManagement,
UnlimitedScripting,
SessionTransferNotAllowed, WindowsCredentialsRequestNotAllowed,
ClipboardSynchronizationNotAllowed, CallingCardDeployment,
ScreenSharingNotAllowed,
CollaborationNotAllowed, DeviceConfiguration, ChatSuppression,
ChatSuppressionConsoleSwitch,
EditCustomField, LOCKEDPERMISSIONS: CallingCardDeployment, ISENBLED: True
STANDARDCONCURRENTLIMIT: 0 MOBILECONCURRENTLIMIT: 0
```

NODEID: The Rescue ID of the user group
NAME: Name of the user group
DESCRIPTION: Description of the user group
TYPE: Group type, which can be Technician Group, Administrator Group, Administrator Group Link, or Pseudo Group
PERMISSIONS: Group permissions as defined on the Organization tab of the Administration Center
LOCKEDPERMISSIONS: Group permissions that only a Master Administrator can change
ISENBLED: Defines whether the group is enabled or not
STANDARDCONCURRENTLIMIT: The maximum number of concurrent technicians with standard licences
MOBILECONCURRENTLIMIT: The maximum number of concurrent technicians with mobile licences

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving information about a user group succeeded.
NOTLOGGEDIN	Retrieving information about a user group failed because the current user is no longer logged in.

Displayed Return Value	Description
INVALIDPARAM_NODE	The specified ID is not the ID of an existing user group.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getGroup` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getGroup.aspx?
node=337364&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getGroup.aspx">
  <input name="node" value="337364">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getGroup>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

$loginparams = array (
  'sEmail' => 'some@email.com',
  'sPassword' => 'secretPassword'
);

$getgroupparams = array (
  'iNodeID' => 337364,
  'sAuthCode' => '4ahx...80u0'
);

$loginResult = $soapclient->login($loginparams);

print_r($loginResult);

$getGroupResult = $soapclient->getGroup($getgroupparams);
```

```
print_r($getResult);  
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using System.Net;  
using System.IO;  
using getGroup.WebServiceClients;  
  
namespace getGroup  
{  
    public partial class _Default : System.Web.UI.Page  
    {  
        protected void Page_Load(object sender, EventArgs e)  
        {  
            try  
            {  
                string sEndpoint = "https://secure.logmeinrescue.com/api/";  
                string sEmail = "some@email.com";  
                string sPwd = "secretPassword";  
                int iNode = 337364;  
                string sAuthCode = "4ahx...80u0";  
  
                HttpWebRequest oLogin =  
                (HttpWebRequest)WebRequest.Create(sEndpoint +  
                "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);  
                oLogin.CookieContainer = new CookieContainer();  
  
                HttpWebResponse oLoginResp =  
                (HttpWebResponse)oLogin.GetResponse();  
                string sLoginResp = new  
                StreamReader(oLoginResp.GetResponseStream())  
                .ReadToEnd();  
                Response.Write(sLoginResp + "<br />");  
  
                CookieContainer sessioncookie = oLogin.CookieContainer;  
  
                HttpWebRequest oGetGroup =  
                (HttpWebRequest)WebRequest.Create(sEndpoint +  
                "getGroup.aspx" + "?node=" + iNode + "&authcode=" + sAuthCode);  
                oGetGroup.CookieContainer = sessioncookie;  
  
                HttpWebResponse oGetGroupResp =  
                (HttpWebResponse)oGetGroup.GetResponse();  
                string sGetGroupResp = new  
                StreamReader(oGetGroupResp.GetResponseStream())  
                .ReadToEnd();  
                Response.Write(sGetGroupResp);  
            }  
            catch (Exception ex)  
            {  
                lblError.Text = ex.Message;  
            }  
        }  
    }  
}
```

```
}  
}
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using System.Net;  
using System.IO;  
using createGroup.WebServiceClients;  
  
namespace createGroup  
{  
    public partial class _Default : System.Web.UI.Page  
    {  
        protected void Page_Load(object sender, EventArgs e)  
        {  
            try  
            {  
                string sEmail = "some@mail.com";  
                string sPwd = "secretpassword";  
                int iNode = [ID of node in which information is being  
retrieved];  
                string sAuthCode = "";  
                Group oGroup = new Group();  
  
                WebServiceClients.API proxy = new WebServiceClients.API();  
                proxy.CookieContainer = new CookieContainer();  
  
                WebServiceClients.loginRet loginResult =  
proxy.login(sEmail, sPwd);  
                Response.Write(loginResult + "<br />");  
  
                WebServiceClients.getGroupRet oGetGroup =  
proxy.getGroup(iNode, sAuthCode,  
out oGroup);  
                Response.Write(oGetGroup.ToString());  
            }  
            catch (Exception ex)  
            {  
                lblError.Text = ex.Message;  
            }  
        }  
    }  
}
```

getGroup_v2

Retrieves information about a user group. You can view the same information in the Rescue Administration Center by selecting a user group. For information about group permissions, see

Changes in Version 2 of getGroup

The following has been introduced in the `getGroup_v2` method:

- `UnattendedAccess` and `SessionHoldNotAllowed` have been added as new node permissions.
- The `RPATCONCURRENTLIMIT` option is now available for displaying the limit of concurrent RPAT-enabled technicians.

Input Parameters

Element	Description
node	The ID of the user group of which information you want to retrieve. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK NODEID: 337364 NAME: TechGroup DESCRIPTION: TechGroup1 TYPE:
  TechnicianGroup PERMISSIONS:
  RemoteControl, RemoteView, SendFile, SendURL, ViewSystemInformation,
  RebootClientComputer,
  ReceiveFile, SessionRecording, PrivateSessions, FileManagement,
  UnlimitedScripting,
  SessionTransferNotAllowed, WindowsCredentialsRequestNotAllowed,
  ClipboardSynchronizationNotAllowed, CallingCardDeployment,
  ScreenSharingNotAllowed,
  CollaborationNotAllowed, DeviceConfiguration, ChatSuppression,
  ChatSuppressionConsoleSwitch,
  EditCustomField, UnattendedAccess, SessionHoldNotAllowed,
  LOCKEDPERMISSIONS:
  CallingCardDeployment, ISENBLED: True STANDARDCONCURRENTLIMIT: 0
  MOBILECONCURRENTLIMIT: 0
  RPATCONCURRENTLIMIT: 0
```

NODEID: The Rescue ID of the user group

NAME: Name of the user group

DESCRIPTION: Description of the user group

TYPE: Group type, which can be Technician Group, Administrator Group, Administrator Group Link, or Pseudo Group

PERMISSIONS: Group permissions as defined on the Organization tab of the Administration Center

LOCKEDPERMISSIONS: Group permissions that only a Master Administrator can change

ISENBLED: Defines whether the group is enabled or not

STANDARDCONCURRENTLIMIT: The maximum number of concurrent technicians with standard licences

MOBILECONCURRENTLIMIT: The maximum number of concurrent technicians with mobile licences

RPATCONCURRENTLIMIT: The maximum number of concurrent RPAT-enabled technicians

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving information about a user group succeeded.
NOTLOGGEDIN	Retrieving information about a user group failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing user group.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getGroup_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getGroup_v2.aspx?node=337364&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/getGroup_v2.aspx">  
  <input name="node" value="337364">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

C# with SOAP

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Net;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using apiSamples.WebServiceClients;  
  
namespace apiSamples  
{  
  public partial class SOAP : System.Web.UI.Page  
  {  
    protected void Page_Load(object sender, EventArgs e)  
    {  
      try
```


Output

```
OK
NODEID: 4626880
NAME: TechGroup
DESCRIPTION: Description of TechnicianGroup
TYPE: TechnicianGroup
PERMISSIONS: RemoteView, SendFile, SendURL, ViewSystemInformation,
  RebootClientComputer,
  ReceiveFile, SessionRecording, PrivateSessions, OneTimePermission,
  FileManagement, ScriptRunning,
  UnlimitedScripting, ChatSuppressionConsoleSwitch,
  SessionTransferNotAllowed, WindowsCredentialsRequestNotAllowed,
  ClipboardSynchronizationNotAllowed, CallingCardDeployment,
  ScreenSharingNotAllowed,
  CollaborationNotAllowed, DeviceConfiguration, ChatSuppression,
  EditCustomField, UnattendedAccess, SessionHoldNotAllowed,
  InviteExternalTechnicians, RemoteControlDesktop,
  RemoteControlMobile
LOCKEDPERMISSIONS: SendFile
ISENABLED: True
STANDARDCONCURRENTLIMIT: 0
MOBILECONCURRENTLIMIT: 0
RPATCONCURRENTLIMIT: 0
```

NODEID: The Rescue ID of the user group
NAME: Name of the user group
DESCRIPTION: Description of the user group
TYPE: Group type, which can be Technician Group, Administrator Group, Administrator Group Link, or Pseudo Group
PERMISSIONS: Group permissions as defined on the Organization tab of the Administration Center
LOCKEDPERMISSIONS: Group permissions that only a Master Administrator can change
ISENABLED: Defines whether the group is enabled or not
STANDARDCONCURRENTLIMIT: The maximum number of concurrent technicians with standard licences
MOBILECONCURRENTLIMIT: The maximum number of concurrent technicians with mobile licences
RPATCONCURRENTLIMIT: The maximum number of concurrent RPAT-enabled technicians

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving information about a user group succeeded.
NOTLOGGEDIN	Retrieving information about a user group failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing user group.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.

Displayed Return Value	Description
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getGroup_v3` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getGroup_v2.aspx?
node=337364&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getGroup_v2.aspx">
  <input name="node" value="337364">
  <input name="authcode" value="4ahx...80u0">
</form>
```

C# with SOAP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@mail.com";
                string sPwd = "secretpassword";
                int iNode = [ID of node in which information is being
retrieved];

                string sAuthCode = "4ahx...80u0";
                Group_V3 oGroupV3 = new Group_V3();

                //Login
                WebServiceClients.loginRet loginResult =
proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");
            }
            catch { }
        }
    }
}
```

```

        WebserviceClients.getGroupRet oGetGroupV3 =
proxy.getGroup_v3(iNode, sAuthCode, out oGroupV3);
        Response.Write(oGetGroupV3.ToString());
    }
    catch (Exception ex)
    {
        lblError.Text = ex.Message;
    }
}
}
}
}
}
}

```

setGroup

Updates the properties of a user group. You can modify the same group details in the Rescue Administration Center by selecting a user group. For information about group permissions, see [How to Create a Technician Group and Assign Permissions](#) in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
node	The ID of the user group. Required.
name	The name of the user group. Required.
status	The status of the user group. Required. The possible values are the following: <ul style="list-style-type: none"> • Enabled • Disabled
description	The description of the user group. Optional.

Element	Description
permission	<p>The possible permissions of a group. Use commas (,) to define multiple permissions. Optional.</p> <p>The possible values are as follows:</p> <ul style="list-style-type: none"> • RemoteControl • RemoteView • SendFile • SendURL • ViewSystemInformation • RebootClientComputer • ReceiveFile • SessionRecording • PrivateSessions • OneTimePermission • FileManagement • ScriptRunning • UnlimitedScripting • SessionTransferNotAllowed • EditCustomField • WindowsCredentialsRequestNotAllowed • ClipboardSynchronizationNotAllowed • CallingCardDeployment • ScreenSharingNotAllowed • CollaborationNotAllowed • DeviceConfiguration • ChatSuppression • ChatSuppressionConsoleSwitch • TechEndSessionSurvey

Element	Description
lockedpermission	<p>The locked permissions of a group. Use commas (,) to define multiple permissions. Optional.</p> <p>The possible values are as follows:</p> <ul style="list-style-type: none"> • RemoteControl • RemoteView • SendURL • SendFile • ViewSystemInformation • RebootClientComputer • ReceiveFile • SessionRecording • PrivateSessions • OneTimePermission • FileManagement • ScriptRunning • UnlimitedScripting • SessionTransferNotAllowed • EditCustomField • WindowsCredentialsRequestNotAllowed • ClipboardSynchronizationNotAllowed • CallingCardDeployment • ScreenSharingNotAllowed • CollaborationNotAllowed • DeviceConfiguration • ChatSuppression • ChatSuppressionConsoleSwitch • TechEndSessionSurvey
standardconcurrentlimit	The limit of the standard concurrent seats. Required.
mobileconcurrentlimit	The limit of the mobile concurrent seats. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Updating user group properties succeeded.
NOTLOGGEDIN	Updating user group properties failed because the current user is no longer logged in.

Displayed Return Value	Description
INVALIDPARAM_NODE	The specified ID is either not the ID of a user node or does not exist.
INVALIDPARAM_NAME	The name is too long or the field is empty.
INVALIDPARAM_STATUS	The specified status is incorrect or does not exist.
INVALIDPARAM_PERMISSION	The defined permissions of the user are either incorrect or empty.
INVALIDPARAM_LOCKEDPERMISSION	The defined locked permission set of the user are either incorrect or empty.
INVALIDPARAM_STANDARDCONCURRENTLIMIT	The number of standard concurrent users cannot exceed the limit defined for either globally for the organization or for the group.
INVALIDPARAM_MOBILECONCURRENTLIMIT	The number of mobile concurrent users cannot exceed the limit defined for either globally for the organization or for the group.
NOT_ENOUGH_SEAT_LICENSES_AVAILABLE	There are not enough standard or mobile licenses available.
NODE_CANNOT_BE_ENABLED_BECAUSE_ITS_PARENT_NODE_IS_DISABLED	The status of the user cannot be set to enabled because the parent node is disabled.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setGroup` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setGroup.aspx?
node=337364&name=TechGroup&status=Enabled
&description=TechGroup1&permission=RemoteControl,
RemoteView&lockedpermission=
CallingCardDeployment&standardconcurrentlimit=10&mobileconcurrentlimit=10
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setGroup.aspx">
  <input name="node" value="337364">
  <input name="name" value="TechGroup">
  <input name="status" value="Enabled">
  <input name="description" value="TechGroup1">
  <input name="permission" value="RemoteControl, RemoteView">
  <input name="lockedpermission" value="CallingCardDeployment">
```

```
<input name="standardconcurrentlimit" value="10">
<input name="mobileconcurrentlimit" value="10">
<input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setGroup>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

$loginparams = array (
    'sEmail' => 'some@email.com',
    'sPassword' => 'secretPassword'
);

$iNodeID = 337364;
$sName = "TechGroup";
$eStatus = Enabled;
$sDescription = "TechGroup1";
$aPermission = array(
    0 => "RemoteControl",
    1 => "RemoteView"
);
$aLockedPermission = array(
    0 => "RemoteControl",
    1 => "RemoteView"
);
$iStandardConcurrentLimit = 10;
$iMobileConcurrentLimit = 10;
$sAuthCode = "4ahx...80u0";

$setgroupparams = array (
    'iNodeID' => $iNodeID,
    'sName' => $sName,
    'eStatus' => $eStatus,
    'sDescription' => $sDescription,
    'aPermission' => $aPermission,
    'aLockedPermission' => $aLockedPermission,
    'iStandardConcurrentLimit' => $iStandardConcurrentLimit,
    'iMobileConcurrentLimit' => $iMobileConcurrentLimit,
    'sAuthCode' => $sAuthCode
);

$loginResult = $soapclient->login($loginparams);

print_r($loginResult);

$setGroupResult = $soapclient->setGroup($setgroupparams);

print_r($setGroupResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iNodeID = 337364;
                string sName = "TechGroup";
                string eNodeStatus = "Enabled";
                string sDescription = "TechGroup1";
                string aPermission = "RemoteControl, RemoteView";
                string aLockedPermission = "RemoteControl, RemoteView" ;
                int iStandardConcurrentLimit = 10;
                int iMobileConcurrentLimit = 10;
                string sAuthCode = "4ahx...80u0";

                HttpWebRequest oLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint +
                "login.aspx?email=" + sEmail + "&pwd=" + sPwd);
                oLogin.CookieContainer = new CookieContainer();
                HttpWebResponse oRespLogin =
                (HttpWebResponse)oLogin.GetResponse();

                string sRespLogin = new
                StreamReader(oRespLogin.GetResponseStream())
                .ReadToEnd();

                Response.Write(sRespLogin + "<br />");

                CookieContainer sessioncookie = oLogin.CookieContainer;

                HttpWebRequest oSetGroup =
                (HttpWebRequest)WebRequest.Create(sEndpoint +
                "setGroup.aspx?node=" + iNodeID + "&name=" + sName + "&status=" +
                eNodeStatus
                + "&description=" + sDescription + "&permission=" + aPermission +
                "&lockedpermission="
                + aLockedPermission + "&standardconcurrentlimit=" +
                iStandardConcurrentLimit
                + "&mobileconcurrentlimit=" + iMobileConcurrentLimit + "&authcode=" +
                sAuthCode);

                oSetGroup.CookieContainer = sessioncookie;
                HttpWebResponse oRespSetGroup =
                (HttpWebResponse)oSetGroup.GetResponse();
            }
        }
    }
}
```

```

        string sRespSetGroup = new
StreamReader(oRespSetGroup.GetResponseStream())
.ReadToEnd();
        Response.Write(sRespSetGroup);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iNodeID = 123456;
                string sName = "Testing";
                nodeStatus eNodeStatus = nodeStatus.Enabled;
                string sDescription = "";
                NODE_PERMISSIONS[] aPermissions =
                { NODE_PERMISSIONS.RemoteControl,
                NODE_PERMISSIONS.RemoteView };
                NODE_PERMISSIONS[] aLockedPermissions =
                { NODE_PERMISSIONS.RemoteControl,
                NODE_PERMISSIONS.RemoteView };
                int iStandardConcurrentLimit = 0;
                int iMobileConcurrentLimit = 0;
                string sAuthCode = "";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
                sPwd);
                Response.Write(oLogin + "<br />");

                WebServiceClients.setGroupRet oSetGroup =
                proxy.setGroup(iNodeID, sName,
                eNodeStatus, sDescription, aPermissions, aLockedPermissions,
                iStandardConcurrentLimit,

```

```

iMobileConcurrentLimit, sAuthCode);
    Response.Write(oSetGroup + "<br />");
}
catch (Exception ex)
{
    Response.Write(ex.Message);
}
}
}
}

```

setGroup_v2

Updates the properties of a user group. You can modify the same group details in the Rescue Administration Center by selecting a user group. For information about group permissions, see *How to Create a Technician Group and Assign Permissions* in the [Rescue Administration Center User Guide](#).

Changes in Version 2 of setGroup

The following have been introduced in the `setGroup_v2` method:

- `UnattendedAccess`, `SessionHoldNotAllowed`, and `ConnectOnLan` have been added as new permissions and locked permissions for nodes.
- The `RPATCONCURRENTLIMIT` parameter is now available for displaying the limit of concurrent RPAT-enabled technicians.

Input Parameters

Element	Description
node	The ID of the user group. Required.
name	The name of the user group. Required.
status	The status of the user group. Required. The possible values are as follows: <ul style="list-style-type: none"> • Enabled • Disabled
description	The description of the user group. Optional.

Element	Description
permission	<p>The possible permissions of a group. Use commas (,) to define multiple permissions. Optional.</p> <p>The possible values are as follows:</p> <ul style="list-style-type: none"> • RemoteControl • RemoteView • SendFile • SendURL • ViewSystemInformation • RebootClientComputer • ReceiveFile • SessionRecording • PrivateSessions • OneTimePermission • FileManagement • ScriptRunning • UnlimitedScripting • SessionTransferNotAllowed • EditCustomField • WindowsCredentialsRequestNotAllowed • ClipboardSynchronizationNotAllowed • CallingCardDeployment • ScreenSharingNotAllowed • CollaborationNotAllowed • DeviceConfiguration • ChatSuppression • ChatSuppressionConsoleSwitch • UnattendedAccess • SessionHoldNotAllowed • ConnectOnLan

Element	Description
lockedpermission	<p>The locked permissions of a group. Use commas (,) to define multiple permissions. Optional.</p> <p>The possible values are as follows:</p> <ul style="list-style-type: none"> • RemoteControl • RemoteView • SendURL • SendFile • ViewSystemInformation • RebootClientComputer • ReceiveFile • SessionRecording • PrivateSessions • OneTimePermission • FileManagement • ScriptRunning • UnlimitedScripting • SessionTransferNotAllowed • EditCustomField • WindowsCredentialsRequestNotAllowed • ClipboardSynchronizationNotAllowed • CallingCardDeployment • ScreenSharingNotAllowed • CollaborationNotAllowed • DeviceConfiguration • ChatSuppression • ChatSuppressionConsoleSwitch • UnattendedAccess • SessionHoldNotAllowed • ConnectOnLan
standardconcurrentlimit	The limit of the standard concurrent seats. Required.
mobileconcurrentlimit	The limit of the mobile concurrent seats. Required.
rpatconcurrentlimit	The limit of the RPAT concurrent seats. This element is only available in the <code>setGroup_v2</code> method. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.

Displayed Return Value	Description
OK	Updating user group properties succeeded.
NOTLOGGEDIN	Updating user group properties failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is either not the ID of a user node or does not exist.
INVALIDPARAM_NAME	The name is too long or the field is empty.
INVALIDPARAM_STATUS	The specified status is incorrect or does not exist.
INVALIDPARAM_PERMISSION	The defined permission set of the user are either incorrect or empty.
INVALIDPARAM_LOCKEDPERMISSION	The defined locked permission set of the user are either incorrect or empty.
INVALIDPARAM_STANDARDCONCURRENTLIMIT	The number of standard concurrent users cannot exceed the limit defined for either globally for the organization or for the group.
INVALIDPARAM_MOBILECONCURRENTLIMIT	The number of mobile concurrent users cannot exceed the limit defined for either globally for the organization or for the group.
INVALIDPARAM_RPATCONCURRENTLIMIT	The number of RPAT concurrent users cannot exceed the limit defined for either globally for the organization or for the group. This error is only available for users of the <code>setGroup_v2</code> method.
NOT_ENOUGH_SEAT_LICENSES_AVAILABLE	There are not enough standard, mobile, or RPAT licenses available.
NODE_CANNOT_BE_ENABLED_BECAUSE_ITS_PARENT_NODE_IS_DISABLED	The status of the user cannot be set to enabled because the parent node is disabled.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setGroup_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setGroup_v2.aspx?
node=337364&name=TechGroup&status=Enabled
&description=TechGroup1&permission=RemoteControl,
RemoteView&lockedpermission=
CallingCardDeployment&standardconcurrentlimit=10&mobileconcurrentlimit=10
```



```
&rpatconcurrentlimit=5&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setGroup_v2.aspx">
  <input name="node" value="337364">
  <input name="name" value="TechGroup">
  <input name="status" value="Enabled">
  <input name="description" value="TechGroup1">
  <input name="permission" value="RemoteControl, RemoteView">
  <input name="lockedpermission" value="CallingCardDeployment">
  <input name="standardconcurrentlimit" value="10">
  <input name="mobileconcurrentlimit" value="10">
  <input name="rpatconcurrentlimit" value="5">
  <input name="authcode" value="4ahx...80u0">
</form>
```

C# with SOAP

```
using System;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iNodeID = 123456;
                string sName = "Testing";
                nodeStatus eNodeStatus = nodeStatus.Enabled;
                string sDescription = "";
                NODE_PERMISSIONS_V2[] aPermissions =
                { NODE_PERMISSIONS_V2.RemoteControl, NODE_PERMISSIONS_V2.RemoteView };
                NODE_PERMISSIONS_V2[] aLockedPermissions =
                { NODE_PERMISSIONS_V2.RemoteControl, NODE_PERMISSIONS_V2.RemoteView };
                int iStandardConcurrentLimit = 0;
                int iMobileConcurrentLimit = 0;
                int iRPATConcurrentLimit = 0;
                string sAuthCode = "4ahx...80u0";

                //Login
                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
                sPwd);
            }
        }
    }
}
```


Element	Description
permission	<p>The possible permissions of a group. Use commas (,) to define multiple permissions. Optional.</p> <p>The possible values are as follows:</p> <ul style="list-style-type: none"> • RemoteControlDesktop • RemoteControlMobile • RemoteView • SendFile • SendURL • ViewSystemInformation • RebootClientComputer • ReceiveFile • SessionRecording • PrivateSessions • OneTimePermission • FileManagement • ScriptRunning • UnlimitedScripting • SessionTransferNotAllowed • EditCustomField • WindowsCredentialsRequestNotAllowed • ClipboardSynchronizationNotAllowed • CallingCardDeployment • ScreenSharingNotAllowed • CollaborationNotAllowed • DeviceConfiguration • ChatSuppression • ChatSuppressionConsoleSwitch • UnattendedAccess • SessionHoldNotAllowed • ConnectOnLan

Element	Description
lockedpermission	<p>The locked permissions of a group. Use commas (,) to define multiple permissions. Optional.</p> <p>The possible values are as follows:</p> <ul style="list-style-type: none"> • RemoteControlDesktop • RemoteControlMobile • RemoteView • SendURL • SendFile • ViewSystemInformation • RebootClientComputer • ReceiveFile • SessionRecording • PrivateSessions • OneTimePermission • FileManagement • ScriptRunning • UnlimitedScripting • SessionTransferNotAllowed • EditCustomField • WindowsCredentialsRequestNotAllowed • ClipboardSynchronizationNotAllowed • CallingCardDeployment • ScreenSharingNotAllowed • CollaborationNotAllowed • DeviceConfiguration • ChatSuppression • ChatSuppressionConsoleSwitch • UnattendedAccess • SessionHoldNotAllowed • ConnectOnLan
standardconcurrentlimit	The limit of the standard concurrent seats. Required.
mobileconcurrentlimit	The limit of the mobile concurrent seats. Required.
rpatconcurrentlimit	The limit of the RPAT concurrent seats. This element is only available in the <code>setGroup_v2</code> method. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Updating user group properties succeeded.
NOTLOGGEDIN	Updating user group properties failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is either not the ID of a user node or does not exist.
INVALIDPARAM_NAME	The name is too long or the field is empty.
INVALIDPARAM_STATUS	The specified status is incorrect or does not exist.
INVALIDPARAM_PERMISSION	The defined permission set of the user are either incorrect or empty.
INVALIDPARAM_LOCKEDPERMISSION	The defined locked permission set of the user are either incorrect or empty.
INVALIDPARAM_STANDARDCONCURRENTLIMIT	The number of standard concurrent users cannot exceed the limit defined for either globally for the organization or for the group.
INVALIDPARAM_MOBILECONCURRENTLIMIT	The number of mobile concurrent users cannot exceed the limit defined for either globally for the organization or for the group.
INVALIDPARAM_RPATCONCURRENTLIMIT	The number of RPAT concurrent users cannot exceed the limit defined for either globally for the organization or for the group. This error is only available for users of the <code>setGroup_v2</code> method.
NOT_ENOUGH_SEAT_LICENSES_AVAILABLE	There are not enough standard, mobile, or RPAT licenses available.
NODE_CANNOT_BE_ENABLED_BECAUSE_ITS_PARENT_NODE_IS_DISABLED	The status of the user cannot be set to enabled because the parent node is disabled.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setGroup_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setGroup_v2.aspx?node=337364&name=TechGroup&status=Enabled
```

```
&description=TechGroup1&permission=RemoteControl,
RemoteView&lockedpermission=
CallingCardDeployment&standardconcurrentlimit=10&mobileconcurrentlimit=10
&rpatconcurrentlimit=5&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setGroup_v2.aspx">
  <input name="node" value="337364">
  <input name="name" value="TechGroup">
  <input name="status" value="Enabled">
  <input name="description" value="TechGroup1">
  <input name="permission" value="RemoteControlDesktop, RemoteView">
  <input name="lockedpermission" value="CallingCardDeployment">
  <input name="standardconcurrentlimit" value="10">
  <input name="mobileconcurrentlimit" value="10">
  <input name="rpatconcurrentlimit" value="5">
  <input name="authcode" value="4ahx...80u0">
</form>
```

C# with SOAP

```
using System;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iNodeID = 123456;
                string sName = "Testing";
                nodeStatus eNodeStatus = nodeStatus.Enabled;
                string sDescription = "";
                NODE_PERMISSIONS_V4[] aPermissions
= { NODE_PERMISSIONS_V4.RemoteControlDesktop,
NODE_PERMISSIONS_V4.RemoteView };
                NODE_PERMISSIONS_V4[] aLockedPermissions
= { NODE_PERMISSIONS_V4.RemoteControlDesktop,
NODE_PERMISSIONS_V4.RemoteView };
                int iStandardConcurrentLimit = 0;
                int iMobileConcurrentLimit = 0;
                int iRPATConcurrentLimit = 0;
            }
        }
    }
}
```

```
        string sAuthCode = "4ahx...80u0";

        //Login
        WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
        Response.Write(oLogin + "<br />");

        WebServiceClients.setGroupRet_v3 oSetGroupV3 =
proxy.setGroup_v3(iNodeID, sName,
eNodeStatus, sDescription, aPermissions,
aLockedPermissions, iStandardConcurrentLimit,
iMobileConcurrentLimit, iRPATConcurrentLimit, sAuthCode);
        Response.Write(oSetGroupV3 + "<br />");
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
```

moveNode

Moves a node from one parent node to another in the company hierarchy. This is equivalent to performing a drag and drop on a node in the organization tree.

Input Parameters

Element	Description
targetNode	The ID of the node that will be the new parent node. Required.
node	The ID of the node that you want to move. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Moving a node succeeded.

Displayed Return Value	Description
NOTLOGGEDIN	Moving a node failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing source node.
INVALIDPARAM_TARGETNODE	The specified ID is not the ID of an existing target node.
INVALID_MOVE	The node cannot be moved to the target node because the target node does not exist or the specified node that you want to move cannot be the child of the target node. For example, a technician cannot be moved to an Administrator Group.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `moveNode` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/moveNode.aspx?
targetnode=337364&node=337366
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
moveNode.aspx">
  <input name="targetNodeID" value="337364">
  <input name="nodeID" value="337366">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=moveNode>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

$loginparams = array (
```



```

'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

$iTargetNodeID = 337364;
$iNodeID = 337366;
$sAuthCode = "4ahx...80u0";

$movenodeparams = array (
    'iTargetNodeID' => $iTargetNodeID,
    'iNodeID' => $iNodeID,
    'sAuthCode' => ''
);

$loginResult = $soapclient->login($loginparams);

print_r($loginResult);

$moveNodeResult = $soapclient->moveNode($movenodeparams);

print_r($moveNodeResult);
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using createGroup.WebServiceClients;

namespace createGroup
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iTargetNode = 337364;
                int iNode = 337366;
                string sAuthCode = "4ahx...80u0"

                HttpWebRequest oReqLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint +
                "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
                oReqLogin.CookieContainer = new CookieContainer();
                CookieContainer sessioncookie = oReqLogin.CookieContainer;

                HttpWebResponse oRespLogin =
                (HttpWebResponse)oReqLogin.GetResponse();
                string sRespLogin = new
                StreamReader(oRespLogin.GetResponseStream())

```

```

.ReadToEnd();
    Response.Write(sRespLogin + "<br />");

    HttpWebRequest oReqMoveNode =
    (HttpWebRequest)WebRequest.Create(sEndpoint +
    "movenode.aspx" + "?targetnode=" + iTargetNode + "&node=" + iNode +
    "&authcode="
    + sAuthCode);
    oReqMoveNode.CookieContainer = sessioncookie;

    HttpWebResponse oRespMoveNode =
    (HttpWebResponse)oReqMoveNode.GetResponse();
    string sRespMoveNode = new
    StreamReader(oRespMoveNode.GetResponseStream())
    .ReadToEnd();
    Response.Write(sRespMoveNode);
}
catch (Exception ex)
{
    lblError.Text = ex.Message;
}
}
}
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using createGroup.WebServiceClients;

namespace createGroup
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iTargetNode = 337364;
                int iNode = 337366;
                string sAuthCode = "4ahx...80u0";

                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                WebServiceClients.loginRet loginResult =
                proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                WebServiceClients.moveNodeRet oMoveNode =
                proxy.moveNode(iTargetNode, iNode,

```

```

sAuthCode);
        Response.Write(oMoveNode);
    }
    catch (Exception ex)
    {
        lblError.Text = ex.Message;
    }
}
}
}

```

deleteNode

Deletes a node from the company hierarchy. You can delete a node in the Rescue Administration Center by right-clicking on the node and selecting **Delete**.

Input Parameters

Element	Description
node	The ID of the node to delete. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Deleting a node succeeded.
NOTLOGGEDIN	Deleting a node failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing node.
INVALID_DELETE	The node cannot be deleted because the user who is currently logged in has insufficient privileges. An administrator cannot delete himself either.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `deleteNode` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/deleteNode.aspx?
node=337366&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
deleteNode.aspx">
  <input name="nodeID" value="337366">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=deleteNode>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
  $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

  $loginparams = array (
    'sEmail' => 'some@email.com',
    'sPassword' => 'secretPassword'
  );

  $deletenodeparams = array (
    'iNodeID' => 337366,
    'sAuthCode' => '4ahx...80u0'
  );

  $loginResult = $soapclient->login($loginparams);

  print_r($loginResult);

  $deleteNodeResult = $soapclient->deleteNode($deletenodeparams);

  print_r($deleteNodeResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using deleteNode.WebServiceClients;
```

```

namespace deleteNode
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iNode = 337366;
                string sAuthCode = "4ahx...80u0"

                HttpWebRequest oReqLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint +
                "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
                oReqLogin.CookieContainer = new CookieContainer();
                CookieContainer sessioncookie = oReqLogin.CookieContainer;

                HttpWebResponse oLoginResp =
                (HttpWebResponse)oReqLogin.GetResponse();
                string sLoginResp = new
                StreamReader(oLoginResp.GetResponseStream())
                .ReadToEnd();
                Response.Write(sLoginResp + "<br />");

                HttpWebRequest oReqDeleteNode =
                (HttpWebRequest)WebRequest.Create
                (sEndpoint + "deleteNode.aspx" + "?node=" + iNode + "&authcode=" +
                sAuthCode);

                HttpWebResponse oDeleteNodeResp = (HttpWebResponse)
                oReqDeleteNode.GetResponse();
                string sDeleteNodeResp = new StreamReader
                (oDeleteNodeResp.GetResponseStream()).ReadToEnd();
                Response.Write(sDeleteNodeResp);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
using deleteNode.WebServiceClients;

namespace deleteNode
{

```

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        int iNode = 337366;
        string sAuthCode = "4ahx...80u0";

        WebServiceClients.API proxy = new WebServiceClients.API();
        proxy.CookieContainer = new CookieContainer();

        WebServiceClients.loginRet loginResult = proxy.login(sEmail,
sPwd);
        Response.Write(loginResult + "<br />");

        WebServiceClients.deleteNodeRet oDeleteNode =
proxy.deleteNode(iNode, sAuthCode);
        Response.Write(oDeleteNode);
    }
}
}

```

assignChannel

Assigns a channel to a technician or Technician Group. For information about assigning channels in the Rescue Administration Center, see *How to Assign a Channel to a Technician Group* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
node	The ID of the technician or Technician Group. Required.
entry	The ID of the channel that you want to assign. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Assigning a channel succeeded.
NOTLOGGEDIN	Assigning a channel failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The ID of a technician or a Technician Group is not defined.
INVALIDPARAM_ENTRY	The ID of a channel is not defined.

Displayed Return Value	Description
INVALID_ASSIGN	The channel cannot be assigned to the selected node because it is not an existing technician or Technician Group.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `assignChannel` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/assignChannel.aspx?node=337364
&entry=1902861082&authcode=...
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
assignChannel.aspx">
  <input name="node" value="337364">
  <input name="entry" value="1902861082">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=assignChannel>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

$loginparams = array (
  'sEmail' => 'some@email.com',
  'sPassword' => 'secretPassword'
);

$assignchannelparams = array (
  'iEntryID' => 1902861082,
  'iNodeID' => 337364,
  'sAuthCode' => '4ahx...80u0'
);

$loginResult = $soapclient->login($loginparams);
```

```

print_r($loginResult);

$assignChannelResult = $soapclient->assignChannel($assignchannelparams);

print_r($assignChannelResult);
?>

```

C# with HttpRequest

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;

namespace assignChannel
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iNode = 337364;
                int iEntry = 1902861082;
                //sAuthCode is the return value of the requestAuthCode API call
                string sAuthCode = "4ahx...80u0";

                HttpRequest oReqLogin =
                (HttpRequest)WebRequest.Create(sEndpoint +
                "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
                oReqLogin.CookieContainer = new CookieContainer();
                CookieContainer sessioncookie = oReqLogin.CookieContainer;

                HttpResponse oLoginResp =
                (HttpResponse)oReqLogin.GetResponse();
                string sLoginResp = new
                StreamReader(oLoginResp.GetResponseStream()).ReadToEnd();
                Response.Write(sLoginResp + "<br />");

                HttpRequest oAssign =
                (HttpRequest)WebRequest.Create(sEndpoint +
                "assignChannel.aspx" + "?node=" + iNode + "&entry=" + iEntry +
                "&authcode=" + sAuthCode);
                oAssign.CookieContainer = sessioncookie;

                HttpResponse oAssignResp =
                (HttpResponse)oAssign.GetResponse();
                string sAssignResp = new
                StreamReader(oAssign.GetResponseStream())
                .ReadToEnd();
                Response.Write(sAssignResp);
            }
            catch (Exception ex)

```



```

        {
            lblError.Text = ex.Message;
        }
    }
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;

namespace assignChannel
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iNode = 337364;
                int iEntry = 1902861082;
                //sAuthCode is the return value of the requestAuthCode API
                string sAuthCode = "4ahx...80u0";

                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                WebServiceClients.loginRet loginResult =
                proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                WebServiceClients.assignChannelRet oAssign =
                proxy.assignChannel(iEntry,
                iNode, sAuthCode);
                Response.Write(oAssign);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}

```

unassignChannel

Unassigns a technician or Technician Group from a channel. For information about unassigning channels in the Rescue Administration Center, see *How to Assign a Channel to a Technician Group* in the *Rescue Administration Center User Guide*.

Input Parameters

Element	Description
node	The ID of the technician or Technician Group. Required.
entry	The ID of the channel. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Unassigning a technician or Technician Group succeeded.
NOTLOGGEDIN	Unassigning a technician or Technician Group failed because the current user is no longer logged in.
INVALIDPARAM_NODE	The ID of a technician or a Technician Group is not defined.
INVALIDPARAM_ENTRY	The ID of a channel is not defined.
INVALID_DEASSIGN	The channel cannot be unassigned from a technician or Technician Group.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `unassignChannel` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/unassignChannel.aspx?node=337364
&entry=1902861082&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/unassignChannel.aspx">
  <input name="node" value="337364">
  <input name="entry" value="1902861082">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=unassignChannel>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
    $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

    $loginparams = array (
        'sEmail' => 'some@email.com',
        'sPassword' => 'secretPassword'
    );

    $iNodeID = 337364;
    $iEntryID = 1902861082;
    $sAuthCode = "4ahx...80u0";

    $unassignchannelparams = array (
        'iNodeID' => $iNodeID,
        'iEntryID' => $iEntryID,
        'sAuthCode' => $sAuthCode
    );

    $loginResult = $soapclient->login($loginparams);

    print_r($loginResult);

    $unassignChannelResult = $soapclient->unassignChannel($unassignchannelparams);

    print_r($unassignChannelResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;
```

```

namespace assignChannel
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iNode = 337364;
                int iEntry = 1902861082;
                string sAuthCode = "4ahx...80u0";

                HttpWebRequest oReqLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
                oReqLogin.CookieContainer = new CookieContainer();
                CookieContainer sessioncookie = oReqLogin.CookieContainer;

                HttpWebResponse oLoginResp =
                (HttpWebResponse)oReqLogin.GetResponse();
                string sLoginResp = new
                StreamReader(oLoginResp.GetResponseStream())
                .ReadToEnd();
                Response.Write(sLoginResp + "<br />");

                HttpWebRequest oUnassign =
                (HttpWebRequest)WebRequest.Create(sEndpoint
+ "unassignChannel.aspx" + "?node=" + iNode + "&entry=" + iEntry +
                "&authcode="
                + sAuthCode);
                oUnassign.CookieContainer = sessioncookie;

                HttpWebResponse oUnnassignResp =
                (HttpWebResponse)oUnassign.GetResponse();
                string sUnnassignResp = new
                StreamReader(oUnnassignResp.GetResponseStream())
                .ReadToEnd();
                Response.Write(sUnnassignResp);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Net;
using System.IO;

```

```

namespace assignChannel
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iNode = 337364;
                int iEntry = 1902861082;
                string sAuthCode = "4ahx...80u0";

                WebserviceClients.API proxy = new WebserviceClients.API();
                proxy.CookieContainer = new CookieContainer();

                WebserviceClients.loginRet loginResult =
proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                WebserviceClients.unassignChannelRet oUnassign =
proxy.unassignChannel
(iEntry, iNode, sAuthCode);
                Response.Write(oUnassign);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}

```

Supporting CRM Systems

These methods enable you to integrate Rescue into your existing CRM system.

generateEmailText

Generates the Subject, Text, Link and Signature of a standard connection email based on Administration Center settings. The link contains the download URL for the Applet with the specified PIN code. For information about starting a session via email, see *How to Start a Session via Email* in the [Rescue Technician Console User Guide](#).



Important: The maximum number of calls to this endpoint is limited to a single call per second to prevent malicious email flooding.

Input Parameters

Element	Description
pincode	The PIN code of the technician session, which must be 6-digit long. Required.

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK SUBJECT:Connection email link TEXT:Please click the link below to
request a live support
session: LINK:http://yourorganization.com/R?i=2&Code=123456 SIGNATURE:Best
Regards, Support Team
```

```
SUBJECT: Email subject
TEXT: Email body
LINK: Link to the support PIN code
SIGNATURE: Email signature
```

Return Values

Displayed Return Value	Description
ERROR	<ul style="list-style-type: none"> An unspecified error occurred, such as timeout. The maximum allowed call rate of a single call per second has been exceeded.
OK	An email was successfully generated.
NOTLOGGEDIN	An email was not generated because the current user is no longer logged in.
NOTTECHNICIAN	The API user is not a technician, so he cannot initiate a private session.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.
INVALIDPINFORMAT	The format of the PIN code in the email is incorrect.

Sample Code

The following are examples for using the `generateEmailText` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/generateEmailText.aspx?pincode=123456
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/generateEmailText.aspx">
    <input name="pincode" value="123456">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=generateEmailText>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

//define parameters
$loginparams = array (
    'sEmail' => 'some@email.com',
    'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//set customer info
//you would normally do this via a form or dynamically from your CRM
$sCField0 = "some customer";
$sCField1 = "first custom field";
$sCField2 = "second custom field";
$sCField3 = "third custom field";
$sCField4 = "fourth custom field";
$sCField5 = "fifth custom field";
$sTracking0 = $sCField0 . date("c"); //to ensure unique
$bNoTech = true; //specifies whether Technician Console must be running

//create params array for SOAP request
$requestPINCodeParams = array(
    'sCField0' => $sCField0,
    'sCField1' => $sCField1,
    'sCField2' => $sCField2,
    'sCField3' => $sCField3,
    'sCField4' => $sCField4,
    'sCField5' => $sCField5,
    'sTracking0' => $sTracking0,
    'bNoTechConsole' => $bNoTech
);

$requestPINCodeResult = $soapclient->requestPINCode($requestPINCodeParams);

//print out the PIN Code full response
```

```

print_r("<b>requestPINCode full response.</b><br />");
print_r($requestPINCodeResult);
echo "<br /><br />"; //formatting

//show the PIN code only
$PINCode = $requestPINCodeResult["iPINCode"];
$sAuthCode = "";

//print PIN Code
print_r("<b>PIN Code.</b><br />");
print_r($PINCode);
echo "<br /><br />"; //formatting

//generate email text
//create array for email text request
$pin = array(
    'iPINCode' => $PINCode,
    'sAuthCode' => $sAuthCode
);

$emailtext = $soapclient->generateEmailText($pin);

//show the full email text response
print_r("<b>generateEmailText full response.</b><br />");
print_r($emailtext);

//email text formatted
print_r("<b>Formatted email text.</b><br />");
print_r($emailtext["sSubject"] . "<br /><br />");
print_r($emailtext["sText"] . "<br /><br />");
print_r($emailtext["sLink"] . "<br /><br />");
print_r($emailtext["sSignature"]);

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
    + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

```



```

        //get the response
        HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

        //get the EmailText info
        HttpRequest oReqEmailText =
            (HttpRequest) System.Net.WebRequest.Create(sEndpoint
                + "generateEmailText.aspx?pincode=" + iPIN + "&authcode=" + sAuthCode);
        oReqEmailText.CookieContainer = sessioncookie;

        HttpResponse oRespEmailText =
            (HttpResponse) oReqEmailText.GetResponse();
        string sRespEmailText = new
            StreamReader(oRespEmailText.GetResponseStream()).ReadToEnd();
        Response.Write("requestEmailText result: " + sRespEmailText + "<br />");
        //You can customize
        the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API generateEmailText Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                int iPIN;
                string sSubject;
                string sText;
                string sLink;
                string sSignature;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
                Response.Write(oLogin + "<br />");

                WebServiceClients.requestPINCodeRet oRequestPIN =
proxy.requestPINCode("", "", "", "", "", "", "", true, sAuthCode, out
iPIN);
                Response.Write(oRequestPIN + "<br />");

                WebServiceClients.generateEmailTextRet oEmailText =
proxy.generateEmailText
(iPIN, sAuthCode, out sSubject, out sText, out sLink, out sSignature);
                Response.Write(oEmailText + "<br />");

                Response.Write("Subject: " + sSubject + "<br />Body: " +
sText +
"<br />Link: " + sLink + "<br />Signature: " + sSignature);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

getLoginTicket

Generates a ticket that allows the technician to log in to a Rescue account from the company intranet website using single sign-on.

Input Parameters

Element	Description
ssoid	The single sign-on ID of the Rescue account. Required.
Password	The password of the corresponding Rescue account. Required.
CompanyID	The identifier of the company using single sign-on to authenticate technicians. Required.

Output

```
OK:https://secure.logmeinrescue.com/SSO/Login.aspx?Ticket=f70..156
```

Return Values

Displayed Return Value	Description
OK	Generating a login ticket succeeded.
INVALIDPASSWORD	The single sign-on ID does not identify a technician, and/or the specified password does not authenticate a technician.
INVALIDCOMPANYID	The specified company ID does not exist.
MISSINGPARAMETERS	At least one of the required parameters was not provided.

Sample Code

The following are examples for using the `getLoginTicket` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/SSO/GetLoginTicket.aspx?  
ssoid=123456&Password=secretPassword  
&CompanyID=654321
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/SSO/  
GetLoginTicket.aspx">  
  <input name="ssoid" value="123456">
```

```
<input name="Password" value="secretPassword">
<input name="CompanyID" value="654321">
</form>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
string sSSOID = [your User ID];
string sSSOPassword = [Should be set manually];

System.Net.HttpWebRequest oReq =
    (System.Net.HttpWebRequest) System.Net.WebRequest.Create
    ("https://secure.logmeinrescue.com/SSO/GetLoginTicket.aspx?SSOID=" + sSSOID
    + "&Password="
    + sSSOPassword + "&CompanyID=112938");
System.Net.HttpWebResponse oResp =
    (System.Net.HttpWebResponse) oReq.GetResponse();
string sResp = new
    System.IO.StreamReader(oResp.GetResponseStream()).ReadToEnd();

if (sResp.StartsWith("OK:"))
    Response.Redirect( sResp.Substring("OK:".Length) );
else
    Response.Write( sResp );
```

getDTCLoginTicket

Generates a ticket that allows the technician to log in to the Rescue Desktop Technician Console from the company intranet website using single sign-on.

Input Parameters

Element	Description
ssoid	The single sign-on ID of the Rescue account. Required.
Password	The password of the corresponding Rescue account. Required.
CompanyID	The identifier of the company using single sign-on to authenticate technicians. Required.
arch=64	To be used when 64-bit Desktop Technician Console is needed. Optional.

Output

```
https://secure.logmeinrescue.com/TechConsole/DesktopApp/DownloadSSO.aspx?
companyid=654321&ticket=4c6f1815-1e0c-43ab-8117-d79b8f523824&arch=64
```

Return Values

Displayed Return Value	Description
OK	Generating a login ticket succeeded.
INVALIDPASSWORD	The single sign-on ID does not identify a technician, and/or the specified password does not authenticate a technician.
INVALIDCOMPANYID	The specified company ID does not exist.
MISSINGPARAMETERS	At least one of the required parameters was not provided.

Sample Code

The following are examples for using the `getDTCLoginTicket` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/SSO/GetDTCLoginTicket.aspx?
ssoid=123456&Password=secretPassword&CompanyID=654321&arch=64
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/SSO/
GetDTCLoginTicket.aspx">
  <input name="ssoid" value="123456">
  <input name="Password" value="secretPassword">
  <input name="CompanyID" value="654321">
  <input name="arch" value="64">
</form>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
string sSSOID = [your User ID];
string sSSOPassword = [Should be set manually];
bool is64Bit = true;

System.Net.HttpWebRequest oReq =
    (System.Net.HttpWebRequest)System.Net.WebRequest.Create
    ("https://secure.logmeinrescue.com/SSO/GetDTCLoginTicket.aspx?SSOID=" +
    sSSOID + "&Password="
    + sSSOPassword + "&CompanyID=112938" + (is64Bit? "&arch=64" : ""));
System.Net.HttpWebResponse oResp =
    (System.Net.HttpWebResponse)oReq.GetResponse();
string sResp = new
    System.IO.StreamReader(oResp.GetResponseStream()).ReadToEnd();

if (sResp.StartsWith("OK:"))
    Response.Redirect(sResp.Substring("OK:".Length));
else
```

```
Response.Write( sResp );
```

requestAuthCode

Generates an authentication code that prevents the technician from logging out if there is a timeout.

Input Parameters

Element	Description
email	The email address of the Rescue account. Required.
pwd	The password of the corresponding Rescue account. Required.

Output

```
OK AUTHCODE:  
4ahx99bppkzxocbs590roo1l99o7au9rmwnea8q3areaiceawlafq55jx89n15zt63hxlo298qf80u0
```

AUTHCODE: Secret authentication code

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Generating an authentication code succeeded.
INVALID	The password and email do not identify and authenticate a technician.
requestAuthCode_PollRateExceeded	There is this response when there is more than 1 requestAuthCode API call within a second. The current limit is one requestAuthCode API call per second.

Sample Code

The following are examples for using the requestAuthCode method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/requestAuthCode.aspx?  
email=some@email.com&pwd=secretPassword
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/requestAuthCode.aspx">
    <input name="email" value="some@email.com">
    <input name="pwd" value="secretPassword">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=requestAuthCode>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
    $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

    $sEmail = "some@email.com";
    $sPwd = "secretPassword";

    $requestauthcodeparams = array(
        'sEmail' => $sEmail,
        'sPassword' => $sPwd
    );

    $requestAuthCodeResult = $soapclient->requestAuthCode($requestauthcodeparams);

    print_r($requestAuthCodeResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
            }
        }
    }
}
```

```

        HttpWebRequest oRequestAuthCode =
        (HttpWebRequest)WebRequest.Create(sEndpoint
        + "requestAuthCode.aspx?email=" + sEmail + "&pwd=" + sPwd);
        HttpWebResponse oRespAuthCode =
        (HttpWebResponse)oRequestAuthCode
        .GetResponse();

        string sRespAuthCode = new
        StreamReader(oRespAuthCode.GetResponseStream())
        .ReadToEnd();
        Response.Write(sRespAuthCode);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                string sAuthCode = "";

                WebServiceClients.requestAuthCodeRet oRequestAuthCode =
                proxy.requestAuthCode
                (sEmail, sPwd, out sAuthCode);
                Response.Write(oRequestAuthCode + "<br />");

                Response.Write("authCode: " + sAuthCode);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```




```
}
```

requestAuthCodeSSO

Generates an authentication code that prevents the technician who uses single sign-on from logging out if there is a timeout.

Input Parameters

Element	Description
ssoid	The single sign-on ID of the Rescue user. Required.
pwd	The single sign-on password of the company's Rescue account. Required.
company	The identifier of the company using single sign-on to authenticate technicians. Required.  Note: The identifier must contain numbers only.

Output

```
OK AUTHCODE:  
4ahx99bppkzxcbs590roo1199o7au9rmwnea8q3areaiceawlafq55jx89n15zt63hxlo298qf80u0
```

AUTHCODE: Secret authentication code

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Generating an authentication code succeeded.
INVALID	The password and single sign-on ID do not identify and authenticate a technician.
INVALIDPARAM_COMPANY	The provided company identifier contains non-numeric characters.

Sample Code

The following are examples for using the `requestAuthCodeSSO` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/requestAuthCodeSSO.aspx?  
ssoid=123456&pwd=secretPassword
```

```
&company=654321
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/requestAuthCodeSSO.aspx">
  <input name="ssoid" value="123456">
  <input name="pwd" value="secretPassword">
  <input name="company" value="654321">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=requestAuthCodeSSO>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
  $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

  $sSSOID = "123456";
  $sPwd = "secretPassword";
  $sCompany = "654321";

  $requestauthcodeparams = array(
    'sSSOID' => $sSSOID,
    'sPassword' => $sPwd,
    'sCompany' => $sCompany
  );

  $requestAuthCodeResult = $soapclient->requestAuthCodeSSO($requestauthcodeparams);

  print_r($requestAuthCodeResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
  public partial class _Default : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
  }
}
```

```

    {
        try
        {
            string sEndpoint = "https://secure.logmeinrescue.com/api/";
            string sSSOID = "123456";
            string sPwd = "secretPassword";
            string sCompany = "654321";

            HttpWebRequest oRequestAuthCode =
            (HttpWebRequest)WebRequest.Create(sEndpoint
            + "requestAuthCodeSSO.aspx?ssoid=" + sSSOID + "&pwd=" + sPwd + "&company="
            + sCompany);
            HttpWebResponse oRespAuthCode =
            (HttpWebResponse)oRequestAuthCode
            .GetResponse();

            string sRespAuthCode = new
            StreamReader(oRespAuthCode.GetResponseStream())
            .ReadToEnd();
            Response.Write(sRespAuthCode);
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
        }
    }
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sSSOID = "123456";
                string sPwd = "secretPassword";
                string sCompany = "654321";

                WebServiceClients.requestAuthCodeRet oRequestAuthCode =
                proxy.requestAuthCodeSSO
                (sEmail, sPwd, out sAuthCode);
                Response.Write(oRequestAuthCode + "<br />");
            }
        }
    }
}

```

```

        Response.Write("authCode: " + sAuthCode);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

requestPINCode

Generates a PIN code and sets the custom fields of a newly generated session. For information about starting a session via email, see *How to Start a Session via PIN Code* in the [Rescue Technician Console User Guide](#).



Important: You can request one PIN code per second.

Input Parameters

Element	Description
cfield0	Custom field 0. The maximum length is 64 characters. Optional.
cfield1	Custom field 1. The maximum length is 512 characters. Optional.
cfield2	Custom field 2. The maximum length is 512 characters. Optional.
cfield3	Custom field 3. The maximum length is 256 characters. Optional.
cfield4	Custom field 4. The maximum length is 64 characters. Optional.
cfield5	Custom field 5. The maximum length is 64 characters. Optional.
tracking0	Tracking field 0. The maximum length is 64 characters. Optional.
notechconsole	Specifies if running the Technician Console is required (0) or not (1). Optional, default is 0.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK PINCODE:123456
```

PINCODE: The 6-digit long PIN code to start a session

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Requesting a PIN code succeeded.
NOTLOGGEDIN	Requesting a PIN code failed because the current user is no longer logged in.
NOTTECHNICIAN	The PIN code was not requested by a technician.
NOTECHCONSOLERUNNING	The technician is offline and the Technician Console is not running, but it is possible to request a session for an offline technician.
OUTOFPINCODES	There are no more PIN codes that can be generated.
POLLRATEEXCEEDED	The specified interval of requesting a new PIN code is too short.
INVALID_SECRETAUTHCODE	The authentication code is incorrect.
USER_IS_DELETED	The user whose authorization code was specified is already deleted.
USER_DELETED_OR_DISABLED	The user whose authorization code was specified is already deleted or disabled.

Sample Code

The following are examples for using the `requestPINCode` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/requestPINCode.aspx?cfield0=custom1&cfield1=custom2&cfield2=custom3&cfield3=custom3&cfield4=custom4&cfield5=custom5&tracking0=track0&notechconsole=0&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/requestPINCode.aspx">  
  <input name="cfield0" value="custom0">  
  <input name="cfield1" value="custom1">  
  <input name="cfield2" value="custom2">  
  <input name="cfield3" value="custom3">  
  <input name="cfield4" value="custom4">  
  <input name="cfield5" value="custom5">  
  <input name="tracking0" value="track0">  
  <input name="notechconsole" value="0">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=requestPINCode>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//get account info
$sCField0 = "some customer";
$sCField1 = "first custom field";
$sCField2 = "second custom field";
$sCField3 = "third custom field";
$sCField4 = "fourth custom field";
$sCField5 = "fifth custom field";
$sTracking0 = $sCField0 . date("c"); //to ensure unique

//create params array for SOAP request
$requestPINCodeParams = array(
'sCField0' => $sCField0,
'sCField1' => $sCField1,
'sCField2' => $sCField2,
'sCField3' => $sCField3,
'sCField4' => $sCField4,
'sCField5' => $sCField5,
'sTracking0' => $sTracking0
);

$requestPINCodeResult = $soapclient->requestPINCode($requestPINCodeParams);

//print out the PIN Code full response
print_r("<b>requestPINCode full response.</b><br />");
print_r($requestPINCodeResult);
echo "<br /><br />"; //formatting

//show the PIN code only
$PINCode = $requestPINCodeResult["iPINCode"];

//print PIN Code
print_r("<b>PIN Code.</b><br />");
print_r($PINCode);

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for
    each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint +
        "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
        Response.Write("Cookie:" + "<br />");
        Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br />");
        Response.Write("Domain: " + cook.Domain + "<br />");
        Response.Write("Path: " + cook.Path + "<br />");
        Response.Write("Port: " + cook.Port + "<br />");
        Response.Write("Secure: " + cook.Secure + "<br />");

        Response.Write("When issued: " + cook.TimeStamp + "<br />");
        Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
        cook.Expired + "<br />");
        Response.Write("Don't save: " + cook.Discard + "<br />");
        Response.Write("Comment: " + cook.Comment + "<br />");
        Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
        Response.Write("Version: RFC " + cook.Version + "<br />");

        // Show the string representation of the cookie.
        Response.Write("String: " + cook.ToString());
    }*/

    //add cookies to cookie container
    CookieContainer sessioncookie = oReq.CookieContainer;
```

```

        //get the PINCode info
        HttpRequest oReqPINCode =
            (HttpRequest)System.Net.WebRequest.Create(sEndpoint
            + "requestPINCode.aspx");
        oReqPINCode.CookieContainer = sessioncookie;

        HttpResponse oRespPINCode = (HttpResponse)oReqPINCode.GetResponse();
        string sRespPINCode = new
            StreamReader(oRespPINCode.GetResponseStream()).ReadToEnd();
        Response.Write("requestPINCode result: " + sRespPINCode + "<br />"); //You
            can customize
            the response

    %>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API requestPINCode Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "";
                string sCField0 = "";
                string sCField1 = "";
                string sCField2 = "";
                string sCField3 = "";
                string sCField4 = "";
                string sCField5 = "";
                string sTracking0 = "";
                int iPIN;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
                sPwd);
            }
        }
    }
}

```



```

        Response.Write(oLogin + "<br />");

        WebServiceClients.requestPINCodeRet oRequestPINCode =
proxy.requestPINCode
(sCField0, sCField1, sCField2, sCField3, sCField4, sCField5, sTracking0,
true, sAuthCode,
out iPIN);

        Response.Write(oRequestPINCode + "<br />");

        Response.Write("PIN code: " + iPIN);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

requestPINCode - EU (v2)

Generates a PIN code and sets the custom fields of a session newly generated on EU Rescue servers. For information about starting a session via email, see *How to Start a Session via PIN Code* in the [Rescue Technician Console User Guide](#).



Important: You can request one PIN code per second.

Input Parameters

Element	Description
cfield0	Custom field 0. The maximum length is 64 characters. Optional.
cfield1	Custom field 1. The maximum length is 512 characters. Optional.
cfield2	Custom field 2. The maximum length is 512 characters. Optional.
cfield3	Custom field 3. The maximum length is 256 characters. Optional.
cfield4	Custom field 4. The maximum length is 64 characters. Optional.
cfield5	Custom field 5. The maximum length is 64 characters. Optional.
tracking0	Tracking field 0. The maximum length is 64 characters. Optional.
notechconsole	Specifies if running the Technician Console is required (0) or not (1). Optional, default is 0.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK PINCODE:012345
```

PINCODE: The 6-digit long PIN code to start a session

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Requesting a PIN code succeeded.
NOTLOGGEDIN	Requesting a PIN code failed because the current user is no longer logged in.
NOTTECHNICIAN	The PIN code was not requested by a technician.
NOTECHCONSOLERUNNING	The technician is offline and the Technician Console is not running, but it is possible to request a session for an offline technician.
OUTOFFPINCODES	There are no more PIN codes that can be generated.
POLLRATEEXCEEDED	The specified interval of requesting a new PIN code is too short.
INVALID_SECRETAUTHCODE	The authentication code is incorrect.
USER_IS_DELETED	The user whose authorization code was specified is already deleted.
USER_DELETED_OR_DISABLED	The user whose authorization code was specified is already deleted or disabled.

Sample Code

The following are examples for using the `requestPINCode_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.eu/API/requestPINCode_v2.aspx?cfield0=custom1&cfield1=custom2&cfield2=custom3&cfield3=custom3&cfield4=custom4&cfield5=custom5&tracking0=track0&notechconsole=0&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.eu/API/requestPINCode.aspx">  
  <input name="cfield0" value="custom0">  
  <input name="cfield1" value="custom1">  
  <input name="cfield2" value="custom2">  
  <input name="cfield3" value="custom3">
```

```
<input name="cfield4" value="custom4">
<input name="cfield5" value="custom5">
<input name="tracking0" value="track0">
<input name="notechconsole" value="0">
<input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=requestPINCodeV2>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//get account info
$sCField0 = "some customer";
$sCField1 = "first custom field";
$sCField2 = "second custom field";
$sCField3 = "third custom field";
$sCField4 = "fourth custom field";
$sCField5 = "fifth custom field";
$sTracking0 = $sCField0 . date("c"); //to ensure unique

//create params array for SOAP request
$requestPINCodeParams = array(
'sCField0' => $sCField0,
'sCField1' => $sCField1,
'sCField2' => $sCField2,
'sCField3' => $sCField3,
'sCField4' => $sCField4,
'sCField5' => $sCField5,
'sTracking0' => $sTracking0
);

$requestPINCodeResult = $soapclient->requestPINCode($requestPINCodeParams);

//print out the PIN Code full response
print_r("<b>requestPINCode full response.</b><br />");
print_r($requestPINCodeResult);
echo "<br /><br />"; //formatting
```

```
//show the PIN code only
$PINCode = $requestPINCodeResult["sPINCode"];

//print PIN Code
print_r("<b>PIN Code.</b><br />");
print_r($PINCode);

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.eu/API/"; //add
    actionName.aspx? for
    each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint +
    "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");

    Response.Write("When issued: " + cook.TimeStamp + "<br />");
    Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
    Response.Write("Don't save: " + cook.Discard + "<br />");
    Response.Write("Comment: " + cook.Comment + "<br />");
    Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
```

```

Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

    //get the PINCode info
HttpRequest oReqPINCode =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
    + "requestPINCode.aspx");
oReqPINCode.CookieContainer = sessioncookie;

HttpResponse oRespPINCode = (HttpResponse) oReqPINCode.GetResponse();
string sRespPINCode = new
    StreamReader(oRespPINCode.GetResponseStream()).ReadToEnd();
Response.Write("requestPINCode result: " + sRespPINCode + "<br />"); //You
    can customize
    the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API requestPINCode Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "";
                string sCField0 = "";
                string sCField1 = "";
            }
        }
    }
}

```

```

        string sCField2 = "";
        string sCField3 = "";
        string sCField4 = "";
        string sCField5 = "";
        string sTracking0 = "";
        string sPIN;

        WebserviceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

        Response.Write(oLogin + "<br />");

        WebserviceClients.requestPINCodeRet oRequestPINCode =
proxy.requestPINCode
(sCField0, sCField1, sCField2, sCField3, sCField4, sCField5, sTracking0,
true, sAuthCode,
out sPIN);

        Response.Write(oRequestPINCode + "<br />");

        Response.Write("PIN code: " + sPIN);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

requestPINCode_v3


Generates a PIN code and sets the custom fields of a newly generated session. The custom fields include the ID of the organizational entity (NodeId) for which the PIN code has been generated. For information about starting a session via email, see *How to Start a Session via PIN Code* in the [Rescue Technician Console User Guide](#).



Important: You can request one PIN code per second.

Input Parameters

Element	Description
cfield0	Custom field 0. The maximum length is 64 characters. Optional.
cfield1	Custom field 1. The maximum length is 512 characters. Optional.
cfield2	Custom field 2. The maximum length is 512 characters. Optional.
cfield3	Custom field 3. The maximum length is 256 characters. Optional.
cfield4	Custom field 4. The maximum length is 64 characters. Optional.
cfield5	Custom field 5. The maximum length is 64 characters. Optional.

Element	Description
tracking0	Tracking field 0. The maximum length is 64 characters. Optional.
notechconsole	Specifies if running the Technician Console is required (0) or not (1). Optional, default is 0.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.
NodeId	The ID (NodeID) of the technician for whom the PIN code has been generated.  Note: This applies to Administrators with a technician license, too.

Output

```
OK PINCODE:012345
```

PINCODE: The 6-digit long PIN code to start a session

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Requesting a PIN code succeeded.
NOTLOGGEDIN	Requesting a PIN code failed because the current user is no longer logged in.
NOTTECHNICIAN	The PIN code was not requested by a technician.
NOTECHCONSOLERUNNING	The technician is offline and the Technician Console is not running, but it is possible to request a session for an offline technician.
OUTOFPINCODES	There are no more PIN codes that can be generated.
POLLRATEEXCEEDED	The specified interval of requesting a new PIN code is too short.
INVALID_SECRETAUTHCODE	The authentication code is incorrect.
USER_IS_DELETED	The user whose authorization code was specified is already deleted.
USER_DELETED_OR_DISABLED	The user whose authorization code was specified is already deleted or disabled.

Sample Code

The following are examples for using the `requestPINCode_v3` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.eu/API/requestPINCode_v3.aspx?cfield0=custom1&cfield1=custom2&cfield2=custom3&cfield3=custom3&cfield4=custom4&cfield5=custom5&tracking0=track0&notechconsole=0&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.eu/API/requestPINCode.aspx">
  <input name="cfield0" value="custom0">
  <input name="cfield1" value="custom1">
  <input name="cfield2" value="custom2">
  <input name="cfield3" value="custom3">
  <input name="cfield4" value="custom4">
  <input name="cfield5" value="custom5">
  <input name="tracking0" value="track0">
  <input name="notechconsole" value="0">
  <input name="authcode" value="4ahx...80u0">
  <input name="NodeId" value="0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=requestPINCodeV3>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//get account info
$sCField0 = "some customer";
```



```

$sCField1 = "first custom field";
$sCField2 = "second custom field";
$sCField3 = "third custom field";
$sCField4 = "fourth custom field";
$sCField5 = "fifth custom field";
$sTracking0 = $sCField0 . date("c"); //to ensure unique

//create params array for SOAP request
$requestPINCodeParams = array(
'sCField0' => $sCField0,
'sCField1' => $sCField1,
'sCField2' => $sCField2,
'sCField3' => $sCField3,
'sCField4' => $sCField4,
'sCField5' => $sCField5,
'sTracking0' => $sTracking0
);

$requestPINCodeResult = $soapclient->requestPINCode($requestPINCodeParams);

//print out the PIN Code full response
print_r("<b>requestPINCode full response.</b><br />");
print_r($requestPINCodeResult);
echo "<br /><br />"; //formatting

//show the PIN code only
$PINCode = $requestPINCodeResult["sPINCode"];

//print PIN Code
print_r("<b>PIN Code.</b><br />");
print_r($PINCode);

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.eu/API/"; //add
    actionName.aspx? for
    each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint +
        "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

```

```

        //get the response
        HttpResponseMessage oResp = (HttpResponseMessage)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the PINCode info
HttpRequest oReqPINCode =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "requestPINCode.aspx");
oReqPINCode.CookieContainer = sessioncookie;

HttpResponseMessage oRespPINCode = (HttpResponseMessage)oReqPINCode.GetResponse();
string sRespPINCode = new
StreamReader(oRespPINCode.GetResponseStream()).ReadToEnd();
Response.Write("requestPINCode result: " + sRespPINCode + "<br />"); //You
can customize
the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
<title>Rescue API requestPINCode Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "";
                string sCField0 = "";
                string sCField1 = "";
                string sCField2 = "";
                string sCField3 = "";
                string sCField4 = "";
                string sCField5 = "";
                string sTracking0 = "";
                string sPIN;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.requestPINCodeRet oRequestPINCode =
proxy.requestPINCode
(sCField0, sCField1, sCField2, sCField3, sCField4, sCField5, sTracking0,
true, sAuthCode,
out sPIN);

                Response.Write(oRequestPINCode + "<br />");

                Response.Write("PIN code: " + sPIN);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

requestLensPINCode

Generates a PIN code and sets the custom fields of newly generated Rescue Lens sessions. For information about starting a Rescue Lens session, see *How to Start a Rescue Lens Session* in the [Rescue Lens User Guide](#).

Input Parameters



Important: You can request one PIN code per second.

Element	Description
cfield0	Custom field 0. The maximum length is 64 characters. Optional.
cfield1	Custom field 1. The maximum length is 512 characters. Optional.
cfield2	Custom field 2. The maximum length is 512 characters. Optional.
cfield3	Custom field 3. The maximum length is 256 characters. Optional.
cfield4	Custom field 4. The maximum length is 64 characters. Optional.
cfield5	Custom field 5. The maximum length is 64 characters. Optional.
tracking0	Tracking field 0. The maximum length is 64 characters. Optional.
notechconsole	Specifies if running the Technician Console is required (0) or not (1). Optional, default is 0.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK PINCODE:123456
```

PINCODE: The 6-digit long PIN code to start a session

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Requesting a PIN code succeeded.
NOTLOGGEDIN	Requesting a PIN code failed because the current user is no longer logged in.

Displayed Return Value	Description
NOTTECHNICIAN	The PIN code was not requested by a technician.
NOTECHCONSOLERUNNING	The technician is offline and the Technician Console is not running, but it is possible to request a session for an offline technician.
OUTOFPINCOCES	There are no more PIN codes that can be generated.
POLLRATEEXCEEDED	The specified interval of requesting a new PIN code is too short.
INVALID_SECRETAUTHCODE	The authentication code is incorrect.
USER_IS_DELETED	The user whose authorization code was specified is already deleted.
USER_DELETED_OR_DISABLED	The user whose authorization code was specified is already deleted or disabled.
NO_ACCESS_TO_LENS	Requesting a PIN code failed because the current user does not have permission to access Rescue Lens.

Sample Code

The following are examples for using the `requestLensPINCode` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/requestLensPINCode.aspx?
cfield0=custom1
&cfield1=custom2&cfield2=custom3&cfield3=custom3&cfield4=custom4&
cfield5=custom5&tracking0=track0&notechconsole=0&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
requestLensPINCode.aspx">
  <input name="cfield0" value="custom0">
  <input name="cfield1" value="custom1">
  <input name="cfield2" value="custom2">
  <input name="cfield3" value="custom3">
  <input name="cfield4" value="custom4">
  <input name="cfield5" value="custom5">
  <input name="tracking0" value="track0">
  <input name="notechconsole" value="0">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=requestLensPINCode>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//get account info
$sCField0 = "some customer";
$sCField1 = "first custom field";
$sCField2 = "second custom field";
$sCField3 = "third custom field";
$sCField4 = "fourth custom field";
$sCField5 = "fifth custom field";
$sTracking0 = $sCField0 . date("c"); //to ensure unique

//create params array for SOAP request
$requestLensPINCodeParams = array(
'sCField0' => $sCField0,
'sCField1' => $sCField1,
'sCField2' => $sCField2,
'sCField3' => $sCField3,
'sCField4' => $sCField4,
'sCField5' => $sCField5,
'sTracking0' => $sTracking0
);

$requestLensPINCodeResult = $soapclient-
>requestLensPINCode($requestLensPINCodeParams);

//print out the PIN Code full response
print_r("<b>requestLensPINCode full response.</b><br />");
print_r($requestLensPINCodeResult);
echo "<br /><br />"; //formatting

//show the PIN code only
$PINCode = $requestLensPINCodeResult["iPINCode"];

//print PIN Code
print_r("<b>PIN Code.</b><br />");
print_r($PINCode);
```

?>

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
actionName.aspx? for
each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";

    //set up the request
    HttpWebRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint +
"login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
```

```

CookieContainer sessioncookie = oReq.CookieContainer;

    //get the PINcode info
HttpRequest oReqPINcode =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
    + "requestLensPINcode.aspx");
oReqPINcode.CookieContainer = sessioncookie;

HttpResponse oRespPINcode = (HttpResponse)oReqPINcode.GetResponse();
string sRespPINcode = new
    StreamReader(oRespPINcode.GetResponseStream()).ReadToEnd();
Response.Write("requestLensPINcode result: " + sRespPINcode + "<br />");
    //You can customize
    the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API requestLensPINcode Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "";
                string sCField0 = "";
                string sCField1 = "";
                string sCField2 = "";
                string sCField3 = "";
                string sCField4 = "";
                string sCField5 = "";
                string sTracking0 = "";
                int iPIN;
            }
        }
    }
}

```



```

WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
Response.Write(oLogin + "<br />");

WebServiceClients.requestLensPINCodeRet oRequestLensPINCode
= proxy.requestLensPINCode
(sCField0, sCField1, sCField2, sCField3, sCField4, sCField5, sTracking0,
true, sAuthCode,
out iPIN);
Response.Write(oRequestLensPINCode + "<br />");

Response.Write("PIN code: " + iPIN);
}
catch (Exception ex)
{
Response.Write(ex.Message);
}
}
}
}

```

requestLensPINCode_v2

Generates a PIN code and sets the custom fields of newly generated Rescue Lens sessions. For information about starting a Rescue Lens session, see *How to Start a Rescue Lens Session* in the [Rescue Lens User Guide](#).

Input Parameters



Important: You can request one PIN code per second.

Element	Description
cfield0	Custom field 0. The maximum length is 64 characters. Optional.
cfield1	Custom field 1. The maximum length is 512 characters. Optional.
cfield2	Custom field 2. The maximum length is 512 characters. Optional.
cfield3	Custom field 3. The maximum length is 256 characters. Optional.
cfield4	Custom field 4. The maximum length is 64 characters. Optional.
cfield5	Custom field 5. The maximum length is 64 characters. Optional.
tracking0	Tracking field 0. The maximum length is 64 characters. Optional.
notechconsole	Specifies if running the Technician Console is required (0) or not (1). Optional, default is 0.

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK PINCODE:123456
```

PINCODE: The 6-digit long PIN code to start a session

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Requesting a PIN code succeeded.
NOTLOGGEDIN	Requesting a PIN code failed because the current user is no longer logged in.
NOTTECHNICIAN	The PIN code was not requested by a technician.
NOTECHCONSOLERUNNING	The technician is offline and the Technician Console is not running, but it is possible to request a session for an offline technician.
OUTOFPINCODES	There are no more PIN codes that can be generated.
POLLRATEEXCEEDED	The specified interval of requesting a new PIN code is too short.
INVALID_SECRETAUTHCODE	The authentication code is incorrect.
USER_IS_DELETED	The user whose authorization code was specified is already deleted.
USER_DELETED_OR_DISABLED	The user whose authorization code was specified is already deleted or disabled.
NO_ACCESS_TO_LENS	Requesting a PIN code failed because the current user does not have permission to access Rescue Lens.

Sample Code

The following are examples for using the `requestLensPINCode_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/requestLensPINCode_v2.aspx?
cfield0=custom1
```

```
&cfield1=custom2&cfield2=custom3&cfield3=custom3&cfield4=custom4&
cfield5=custom5&tracking0=track0&notechconsole=0&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
requestLensPINCode_v2.aspx">
  <input name="cfield0" value="custom0">
  <input name="cfield1" value="custom1">
  <input name="cfield2" value="custom2">
  <input name="cfield3" value="custom3">
  <input name="cfield4" value="custom4">
  <input name="cfield5" value="custom5">
  <input name="tracking0" value="track0">
  <input name="notechconsole" value="0">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit:<https://secure.logmeinrescue.com/api/API.asmx?op=requestLensPINCodeV2>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//get account info
$sCField0 = "some customer";
$sCField1 = "first custom field";
$sCField2 = "second custom field";
$sCField3 = "third custom field";
$sCField4 = "fourth custom field";
$sCField5 = "fifth custom field";
$sTracking0 = $sCField0 . date("c"); //to ensure unique

//create params array for SOAP request
$requestLensPINCode_v2Params = array(
'sCField0' => $sCField0,
'sCField1' => $sCField1,
'sCField2' => $sCField2,
'sCField3' => $sCField3,
```

```

'sCField4' => $sCField4,
'sCField5' => $sCField5,
'sTracking0' => $sTracking0
);

$requestLensPINCode_v2Result = $soapclient-
>requestLensPINCode($requestLensPINCode_v2Params);

//print out the PIN Code full response
print_r("<b>requestLensPINCode_v2 full response.</b><br />");
print_r($requestLensPINCodeResult_v2);
echo "<br /><br />"; //formatting

//show the PIN code only
$PINCode = $requestLensPINCode_v2Result["iPINCode"];

//print PIN Code
print_r("<b>PIN Code.</b><br />");
print_r($PINCode);

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for
    each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint +
    "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
    Response.Write("Cookie:" + "<br />");
    }
    */

```

```

Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

    //get the PINCode info
HttpRequest oReqPINCode =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
    + "requestLensPINCode_v2.aspx");
oReqPINCode.CookieContainer = sessioncookie;

HttpResponse oRespPINCode = (HttpResponse)oReqPINCode.GetResponse();
string sRespPINCode = new
    StreamReader(oRespPINCode.GetResponseStream()).ReadToEnd();
Response.Write("requestLensPINCode_v2 result: " + sRespPINCode + "<br />");
    //You can customize
    the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API requestLensPINCode_v2 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page

```

```

{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            WebServiceClients.API proxy = new WebServiceClients.API();
            proxy.CookieContainer = new CookieContainer();

            string sEmail = "some@email.com";
            string sPwd = "secretPassword";
            string sAuthCode = "";
            string sCField0 = "";
            string sCField1 = "";
            string sCField2 = "";
            string sCField3 = "";
            string sCField4 = "";
            string sCField5 = "";
            string sTracking0 = "";
            int iPIN;

            WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
            Response.Write(oLogin + "<br />");

            WebServiceClients.requestLensPINCodeRet oRequestLensPINCode
= proxy.requestLensPINCode_v2
(sCField0, sCField1, sCField2, sCField3, sCField4, sCField5, sTracking0,
true, sAuthCode,
out iPIN);
            Response.Write(oRequestLensPINCode + "<br />");

            Response.Write("PIN code: " + iPIN);
        }
        catch (Exception ex)
        {
            Response.Write(ex.Message);
        }
    }
}
}

```

requestLensPINCode_v3

Generates a PIN code and sends it in an invitation SMS to the phone number provided. Sets the custom fields of newly generated Rescue Lens sessions. For information about starting a Rescue Lens session, see *How to Start a Rescue Lens Session* in the [Rescue Lens User Guide](#), .

Input Parameters



Important: You can request one PIN code per second.

Element	Description
cfield0	Custom field 0. The maximum length is 64 characters. Optional.

Element	Description
cfield1	Custom field 1. The maximum length is 512 characters. Optional.
cfield2	Custom field 2. The maximum length is 512 characters. Optional.
cfield3	Custom field 3. The maximum length is 256 characters. Optional.
cfield4	Custom field 4. The maximum length is 64 characters. Optional.
cfield5	Custom field 5. The maximum length is 64 characters. Optional.
tracking0	Tracking field 0. The maximum length is 64 characters. Optional.
notechconsole	Specifies if running the Technician Console is required (0) or not (1). Optional, default is 0.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.
PhoneNumber	The phone number to which the invitation SMS containing the PIN Code is to be sent. Optional.
LangCode	The code of the language in which the invitation SMS is to be sent. Possible values: <ul style="list-style-type: none"> • de for German • en for English • es For Spanish • fr for French • nl for Dutch • pt-br for Portuguese (Brazilian) Optional, default is en.

Output

```
OK PINCODE:123456
```

PINCODE: The 6-digit long PIN code to start a session

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Requesting a PIN code succeeded.
NOTLOGGEDIN	Requesting a PIN code failed because the current user is no longer logged in.

Displayed Return Value	Description
NOTTECHNICIAN	The PIN code was not requested by a technician.
NOTECHCONSOLERUNNING	The technician is offline and the Technician Console is not running, but it is possible to request a session for an offline technician.
OUTOFFPINCODES	There are no more PIN codes that can be generated.
POLLRATEEXCEEDED	The specified interval of requesting a new PIN code is too short.
INVALID_SECRETAUTHCODE	The authentication code is incorrect.
USER_IS_DELETED	The user whose authorization code was specified is already deleted.
USER_DELETED_OR_DISABLED	The user whose authorization code was specified is already deleted or disabled.
NO_ACCESS_TO_LENS	Requesting a PIN code failed because the current user does not have permission to access Rescue Lens.
INVALID_PHONENUMBER	The phone number to which the SMS was to be sent is not valid.
SMS_CANNOT_BE_SENT	The SMS cannot be delivered to the recipient.
NO_MORE_SMS	The Rescue account has used up all of its allocated SMSs.

Sample Code

The following are examples for using the `requestLensPINCode_v3` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/requestLensPINCodev3.aspx?
cfield0=custom1
&cfield1=custom2&cfield2=custom3&cfield3=custom3&cfield4=custom4&
cfield5=custom5&tracking0=track0&notechconsole=0&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
requestLensPINCodev3.aspx">
  <input name="cfield0" value="custom0">
  <input name="cfield1" value="custom1">
  <input name="cfield2" value="custom2">
  <input name="cfield3" value="custom3">
  <input name="cfield4" value="custom4">
  <input name="cfield5" value="custom5">
  <input name="tracking0" value="track0">
```



```
<input name="notechconsole" value="0">
<input name="authcode" value="4ahx...80u0">
<input name="PhoneNumber" value="0">
<input name="LangCode" value="0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=requestLensPINCodeV3>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//get account info
$sCField0 = "some customer";
$sCField1 = "first custom field";
$sCField2 = "second custom field";
$sCField3 = "third custom field";
$sCField4 = "fourth custom field";
$sCField5 = "fifth custom field";
$sTracking0 = $sCField0 . date("c"); //to ensure unique

//create params array for SOAP request
$requestLensPINCode_v3Params = array(
'sCField0' => $sCField0,
'sCField1' => $sCField1,
'sCField2' => $sCField2,
'sCField3' => $sCField3,
'sCField4' => $sCField4,
'sCField5' => $sCField5,
'sTracking0' => $sTracking0
);

$requestLensPINCode_v3Result = $soapclient-
>requestLensPINCode($requestLensPINCode_v3Params);

//print out the PIN Code full response
print_r("<b>requestLensPINCode_v3 full response.</b><br />");
print_r($requestLensPINCodeResult_v3);
echo "<br /><br />"; //formatting
```

```
//show the PIN code only
$PINCode = $requestLensPINCode_v3Result["iPINCode"];

//print PIN Code
print_r("<b>PIN Code.</b><br />");
print_r($PINCode);

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for
    each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint +
    "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");

    Response.Write("When issued: " + cook.TimeStamp + "<br />");
    Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
    Response.Write("Don't save: " + cook.Discard + "<br />");
    Response.Write("Comment: " + cook.Comment + "<br />");
    Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
```

```

Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

    //get the PINCode info
HttpRequest oReqPINCode =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
    + "requestLensPINCode_v3.aspx");
oReqPINCode.CookieContainer = sessioncookie;

HttpResponse oRespPINCode = (HttpResponse) oReqPINCode.GetResponse();
string sRespPINCode = new
    StreamReader(oRespPINCode.GetResponseStream()).ReadToEnd();
Response.Write("requestLensPINCode_v3 result: " + sRespPINCode + "<br />");
    //You can customize
    the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API requestLensPINCode_v3 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "";
                string sCField0 = "";
                string sCField1 = "";
            }
        }
    }
}

```

```

        string sCField2 = "";
        string sCField3 = "";
        string sCField4 = "";
        string sCField5 = "";
        string sTracking0 = "";
        int iPIN;

        WebserviceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

        Response.Write(oLogin + "<br />");

        WebserviceClients.requestLensPINCodeRet oRequestLensPINCode
= proxy.requestLensPINCode_v3
(sCField0, sCField1, sCField2, sCField3, sCField4, sCField5, sTracking0,
true, sAuthCode,
out iPIN);

        Response.Write(oRequestLensPINCode + "<br />");

        Response.Write("PIN code: " + iPIN);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

requestLensPINCode_v4


Generates a PIN code and sends it in an invitation SMS to the phone number provided. Sets the custom fields of newly generated Rescue Lens sessions including the ID of the technician (NodeID) for whom the PIN code has been generated. For information about starting a Rescue Lens session, see *How to Start a Rescue Lens Session* in the [Rescue Lens User Guide](#), .

Input Parameters



Important: You can request one PIN code per second.

Element	Description
cfield0	Custom field 0. The maximum length is 64 characters. Optional.
cfield1	Custom field 1. The maximum length is 512 characters. Optional.
cfield2	Custom field 2. The maximum length is 512 characters. Optional.
cfield3	Custom field 3. The maximum length is 256 characters. Optional.
cfield4	Custom field 4. The maximum length is 64 characters. Optional.
cfield5	Custom field 5. The maximum length is 64 characters. Optional.

Element	Description
tracking0	Tracking field 0. The maximum length is 64 characters. Optional.
notechconsole	Specifies if running the Technician Console is required (0) or not (1). Optional, default is 0.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.
PhoneNumber	The phone number to which the invitation SMS containing the PIN Code is to be sent. Optional.
LangCode	The code of the language in which the invitation SMS is to be sent. Possible values: <ul style="list-style-type: none"> • de for German • en for English • es For Spanish • fr for French • nl for Dutch • pt-br for Portuguese (Brazilian) Optional, default is en.
NodeId	The ID (NodeID) of the technician for whom the PIN code has been generated.  Note: This applies to Administrators with a technician license, too.

Output

```
OK PINCODE:123456
```

```
SMS_RESULT: sendInvitationForMobileSession_OK
```

PINCODE: The 6-digit long PIN code to start a session

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Requesting a PIN code succeeded.
NOTLOGGEDIN	Requesting a PIN code failed because the current user is no longer logged in.
NOTTECHNICIAN	The PIN code was not requested by a technician.

Displayed Return Value	Description
NOTECHCONSOLERUNNING	The technician is offline and the Technician Console is not running, but it is possible to request a session for an offline technician.
OUTOFPINCODES	There are no more PIN codes that can be generated.
POLLRATEEXCEEDED	The specified interval of requesting a new PIN code is too short.
INVALID_SECRETAUTHCODE	The authentication code is incorrect.
USER_IS_DELETED	The user whose authorization code was specified is already deleted.
USER_DELETED_OR_DISABLED	The user whose authorization code was specified is already deleted or disabled.
NO_ACCESS_TO_LENS	Requesting a PIN code failed because the current user does not have permission to access Rescue Lens.
INVALID_PHONENUMBER	The phone number to which the SMS was to be sent is not valid.
SMS_CANNOT_BE_SENT	The SMS cannot be delivered to the recipient.
NO_MORE_SMS	The Rescue account has used up all of its allocated SMSs.

Sample Code

The following are examples for using the `requestLensPINCode_v4` method that you can call in your environment.

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=requestLensPINCodeV4>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);
```

```

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//get account info
$sCField0 = "some customer";
$sCField1 = "first custom field";
$sCField2 = "second custom field";
$sCField3 = "third custom field";
$sCField4 = "fourth custom field";
$sCField5 = "fifth custom field";
$sTracking0 = $sCField0 . date("c"); //to ensure unique

//create params array for SOAP request
$requestLensPINCode_v4Params = array(
'sCField0' => $sCField0,
'sCField1' => $sCField1,
'sCField2' => $sCField2,
'sCField3' => $sCField3,
'sCField4' => $sCField4,
'sCField5' => $sCField5,
'sTracking0' => $sTracking0
);

$requestLensPINCode_v4Result = $soapclient-
>requestLensPINCode($requestLensPINCode_v4Params);

//print out the PIN Code full response
print_r("<b>requestLensPINCode_v4 full response.</b><br />");
print_r($requestLensPINCodeResult_v4);
echo "<br /><br />"; //formatting

//show the PIN code only
$PINCode = $requestLensPINCode_v4Result["iPINCode"];

//print PIN Code
print_r("<b>PIN Code.</b><br />");
print_r($PINCode);

?>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try

```


Session Data	Description
TechName	Technician name (as defined on the Organization tab)
TechEmail	Technician email (as defined on the Organization tab)
TechDescr	Technician description (as defined on the Organization tab)
CField0	Value returned for the Name field (as defined on the Global Settings tab in the Administration Center)
CField1	Value returned for Custom field 1 (as defined on the Global Settings tab)
CField2	Value returned for Custom field 2 (as defined on the Global Settings tab)
CField3	Value returned for Custom field 3 (as defined on the Global Settings tab)
CField4	Value returned for Custom field 4 (as defined on the Global Settings tab)
CField5	Value returned for Custom field 5 (as defined on the Global Settings tab)
Tracking0	Value returned for the Tracking field; typically used for mapping Rescue sessions to a CRM. If a customer can ask for assistance from multiple websites, the <code>tracking0</code> field may contain, for example, the identifier of the website that the customer visited.
ChatLog	Transcript of all chat held since the previous post.
Notes	Notes saved by the technician. This is always empty at the beginning of the session.
WaitingTime	From the beginning of Waiting status until session start (Active status) in seconds
PickupTime	The exact time when the session entered Active status (UTC)
ClosingTime	The exact time when the session entered Closed or Timed Out status (UTC)
WorkTime	Actual Technician Console utilization time during the session (until the post) in seconds. This is always empty at the beginning of the session.
LastActionTime	The exact time of the last action taken by the technician in the Technician Console (UTC). This is always empty at the beginning of the session.

Session Data	Description
Transmitted	Amount of data transmitted during the session (until the post) in bytes. This is always empty at the beginning of the session.
Platform	The platform of the customer device

The following data is saved at the end of a Rescue session in addition to the data that is saved at the beginning of the session:

Session Data	Description
TSurvey0..9	Value returned for Technician Survey Question 1 to 10 (as defined on the Settings tab).

How it Works

- The Rescue technician starts a support session
- At the beginning and/or end of a session, the session data are transferred via HTTP Post or XML to the specified URL
- Your script processes the data as specified in your code

HTTP Post based

This method submits the URL with the POST variables on the end. This is the same as submitting an HTML form. The variables use the same naming convention as the XML format.

```
https://example.com/script.aspx?
SessionID=[...] &TechID=[...] &TechSSOID=[...]
&TechDescr=[...] &CField0=[...] &CField1=[...] &CField2=[...] &CField3=[...]
&CField4=[...] &CField5=[...] &Tracking0=[...] &ChatLog=[...] &
Notes=[...] &WaitingTime=[...] &PickupTime=[...] &ClosingTime=[...] &
WorkTime=[...] &LastActionTime=[...] &Transmitted=[...] &TSurvey0=[...] &
TSurvey1=[...] &TSurvey2=[...] &TSurvey3=[...] &TSurvey4=[...] &TSurvey5=[...] &
TSurvey6=[...] &TSurvey7=[...] &TSurvey8=[...] &TSurvey9=[...]
```



Note: The HTTP POST option is actual POST data. Using the GET method will not work properly.

Session Management

These methods enable you to manage technician sessions.



Note: The `authcode` parameter enables you to use these methods without logging in to Rescue.

getSession

Retrieves the current sessions of any hierarchy node (technician or channel). You can see the list of sessions on the **Sessions** tab of the Rescue Administration Center.

Input Parameters

Element	Description
node	The sessions of this node are listed. This is an optional element. If the ID of a node is not specified, it returns all existing sessions for all technicians and channels. Required.
noderef	The reference of the node, which can be NODE or CHANNEL. Optional, default is NODE.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK ID|Status|Entry ID|Entry|Technician ID|Technician|Start Time|Waiting  
Time|Custom Field 0|  
Custom Field 1|Custom Field 2|Language|Transferred To|Transferred Comment|  
Is Lead Technician|  
HandingOff|1663112|Waiting|0||337366|John Doe|2/16/2012 9:00 AM|13|  
Customer|||en|||yes||
```

The list that the `getSession` method retrieves contains information about a session in a table. This table has the following columns:

- ID: Session ID

- Status: Session status

Possible values:

- Aborted: technician was deleted or disabled
- Active
- Closed by active customer
- Closed by customer
- Closed by technician
- Closed by waiting customer
- Connecting
- Disconnected
- Not available
- Offline
- On Hold
- Rebooted
- Rebooting
- Reconnecting
- Timed out
- Timed out: closed by technician
- Transferred
- Transferring
- Waiting
- Entry ID: Channel's ID
- Entry: Channel's name
- Technician ID: Technician's ID
- Technician: Technician's name
- Start Time: Start time of the session
- Waiting Time: Time elapsed since the last session status change
- Custom Field 0-2: Custom fields that the Technician fills in when he initiates a session
- Language: The two-digit language code of the customer's operation system
- Transferred To: The recipient Technician of a transferred session
- Transferred Comment: Comment to the recipient Technician when a session transferred
- Is Lead Technician: Denotes whether the technician of the session is the lead technician
- HandingOff: Denotes whether the session was handed over to the technician

Session List Elements

The table containing the session list has the following elements:

Session list element	Content
<sessions>	<session_header> <session_rows>
<session_header>	<field_value> <delimiter> <session_header> <field_value> <new_line_char>
<session_rows>	<session_row> <session_row> <session_row>
<session_row>	<field_value> <delimiter> <session_row> <field_value> <new_line_char>
<field_value>	<char> <field_value> <char>
<char>	[.]

Session list element	Content
<new_line_char>	'\n'

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving session information succeeded.
NOTLOGGEDIN	Retrieving session information failed because the current is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing node.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getSession` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getSession.aspx?
node=337366&noderef=NODE
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getSession.aspx">
  <input name="node" value=337366>
  <input name="noderef" value=NODE>
  <input name=authcode value=4ahx...80u0>
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getSession>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");
```

```

//define variables
$email = "someEmail@domain.com";
$password = "secretPassword";
$iNodeID = "337366"; //define this with your own data
$eNodeRef = "NODE"; //define this appropriately based on your data
$sAuthCode = "4ahx...80u0";

//define parameters
$loginparams = array (
'sEmail' => $email,
'sPassword' => $password);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//switch to XML for easier formatting of output
$output = array(
'eOutput' => "XML"
);

$outputResponse = $soapclient->setOutput($output);
print_r("<b>setOutput Full response.</b> <br />");
print_r($outputResponse);
echo "<br /><br />"; //formatting

//set up array for getSession
$sessionparams = array(
'iNodeID' => $iNodeID,
'eNodeRef' => $eNodeRef,
'sAuthCode' => $sAuthCode
);

//get session(s)
$sessionresult = $soapclient->getSession($sessionparams);

//show the full getSession response
print_r("<b>getSession Text full response.</b><br />");
print_r($sessionresult);

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
for each action called

```

```

string sEmail = "some@email.com";
string sPwd = "secretPassword";
string sAuthCode = "4ahx...80u0";

//set up the request
HttpWebRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

//create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

//get the response
HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Session info
//remember to tag your intended node to the URL
int iNode = "";
string eNodeRef = "NODE";

HttpWebRequest oReqSession =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "getSession.aspx?node=" + sNode + "&noderef=" + eNodeRef + "&authcode="
+ sAuthCode);
oReqSession.CookieContainer = sessioncookie;

HttpWebResponse oRespSession = (HttpWebResponse)oReqSession.GetResponse();
string sRespSession = new
StreamReader(oRespSession.GetResponseStream()).ReadToEnd();

```

```

Response.Write("getSession result: " + sRespSession + "<br />"); //You can
    customize
    the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getSession Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                int iNodeID = 337366;
                NODE_REF eNodeRef = NODE_REF.NODE;
                SESSION[] aSessions;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.getSessionRet oGetSession =
proxy.getSession(iNodeID,
eNodeRef, sAuthCode, out aSessions);
                Response.Write(oGetSession + "<br />");

                foreach(SESSION s in aSessions)
                {
                    Response.Write("Session ID: " + s.iID + " Technician: "
+ s.sTechnician);
                }
            }
            catch (Exception ex)

```



```

        {
            Response.Write(ex.Message);
        }
    }
}

```

getSession_v2

Retrieves the current sessions of any hierarchy node (technician or channel). You can see the list of sessions on the **Sessions** tab of the Rescue Administration Center.

Changes in Version 2 of getSession

The following session properties have been introduced in the `getSession_v2` method:

- IsLead** Denotes whether the technician of the session is the lead technician.
- HandingOff** Denotes whether the session was handed over to the technician.

Input Parameters

Element	Description
node	The sessions of this node are listed. Required.
noderef	The reference of the node, which can be NODE or CHANNEL. Optional, default is NODE.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```

OK ID|Status|Entry ID|Entry|Technician ID|Technician|Start Time|Waiting
Time|Custom Field 0|
Custom Field 1|Custom Field 2|Language|Transferred To|Transferred Comment|
Is Lead Technician|
HandingOff|1663112|Waiting|0||337366|John Doe|2/16/2012 9:00 AM|13|
Customer|||en|||yes||

```

The list that the `getSession` method retrieves contains information about a session in a table. This table has the following columns:

- ID: Session ID

- Status: Session status

Possible values:

- Aborted: technician was deleted or disabled
- Active
- Closed by active customer
- Closed by customer
- Closed by technician
- Closed by waiting customer
- Connecting
- Disconnected
- Not available
- Offline
- On Hold
- Rebooted
- Rebooting
- Reconnecting
- Timed out
- Timed out: closed by technician
- Transferred
- Transferring
- Waiting
- Entry ID: Channel's ID
- Entry: Channel's name
- Technician ID: Technician's ID
- Technician: Technician's name
- Start Time: Start time of the session
- Waiting Time: Time elapsed since the last session status change
- Custom Field 0-2: Custom fields that the Technician fills in when he initiates a session
- Language: The two-digit language code of the customer's operation system
- Transferred To: The recipient Technician of a transferred session
- Transferred Comment: Comment to the recipient Technician when a session transferred
- Is Lead Technician: Denotes whether the technician of the session is the lead technician
- HandingOff: Denotes whether the session was handed over to the technician

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving session information succeeded.
NOTLOGGEDIN	Retrieving session information failed because the current is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing node.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getSession_v2` method that you can call in your environment.

C# with SOAP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                int iNodeID = 337366;
                NODE_REF eNodeRef = NODE_REF.NODE;
                SESSION_V2[] aSessions;

                //Login
                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.getSessionRet oGetSessionV2 =
proxy.getSession_V2(iNodeID, eNodeRef, sAuthCode, out aSessions);
                Response.Write(oGetSessionV2 + "<br />");

                foreach (SESSION_V2 s in aSessions)
                {
                    Response.Write("Session ID: " + s.iID + " Technician: "
+ s.sTechnician);
                }
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

getSession_v3

Retrieves the current sessions of any hierarchy node (technician or channel). You can see the list of sessions on the **Sessions** tab of the Rescue Administration Center.

Changes in Version 3 of getSession

The following fields have been introduced in the `getSession_v3` method:

sCustomField3, The content of custom fields 3, 4, and 5.
sCustomField4,
and
sCustomField5

The following session statuses have been introduced in the `getSession_v3` method:

SESSION_UACCESS_EXPIRED_WASPICKEDUPBEFORE unattended access expired and it was picked up by the technician.

SESSION_UACCESS_EXPIRED_NOTPICKEDUPBEFORE unattended access expired without the technician picking it up.

SESSION_UACCESS_REVOKEDBYCUSTOMER_WASPICKEDUPBEFORE customer revoked the unattended access right from the technician who picked up the computer.

SESSION_UACCESS_REVOKEDBYCUSTOMER_NOTPICKEDUPBEFORE customer revoked the unattended access right from the technician who did not pick up the computer.

SESSION_UACCESS_REVOKEDBYADMIN_WASPICKEDUPBEFORE admin revoked the unattended access right from the technician who picked up the computer.

SESSION_UACCESS_REVOKEDBYADMIN_NOTPICKEDUPBEFORE admin revoked the unattended access right from the technician who did not pick up the computer.

SESSION_TERMSANDCONDITIONS_DECLINED terms and conditions declined by the customer by not accepting the terms and conditions.

SESSION_TIMEOUT the session timed out or it was closed by the customer.

TECHCONNECTIONLOSTOR
SESSIONWASNOTCLOSEDBYTECH

Input Parameters

Element	Description
node	The sessions of this node are listed. Required.
noderef	The reference of the node, which can be NODE or CHANNEL. Optional, default is NODE.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK ID|Status|Entry ID|Entry|Technician ID|Technician|Start Time|Waiting  
Time|Custom Field 0|  
Custom Field 1|Custom Field 2|Custom Field 3|Custom Field 4|Custom Field 5|  
Language|  
Transferred To|Transferred Comment|Is Lead Technician|HandingOff|  
1663356|Active|0||337366|John Doe|2/16/2012 4:49 PM|104|Customer|||||en|||  
yes||
```

The list that the `getSession_v3` method retrieves contains information about a session in a table. This table has the following columns:

- ID: Session ID
- Status: Session status

Possible values:

- Aborted: technician was deleted or disabled
- Active
- Closed by active customer
- Closed by customer
- Closed by technician
- Closed by waiting customer
- Connecting
- Declined by customer
- Disconnected
- Expired
- Not available
- Offline
- On Hold
- Rebooted
- Rebooting
- Reconnecting
- Revoked
- Timed out
- Timed out: closed by technician
- Transferred
- Transferring
- Waiting
- Entry ID: Channel's ID
- Entry: Channel's name
- Technician ID: Technician's ID
- Technician: Technician's name
- Start Time: Start time of the session
- Waiting Time: Time elapsed since the last session status change
- Custom Field 0-5: Custom fields that the Technician fills in when he initiates a session
- Language: The two-digit language code of the customer's operation system
- Transferred To: The recipient Technician of a transferred session
- Transferred Comment: Comment to the recipient Technician when a session transferred
- Is Lead Technician
- HandingOff

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving session information succeeded.
NOTLOGGEDIN	Retrieving session information failed because the current is no longer logged in.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing node.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getSession_v3` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getSession_v3.aspx?node=337366
&noderef=NODE&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getSession_v3.aspx">
  <input name="node" value="337366">
  <input name="noderef" value="NODE">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getSession_V3.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define variables
$email = "someEmail@domain.com";
$password = "secretPassword";
$nodeID = "337366"; //define this with your own data
$nodeRef = "NODE"; //define this appropriately based on your data
$authCode = "4ahx...80u0";
```

```

//define parameters
$loginparams = array (
'sEmail' => $sEmail,
'sPassword' => $sPassword);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//switch to XML for easier formatting of output
$output = array(
'eOutput' => "XML"
);

$outputResponse = $soapclient->setOutput($output);
print_r("<b>setOutput Full response.</b> <br />");
print_r($outputResponse);
echo "<br /><br />"; //formatting

//set up array for getSession_v3
$sessionparams = array(
'iNodeID' => $iNodeID,
'eNodeRef' => $eNodeRef,
'sAuthCode' => $sAuthCode
);

//get session(s)
$sessionresult = $soapclient->getSession_v3($sessionparams);

//show the full getSession response
print_r("<b>getSession Text full response.</b><br />");
print_r($sessionresult);

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

    //set up the request

```

```

    HttpWebRequest oReq =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
    + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Session info
//remember to tag your intended node to the URL
int iNode = "";
string eNodeRef = "NODE";

HttpWebRequest oReqSession =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
    + "getSession_v3.aspx?node=" + sNode + "&noderef=" + eNodeRef +
    "&authcode=" + sAuthCode);
oReqSession.CookieContainer = sessioncookie;

HttpWebResponse oRespSession = (HttpWebResponse) oReqSession.GetResponse();
string sRespSession = new
    StreamReader(oRespSession.GetResponseStream()).ReadToEnd();
Response.Write("getSession result: " + sRespSession + "<br />"); //You can
customize
the response

```



```

%>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Rescue API getSession Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                int iNodeID = 337366;
                NODE_REF eNodeRef = NODE_REF.NODE;
                SESSION_V3[] aSessions;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.getSessionRet oGetSession =
proxy.getSession_v3(iNodeID,
eNodeRef, sAuthCode, out aSessions);
                Response.Write(oGetSession + "<br />");

                foreach(SESSION s in aSessions)
                {
                    Response.Write("Session ID: " + s.iID + " Technician: "
+ s.sTechnician);
                }
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

```
}  
}
```

startSession

An administrator can start a session of a technician. You can start a session from the **Sessions** tab of the Rescue Administration Center.

Input Parameters

Element	Description
session	The ID of the target session. Required.
node	The session is picked up by this node. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Starting a session succeeded.
NOTLOGGEDINASADMIN	Starting a session failed because the administrator is not logged in.
ACTIONFAILED	Starting a session failed because the technician is offline and the Technician Console is not running.
INVALIDPARAM_SESSION	The ID of the session is incorrect.
INVALIDPARAM_NODE	The ID of the user who starts the session is incorrect.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `startSession` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/startSession.aspx?session=12345678  
&node=337366&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/startSession.aspx">
  <input name="session" value="12345678">
  <input name="node" value="337366">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=startSession>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
    $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

    $loginparams = array (
        'sEmail' => 'some@email.com',
        'sPassword' => 'secretpassword'
    );

    $iSessionID = 12345678;
    $iNodeID = 337366;
    $sAuthCode = "4ahx...80u0";

    $startsessionparams = array (
        'iSessionID' => $iSessionID,
        'iNodeID' => $iNodeID,
        'sAuthCode' => $sAuthCode
    );

    $loginResult = $soapclient->login($loginparams);

    print_r($loginResult);

    $startSessionResult = $soapclient->startSession($startsessionparams);

    print_r($startSessionResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;
```

```

namespace apiSamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iSession = 12345678;
                int iNodeID = 337366;
                string sAuthCode = "4ahx...80u0";

                HttpWebRequest oLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint
                + "login.aspx?email=" + sEmail + "&pwd=" + sPwd);
                oLogin.CookieContainer = new CookieContainer();
                HttpWebResponse oRespLogin =
                (HttpWebResponse)oLogin.GetResponse();

                string sRespLogin = new
                StreamReader(oRespLogin.GetResponseStream())
                .ReadToEnd();
                Response.Write(sRespLogin + "<br />");

                CookieContainer sessioncookie = oLogin.CookieContainer;

                HttpWebRequest oReqStartSession =
                (HttpWebRequest)WebRequest.Create(sEndpoint
                + "startSession.aspx?session=" + iSession + "&node=" + iNodeID +
                "&authcode=" + sAuthCode);
                HttpWebResponse oRespStartSession = (HttpWebResponse)
                oReqStartSession.GetResponse();

                string sRespStartSession = new StreamReader
                (oRespStartSession.GetResponseStream()).ReadToEnd();
                Response.Write(sRespStartSession);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples

```

```

{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iSession = 12345678;
                int iNodeID = 337366;
                string sAuthCode = "4ahx...80u0";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.startSessionRet oStartSession =
proxy.startSession
(iSession, iNodeID, sAuthCode);
                Response.Write(oStartSession);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

holdSession

An administrator can put a technician session on hold. You can do this from the **Sessions** tab of the Rescue Administration Center.

Input Parameters

Element	Description
session	The ID of the target session. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Putting the session on hold succeeded.
NOTLOGGEDINASADMIN	Putting the session on hold failed because the administrator is not logged in.

Displayed Return Value	Description
ACTIONFAILED	Putting the session on hold failed because, for example, the session is on hold.
INVALIDPARAM_SESSION	The type of the session ID is incorrect. The type must be an integer.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `holdSession` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/holdSession.aspx?
session=12345678&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
holdSession.aspx">
  <input name="session" value=12345678>
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=holdSession>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

$loginparams = array (
  'sEmail' => 'some@email.com',
  'sPassword' => 'secretPassword'
);

$iSessionID = 12345678
$sAuthCode = "4ahx...80u0";

$holdsessionparams = array (
  'iSessionID' => $iSessionID,
  'sAuthCode' => ''
);
```

```

$loginResult = $soapclient->login($loginparams);

print_r($loginResult);

$holdSessionResult = $soapclient->holdSession($holdsessionparams);

print_r($holdSessionResult);
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                //sAuthCode is the return value from the requestAuthCode
                API
                string sAuthCode = "4ahx...80u0";

                HttpWebRequest oReqLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint
                + "login.aspx?email=" + sEmail + "&pwd=" + sPwd);
                oReqLogin.CookieContainer = new CookieContainer();

                HttpWebResponse oRespLogin =
                (HttpWebResponse)oReqLogin.GetResponse();
                string sRespLogin = new
                StreamReader(oRespLogin.GetResponseStream())
                .ReadToEnd();
                Response.Write(sRespLogin + "<br />");

                CookieContainer sessioncookie = oReqLogin.CookieContainer;

                HttpWebRequest oReqHoldSession =
                (HttpWebRequest)WebRequest.Create(sEndpoint
                + "holdSession.aspx?session=123456&authcode=" + sAuthCode);
                oReqHoldSession.CookieContainer = sessioncookie;

                HttpWebResponse oRespHoldSession = (HttpWebResponse)
                oReqHoldSession.GetResponse();
                string sRespHoldSession = new StreamReader(oRespHoldSession
                .GetResponseStream()).ReadToEnd();
            }
        }
    }
}

```

```

        Response.Write(sRespHoldSession + "<br />");
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                int iSession = 12345678;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
                Response.Write(oLogin + "<br />");

                WebServiceClients.holdSessionRet oHoldSession =
proxy.holdSession(iSession,
sAuthCode);
                Response.Write(oHoldSession + "<br />");
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```


transferSession

An administrator can transfer a technician session to another technician. You can transfer a session from the **Sessions** tab of the Rescue Administration Center. For more information, see *How to Transfer a Session* in the [Rescue Administration Center User Guide](#).



Important: You can only transfer sessions to technicians with the `transferSession` API method.

Input Parameters

Element	Description
<code>session</code>	The ID of the target session. Required.
<code>node</code>	The session is picked up by this technician. If a node is a technician and, for example, an administrator as well, then you can only transfer a session to the node as a technician. Therefore, you have to use the technician ID of the node. Required.
<code>noderef</code>	The reference of the target node, which can be NODE or CHANNEL. Optional, default is NODE.
<code>comment</code>	A comment on the transfer. Optional.
<code>authcode</code>	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Transferring the session succeeded.
NOTLOGGEDINASADMIN	Transferring the session failed because the administrator is not logged in.
ACTIONFAILED	Transferring the session failed because the technician is offline and the Technician Console is not running.
INVALIDPARAM_NODE	The ID of the user who transfers the session is incorrect. This error occurs in the following cases: <ul style="list-style-type: none">You define a technician for <code>node</code> and CHANNEL for <code>noderef</code>You define a channel for <code>node</code> and NODE for <code>noderef</code>The technician defined by the <code>node</code> parameter is associated with another Rescue account.

Displayed Return Value	Description
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `transferSession` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/transferSession.aspx?session=12345678
&node=337366&noderef=NODE&comment=transfer session&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
transferSession.aspx">
  <input name="session" value="12345678">
  <input name="node" value="337366">
  <input name="noderef" value="NODE">
  <input name="comment" value="transfer session">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=transferSession>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

$loginparams = array (
  'sEmail' => 'some@email.com',
  'sPassword' => 'secretPassword'
);

$iSessionID = 12345678;
$iNodeID = 337366;
$eNodeRef = NODE;
$sComment = "transfer session";
$sAuthCode = "4ahx...80u0";

$transfersessionparams = array (
  'iSessionID' => $iSessionID,
  'iNodeID' => $iNodeID,
  'eNodeRef' => $eNodeRef,
```

```

'sComment' => $sComment,
'sAuthCode' => $sAuthCode
);

$loginResult = $soapclient->login($loginparams);

print_r($loginResult);

$transferSessionResult = $soapclient-
>transferSession($transfersessionparams);

print_r($transferSessionResult);
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                int iSession = 12345678;
                int iNodeID = 337366;
                string eNodeRef = "NODE";
                string sComment = "transfer session";
                string sAuthCode = "4ahx...80u0";

                HttpWebRequest oLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint
                + "login.aspx?email=" + sEmail + "&pwd=" + sPwd);
                oLogin.CookieContainer = new CookieContainer();
                HttpWebResponse oRespLogin =
                (HttpWebResponse)oLogin.GetResponse();

                string sRespLogin = new
                StreamReader(oRespLogin.GetResponseStream())
                .ReadToEnd();
                Response.Write(sRespLogin + "<br />");

                CookieContainer sessioncookie = oLogin.CookieContainer;

                HttpWebRequest oReqTransferSession =
                (HttpWebRequest)WebRequest.Create

```

```

(sEndpoint + "transferSession.aspx?session=" + iSession + "&node=" +
iNodeID + "&noderef="
+ eNodeRef + "&comment=" + sComment + "&authcode=" + sAuthCode);
    HttpWebResponse oRespTransferSession = (HttpWebResponse)
oReqTransferSession.GetResponse();

        string sRespTransferSession = new StreamReader
(oRespTransferSession.GetResponseStream()).ReadToEnd();
        Response.Write(sRespTransferSession);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iSession = 12345678;
                int iNodeID = 337366;
                NODE_REF eNodeRef = NODE_REF.NODE;
                string sComment = "Escalating";
                string sAuthCode = "";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.transferSessionRet oTransferSession
= proxy.transferSession(iSession, iNodeID, eNodeRef, sComment, sAuthCode);
                Response.Write(oTransferSession);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

```
}  
}  
}
```

closeSession

An administrator can close a technician session. You can close a session from the **Sessions** tab of the Rescue Administration Center.

Input Parameters

Element	Description
session	The ID of the target session. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Closing the session succeeded.
NOTLOGGEDINASADMIN	Closing the session failed because the administrator is not logged in.
ACTIONFAILED	Closing a session failed because the technician is offline and the Technician Console is not running.
INVALIDPARAM_SESSION	The ID of the session is incorrect.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `closeSession` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/closeSession.aspx?session=12345678  
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
closeSession.aspx">
    <input name="session" value="12345678">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=closeSession>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
    $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

    $loginparams = array (
        'sEmail' => 'some@email.com',
        'sPassword' => 'secretPassword'
    );

    $cancelactionparams = array (
        'iSession' => 12345678,
        'sAuthCode' => '4ahx...80u0'
    );

    $loginResult = $soapclient->login($loginparams);

    print_r($loginResult);

    $closeSessionResult = $soapclient->closeSession($closesessionparams);

    print_r($closeSessionResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace APIexamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string sEndpoint = "https://secure.logmeinrescue.com/api/";

```

```

        string sEmail = "some@email.com"; //Rescue account email
address
        string sPwd = "secretPassword"; //Rescue account password
        int iSession = 12345678; //Session ID goes here
        string sAuthCode = "4ahx...80u0";

        //Login
        HttpWebRequest oLogin =
(HttpWebRequest)WebRequest.Create(sEndpoint
+ "login.aspx?email=" + sEmail + "&pwd=" + sPwd);
        oLogin.CookieContainer = new CookieContainer();
        //Get login response
        HttpWebResponse oLoginResp =
(HttpWebResponse)oLogin.GetResponse();
        string sLoginResp = new
StreamReader(oLoginResp.GetResponseStream()).ReadToEnd();
        Response.Write("Login status: " + sLoginResp + "<br />");
        //Add session cookie to CookieContainer to persist login
        CookieContainer sessioncookie = oLogin.CookieContainer;

        //Close the session
        HttpWebRequest oCloseSession =
(HttpWebRequest)WebRequest.Create(sEndpoint
+ "closeSession.aspx?session=" + iSession + "&authcode=" + sAuthCode);
        oCloseSession.CookieContainer = sessioncookie;
        //Get closeSession response
        HttpWebResponse oCloseSessionResp =
(HttpWebResponse)oCloseSession.GetResponse();
        string sCloseSessionResp = new StreamReader(oCloseSessionResp.
GetResponseStream()).ReadToEnd();
        Response.Write("Session: " + sCloseSessionResp);
    }
}
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using APIexamples.WebServiceClients;

namespace APIexamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            string sEmail = "some@email.com"; //Rescue account email
address
            string sPwd = "secretPassword"; //Rescue account password
            int iSession = 12345678; //ID of session to close
            string sAuthCode = "4ahx...80u0";

            WebServiceClients.API proxy = new WebServiceClients.API();

```

```

        proxy.CookieContainer = new CookieContainer();
        //Login
        WebServiceClients.loginRet oLogin = proxy.login(sEmail, sPwd);
        Response.Write(oLogin.ToString() + "<br />"); //You can
customize the response
        //Close session
        WebServiceClients.closeSessionRet oCloseSession =
proxy.closeSession(iSession,
sAuthCode);
        Response.Write(oCloseSession.ToString()); //You can customize
the response
    }
}
}
}

```

cancelAction

An administrator can cancel the action that he performed on a technician session. You can cancel an action from the **Sessions** tab of the Rescue Administration Center.

Input Parameters

Element	Description
session	The ID of the target session. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Cancelling the action succeeded.
NOTLOGGEDINASADMIN	Cancelling the action failed because the administrator is not logged in.
ACTIONFAILED	Cancelling the action failed because the technician is offline and the Technician Console is not running.
INVALIDPARAM_SESSION	The ID of the session on which you want to cancel an action is incorrect.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `cancelAction` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/cancelAction.aspx?session=12345678
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
cancelAction.aspx">
    <input name="session" value="12345678">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=cancelAction>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
    $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

    $loginparams = array (
        'sEmail' => 'some@email.com',
        'sPassword' => 'secretPassword'
    );

    $cancelactionparams = array (
        'iSession' => 12345678,
        'sAuthCode' => '4ahx...80u0'
    );

    $loginResult = $soapclient->login($loginparams);

    print_r($loginResult);

    $cancelActionResult = $soapclient->cancelAction($cancelactionparams);

    print_r($cancelActionResult);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
using System;
```

```

using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace apiSamples
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                string sEndpoint = "https://secure.logmeinrescue.com/api/";
                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                //ID of the session
                int iSession = 12345678;
                //sAuthCode is the return value from the requestAuthCode
                string sAuthCode = "4ahx...80u0";

                HttpWebRequest oReqLogin =
                (HttpWebRequest)WebRequest.Create(sEndpoint
                + "login.aspx?email=" + sEmail + "&pwd=" + sPwd);
                oReqLogin.CookieContainer = new CookieContainer();

                HttpWebResponse oRespLogin =
                (HttpWebResponse)oReqLogin.GetResponse();
                string sRespLogin = new
                StreamReader(oRespLogin.GetResponseStream())
                .ReadToEnd();
                Response.Write(sRespLogin + "<br />");

                CookieContainer sessioncookie = oReqLogin.CookieContainer;

                HttpWebRequest oReqCancelAction =
                (HttpWebRequest)WebRequest.Create(sEndpoint
                + "cancelAction.aspx?session=" + iSession + "&authcode=" + sAuthCode);
                oReqCancelAction.CookieContainer = sessioncookie;

                HttpWebResponse oRespCancelAction =
                (HttpWebResponse)oReqCancelAction
                .GetResponse();
                string sRespCancelAction = new
                StreamReader(oRespCancelAction.
                GetResponseStream()).ReadToEnd();
                Response.Write(sRespCancelAction + "<br />");
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iSession = 12345678;
                string sAuthCode = "4ahx...80u0";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
                Response.Write(oLogin + "<br />");

                WebServiceClients.cancelActionRet oCancelAction =
proxy.cancelAction
(iSession, sAuthCode);
                Response.Write(oCancelAction + "<br />");
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

isAnyTechAvailableOnChannel

Retrieves whether a channel has technicians online. You can see online technicians in the organization tree of the Rescue Administration Center.

Input Parameters

Element	Definition
ChannelID	The ID of the channel in which you are looking for online technicians. Required.

Element	Definition
ExcludingAwayAndBusy	<p>Determines whether away and busy statuses of technicians are considered as online. Optional.</p> <p>The possible values are the following:</p> <ul style="list-style-type: none"> • true • false <p>Setting this parameter to <code>false</code> disregards the status of the technician. If there is a technician on the channel, the return value of the <code>isAnyTechAvailableOnChannel</code> method will be <code>true</code>.</p> <p>Setting this parameter to <code>true</code> takes the status of technicians into account as follows:</p> <ul style="list-style-type: none"> • If there is a technician in online status, the return value of the <code>isAnyTechAvailableOnChannel</code> method will be <code>true</code>. • If all technicians on the channel are in busy or away status, the return value of the <code>isAnyTechAvailableOnChannel</code> method will be <code>false</code>.


Output

TRUE

TRUE / FALSE: See the description below

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	This return value is used for SOAP messages only and is displayed whenever the <code>isAnyTechAvailableOnChannel</code> method is called.

Displayed Return Value	Description
TRUE	<p>There is a technician available on the specified channel.</p> <p>If working hours are defined, the return value is TRUE if the following conditions are all true:</p> <ul style="list-style-type: none"> • There is a technician online. • A customer tries to start a session during non-working hours. • The Keep sessions alive option is selected. <p> Note: If working hours are defined, the return value is also TRUE if there is an online technician, the Abort sessions and show this webpage to the customer option is selected, and a customer tries to start a session during non-working hours.</p> <p>For information on defining working hours, see the Rescue Administration Center User Guide.</p>
FALSE	<p>There is no technician available on the specified channel.</p> <p>If working hours are defined, the return value is FALSE if there is no online technician.</p> <p>For information on defining working hours, see the Rescue Administration Center User Guide.</p>
HASONLINETECHNICIANS	<p>This return parameter is used for SOAP messages only and its value can be TRUE or FALSE depending on the conditions described above.</p>

Sample Code

The following are examples for using the `isAnyTechAvailableOnChannel` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/isAnyTechAvailableOnChannel.aspx?
ChannelID=1902861082&ExcludingAwayAndBusy=true
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
isAnyTechAvailableOnChannel.aspx">
  <input name="ChannelID" value="1902861082">
  <input name="ExcludingAwayAndBusy" value="true">
</form>
```

C# with SOAP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";

                //Login
                WebServiceClients.loginRet loginResult =
                proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                int iChannelID = [ID of channel node];
                bool bExcludingAwayAndBusy = true;
                bool bHasOnlineTechnicians;

                //Get result
                WebServiceClients.isAnyTechAvailableOnChannelRet
                isAnyTechAvailableResult =
                proxy.isAnyTechAvailableOnChannel(iChannelID,
                bExcludingAwayAndBusy, out bHasOnlineTechnicians);

                Response.Write(bHasOnlineTechnicians);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}
```

Reporting

These methods retrieve information for generating reports.

You can specify what information you need and how to display that information in your reports.



Important: Although the `authcode` parameter is available for all reporting methods, you cannot generate reports without logging in to Rescue.

getChat

Retrieves the chat log of any specific session of the user who is currently logged in. You can retrieve chat logs by selecting the **Chatlog** report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).



Note: Since this API method is session-based, changing the date, time, or format of the generated reports does not effect the output. For example, even if you set the reporting time between 10 am and 4 pm, this API method will return the complete chat log of a session that started outside the specified time span.

Input Parameters

Element	Description
<code>session</code>	The chat log of the specified session will be generated. Required.
<code>authcode</code>	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK CHATLOG:10:09 AM Connecting to: control.dev-app01.3amlabs.net
(192.168.2.180:443)
10:09 AM Connected to Applet (RSA 2048 bits, AES256-SHA 256 bits) 10:09 AM
Switched to P2P
10:09 AM Technician: Hello, how can I help you? 10:09 AM Customer Chris: I
have problems
connecting an external drive. 10:09 AM Technician: Let me see what I can do
for you.
10:10 AM Requesting Dashboard information. 10:10 AM Waiting for customer
permission to view
system information. 10:10 AM Dashboard Memory information received. 10:10
AM Dashboard Drives
information received. 10:10 AM Dashboard OS information received. 10:10 AM
Dashboard Scheduled
Tasks information received. 10:10 AM Dashboard Events information received.
10:10 AM Dashboard
CPU information received. 10:10 AM Dashboard Processes information
received. 10:10 AM Remote
Control successfully initiated. 10:12 AM Technician: OK, you are all set.
10:12 AM The
technician ended the session.
```

Chatlog: The complete chat log of a session.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the chat log succeeded.
NOTLOGGEDIN	Retrieving the chat log failed because the current user is no longer logged in.
INVALIDPARAM_SESSION	The specified session ID is invalid.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getChat` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getChat.aspx?session=12345678
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getChat.aspx">
  <input name="session" value="12345678">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getChat>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
```



```

$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//set up array for getChat request
$iSessionID = "12345678"; //populate this dynamically
$sAuthCode = "4ahx...80u0";

$getChatParams = array(
    'iSessionID' => $iSessionID,
    'sAuthCode' => $sAuthCode
);

$getChatResult = $soapclient->getChat($getChatParams);

//print out the chat full response
print_r("<b>getChat full response.</b><br />");
print_r($getChatResult);
echo "<br /><br />"; //formatting

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

        string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
        Response.Write("Login result: " + sResp + "<br />"); //You can customize
        the response

/*
//debug cookies

```

```

foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Chat info
string sSession = "12345678"; //populate this sessionID dynamically
HttpRequest oReqChat =
(HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
+ "getChat.aspx?session=" + sSession + "&authcode=" + sAuthCode);
oReqChat.CookieContainer = sessioncookie;

HttpWebResponse oRespChat = (HttpWebResponse) oReqChat.GetResponse();
string sRespChat = new
StreamReader(oRespChat.GetResponseStream()).ReadToEnd();
Response.Write("getChat result: " + sRespChat + "<br />"); //You can
customize the
response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
<title>Rescue API getChat Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

```

```

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                int iSession = 12345678;
                string sAuthCode = "4ahx...80u0";
                string sChatLog = "";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.getChatRet oGetChat =
proxy.getChat(iSession,
sAuthCode, out sChatLog);
                Response.Write(oGetChat + "<br />");

                Response.Write(sChatLog + "<br />");
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

getNote

Retrieves the notes of any specific session of the user who is currently logged in. You can retrieve session notes by selecting the **Chatlog** report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).



Note: Since this API method is session-based, changing the date, time, or format of the generated reports does not effect the output. For example, even if you set the reporting time between 10 am and 4 pm, this API method will return the Technician's note on a session that started outside the specified time span.

Input Parameters

Element	Description
session	The notes of the specified session will be listed. Required.

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK NOTE:Customer had issue with connecting an external drive.
```

Note: Technician's notes on a session

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving notes succeeded.
NOTLOGGEDIN	Retrieving notes failed because the current user is no longer logged in.
INVALIDPARAM_SESSION	The specified session ID is invalid.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getNote` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getNote.aspx?
session=12345678&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getNote.aspx">
  <input name="session" value="12345678">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getNote>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//set up array for getNote request
$iSessionID = "iSessionID"; //populate this dynamically
$sAuthCode = "";

$getNoteParams = array(
'iSessionID' => $iSessionID,
'sAuthCode' => $sAuthCode
);

$getNoteResult = $soapclient->getNote($getNoteParams);

//print out the note full response
print_r("<b>getNote full response.</b><br />");
print_r($getNoteResult);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/";
//add actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "4ahx...80u0";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
    + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
```

```

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

        string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
        Response.Write("Login result: " + sResp + "<br />"); //You can customize
        the response

        /*
        //debug cookies
        foreach (Cookie cook in oResp.Cookies)
        {
            Response.Write("Cookie:" + "<br />");
            Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
            >");
            Response.Write("Domain: " + cook.Domain + "<br />");
            Response.Write("Path: " + cook.Path + "<br />");
            Response.Write("Port: " + cook.Port + "<br />");
            Response.Write("Secure: " + cook.Secure + "<br />");

            Response.Write("When issued: " + cook.TimeStamp + "<br />");
            Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
            cook.Expired
            + "<br />");
            Response.Write("Don't save: " + cook.Discard + "<br />");
            Response.Write("Comment: " + cook.Comment + "<br />");
            Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
            Response.Write("Version: RFC " + cook.Version + "<br />");

            // Show the string representation of the cookie.
            Response.Write("String: " + cook.ToString());
        }*/

        //add cookies to cookie container
        CookieContainer sessioncookie = oReq.CookieContainer;

        //get the Note info
        string sSession = "12345678"; //populate this sessionID dynamically
        HttpRequest oReqNote =
            (HttpRequest)System.Net.WebRequest.Create(sEndpoint
            + "getNote.aspx?session=" + sSession + "&authcode=" + sAuthCode);
        oReqNote.CookieContainer = sessioncookie;

        HttpWebResponse oRespNote = (HttpWebResponse)oReqNote.GetResponse();
        string sRespNote = new
            StreamReader(oRespNote.GetResponseStream()).ReadToEnd();
        Response.Write("getNote result: " + sRespNote + "<br />"); //You can
        customize the
        response

        %>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getNote Test</title>
</head>
<body>

```

```
</body>  
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Net;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using apiSamples.WebServiceClients;  
  
namespace apiSamples  
{  
    public partial class SOAP : System.Web.UI.Page  
    {  
        protected void Page_Load(object sender, EventArgs e)  
        {  
            try  
            {  
                WebServiceClients.API proxy = new WebServiceClients.API();  
                proxy.CookieContainer = new CookieContainer();  
  
                string sEmail = "some@email.com";  
                string sPwd = "secretPassword";  
                string sAuthCode = "4ahx...80u0";  
                int iSession = 12345678;  
                string sNote = "";  
  
                WebServiceClients.loginRet oLogin = proxy.login(sEmail,  
sPwd);  
  
                Response.Write(oLogin + "<br />");  
  
                WebServiceClients.getNoteRet oGetNote =  
proxy.getNote(iSession,  
sAuthCode, out sNote);  
                Response.Write(oGetNote + "<br />");  
  
                Response.Write(sNote);  
            }  
            catch (Exception ex)  
            {  
                Response.Write(ex.Message);  
            }  
        }  
    }  
}
```

getReport

Retrieves the report based on previously defined parameters. The output of the report depends on the parameters defined with the `setReportArea`, `setDelimiter`, `setOutput`, `setReportDate`, `setReportTime`, and `setTimeZone` methods. You can retrieve a report in the

Rescue Administration Center from the **Reports** tab. For information on reports, see *Generating Reports* in the *Rescue Administration Center User Guide*.



Important: You can retrieve reports every 60 seconds.

Input Parameters

Element	Description
node	The report is generated based on this node. Required.
nodetype	The reference of the node, which can be NODE or CHANNEL. Optional, default is NODE.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK Start Time|End Time|Last Action Time|Technician Name|Technician ID|
Session ID|Session Type|Status|Name|Custom field 1|Custom field 2|Custom
field 3|
Custom field 4|Custom field 5|Tracking ID|Customer IP|Device ID|Incident
Tools Used|
Resolved Unresolved|Channel ID|Channel Name|Calling Card|Connecting Time|
Waiting Time|
Total Time|Active Time|Work Time|Hold Time|Time in Transfer|Rebooting Time|
Reconnecting Time|Platform| 16/7/2012 8:17:23 AM|16/7/2012 8:18:07 AM|
16/7/2012
8:18:06 AM|Justin Case|1028224|1595374|Applet|Closed by technician|
sadcac|||||
|192.168.5.74|7945e345a43a29b5074df0ba189f9cb8|RV PS SP|||||00:00:06|
00:00:01
|00:00:44|00:00:44|00:00:38|00:00:00|00:00:00|00:00:00|00:00:00|Windows 7|
```

The session report has the following header columns:

- Start Time
- End Time
- Technician Name
- Technician ID
- Session ID
- Session Type
- Status
- [Defined name for the name field]
- [Defined name for the custom field 1]
- [Defined name for the custom field 2]
- [Defined name for the custom field 3]
- [Defined name for the custom field 4]
- [Defined name for the custom field 5]
- Tracking ID
- Customer IP
- Computer ID
- Incident Tools Used ID

- Resolved
- Unresolved
- Channel ID
- Channel Name
- Calling Card
- Connecting Time
- Waiting Time
- Total Time
- Active Time
- Work Time
- Hold Time
- Transfer Time
- Rebooting Time
- Reconnecting Time
- Platform
- Technician Email
- Browser Type

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving a report succeeded.
NOTLOGGEDIN	Retrieving a report failed because the current user is no longer logged in.
POLLRATEEXCEEDED	This error is displayed when you call the <code>getReport</code> method more than 10 times in three minutes. Should you need to retrieve reports more often, you must use the authcode implementation.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing node.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReport` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReport.aspx?node=337366
&nodetype=NODE&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/getReport.aspx">
  <input name="node" value="337366">
  <input name="nodetype" value="NODE">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getReport>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

//define variables
$email = "some@email.com";
$password = "secretPassword";
$beginDate = "05/09/2007"; //define this with your own data
$endDate = "07/01/2007"; //define this appropriately based on your data
$reportArea = "SESSION";
$nodeID = "337366";
$nodeRef = "NODE";
$authCode = "";

//define parameters
$params = array (
  'email' => $email,
  'password' => $password);

//login
$result = $soapclient->login($params);

/*****uncomment for debug
//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($result);
echo "<br /><br />"; //formatting
*****/

//set the report area
//define parameters
$params = array (
  'reportArea' => $reportArea,
  'authCode' => $authCode
);

$reportAreaResponse = $soapclient->setReportArea($params);

/*****uncomment for debug
print_r("<b>setReportArea Full response.</b> <br />");
print_r($reportAreaResponse);
echo "<br /><br />"; //formatting
*****/
```

```

//set the time frame
$reportDateParams= array(
'sBeginDate' => $sBeginDate,
'sEndDate' => $sEndDate,
'sAuthCode' => $sAuthCode
);

$setReportDateResponse = $soapclient->setReportDate($reportDateParams);

/***** uncomment for debug
print_r("<b>setReportDate Full response.</b> <br />");
print_r($setReportDateResponse);
echo "<br /><br />"; //formatting
*****/

//finally, get the report
//set up array
$getReportParams = array(
'iNodeID' => $iNodeID,
'eNodeRef' => $eNodeRef,
'sAuthCode' => $sAuthCode
);

$getReportResponse = $soapclient->getReport($getReportParams);

/***** uncomment for debug
print_r("<b>getReport Full response (need to parse.)</b> <br />");
print_r($getReportResponse);
echo "<br /><br />"; //formatting
*****/

//parse results into an array (NuSOAP stinks at multilevel XML
$reportdata = explode("\n", $getReportResponse['sReport']);
foreach($reportdata as $key => $val) {
    if($key == 0) {
        $COLUMN = explode ("|", $val);
    }
    $COLDATA = explode("|", $val);
    foreach($COLDATA as $ckey => $val) {
        if(empty($COLUMN[$ckey])) {
            $COLUMN[$ckey] = $ckey;
        } else {
            $COLUMN[$ckey] = str_replace(" ", "", $COLUMN[$ckey]);
        }
        $REPORT[$key][$COLUMN[$ckey]] = $val;
    }
}

/*****uncomment for debug
//print array to browser
print_r("<b>results parsed into an array.</b><br />");
print_r($REPORT);
print_r("<br /><br />");
*****/

$reportdata = $REPORT;
//get just the report data, and number off nodes in it
$reportentries = count($REPORT);
print_r("<b>Number of report lines:" . $reportentries . "</b><br /><br />");

```

```
//format report nicely
print_r("<b>formatted report.</b><br />");
print_r("<table frame = \"border\" border = \"0\">");
for ($i = 0; $i < $reportentries; $i += 1)
{
    print_r("<tr>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["SessionID"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["StartTime"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["EndTime"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["TechnicianID"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["CustomerIP"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["ComputerID"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["Status"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["Name"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["PhoneNumber"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["EmailAddress"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["Product"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["OrderID"] . "</td>");
    print_r("<td style=white-space: nowrap>"
    . $reportdata[$i]["ChannelID"] . "</td>");
    print_r("</tr>");
}
print_r("</table>");

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
        NODE_REF eNodeRef = NODE_REF.NODE;

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
```

```

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Report info
//remember to tag your intended node to the URL
string sNode = "";
HttpRequest oReqReport =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "getReport.aspx?node=" + sNode + "&noderef=" + eNodeRef + "&authcode=" +
sAuthCode);
oReqReport.CookieContainer = sessioncookie;

HttpWebResponse oRespReport = (HttpWebResponse)oReqReport.GetResponse();
string sRespReport = new
StreamReader(oRespReport.GetResponseStream()).ReadToEnd();
Response.Write("getReport result: " + sRespReport + "<br />"); //You can
customize the
response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getReport Test</title>
</head>
<body>

```

```
</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                int iNodeID = 337366;
                NODE_REF eNodeRef = NODE_REF.NODE;
                string sReport = "";
                DateTime dBeginDate = new DateTime(2010, 7, 1);
                DateTime dEndDate = new DateTime(2010, 7, 8);

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
                Response.Write(oLogin + "<br />");

                WebServiceClients.setReportAreaRet oSetReportArea =
proxy.setReportArea
(REPORT.LOGIN, sAuthCode);
                Response.Write(oSetReportArea + "<br />");

                WebServiceClients.setReportDateRet oSetReportDate =
proxy.setReportDate_v2
(dBeginDate, dEndDate, sAuthCode);

                WebServiceClients.getReportRet oGetReport =
proxy.getReport(iNodeID,
eNodeRef, sAuthCode, out sReport);
                Response.Write(oGetReport + "<br />");

                Response.Write(sReport);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

```
}  
}
```

Structure of Generated Reports

Reports can be generated in text and XML formats.

Text-based Report Output

Text-based reports have the following elements:

Report element	Content
<report>	<header_columns> <report_rows>
<header_columns>	<field_value> <delimiter> <header_columns> <field_value> <new_line_char>
<report_rows>	<report_row> <report_row> <report_row>
<report_row>	<field_value> <delimiter> <report_row> <field_value> <new_line_char>
<field_value>	<char> <field_value> <char>
<char>	[.]
<new_line_char>	'\n'

XML-based Report Output

XML-based reports have the following elements:

Report element	Content
<report>	<report_begin> <report_header_begin> <header_columns> <report_header_end> <report_data_begin> <report_rows> <report_data_end> <report_end>
<header_columns>	<report_field_begin> <field_value> <report_field_end> <header_columns> <report_field_begin> <field_value> <report_field_end>
<report_rows>	<report_row_begin> <report_row> <report_row_end> <report_rows> <report_row_begin> <report_row> <report_row_end>
<report_row>	<report_field_begin> <field_value> <report_ field _end> <report_row> <report_ field _begin> <field_value> <report_ field _end>
<field_value>	<char> <field_value> <char>
<char>	[.]
<report_begin>	'<report>'

Report element	Content
<report_end>	'</report>'
<report_header_begin>	'<header>'
<report_header_end>	'</header>'
<report_data_begin>	'<data>'
<report_data_end>	'</data>'
<report_row_begin>	'<row>'
<report_row_end>	'</row>'
<report_field_begin>	'<field>'
<report_field_end>	'</field>'

Columns of Generated Reports

Session Report (Report area: 0)

Session reports have the following columns:

Customer Survey Report (Report area: 1)

The customer survey report contains the following columns:

Report column name	Data type	Data length
Source	String	128 characters
Session ID	Integer	unspecified
Date	DateTime	unspecified
Name	String	128 characters
[Defined questions for the survey]	String	128 characters
Technician Name	String	128 characters
Technician ID	Integer	unspecified
Technician Email	String	128 characters

Login Report (Report area: 2)

Login reports have the following columns:

Report column name	Data type	Data length
Login Date	DateTime	unspecified
Name	String	128 characters
User ID	Integer	unspecified

Report column name	Data type	Data length
Email	String	128 characters
Start Time	DateTime	unspecified
End Time	DateTime	unspecified
Total Login Time	DateTime	unspecified
IP Address	String	15 characters
Busy Time	DateTime	unspecified
Away Time	DateTime	unspecified
Idle Time	DateTime	unspecified

Missed Sessions Report (Report area: 3)

The missed sessions report has the following columns:

Report column name	Data type	Data length
Start Time	DateTime	unspecified
End Time	DateTime	unspecified
Total Time	DateTime	unspecified
Session ID	Integer	unspecified
Session Type	String	100 characters
Status	String	64 characters
[Defined name for the name field]	String	128 characters
[Defined name for the custom field 1..5]	String	128 characters
Tracking ID	String	256 characters
Customer IP	String	15 characters
Private Session	String	128 characters
Channel	String	64 characters
Technician Group	String	128 characters

Performance Report (Report area: 4)

The performance report contains the following columns:

Report column name	Data type	Data length
Technician Name	String	128 characters
Technician ID	Integer	unspecified
Technician Email	String	128 characters
Total Login Time	DateTime	unspecified

Report column name	Data type	Data length
Number of Sessions	Integer	unspecified
Number of Sessions per Hour	String	128 characters
Avg Pick-up Speed	DateTime	unspecified
Avg Duration	DateTime	unspecified
Average Work Time	DateTime	unspecified
Longest Session	DateTime	unspecified
Total Active Time	DateTime	unspecified
Total Work Time	DateTime	unspecified

Chat Log Report (Report area: 5)

The chat log report contains the following columns:

Report column name	Data type	Data length
Start Time	DateTime	unspecified
End Time	DateTime	unspecified
Total Time	DateTime	unspecified
Session ID	Integer	unspecified
[Defined name for the name field]	String	128 characters
Technician Name	String	128 characters
Technician ID	Integer	unspecified
Technician Email	String	128 characters
Chatlog	String	2048 characters
Notes	String	1024 characters

Custom Fields Report (Report area: 6)

The custom fields report contains the following columns:

Report column name	Data type	Data length
Start Time	DateTime	unspecified
End Time	DateTime	unspecified
Total Time	DateTime	unspecified
Session ID	Integer	unspecified
[Defined name for the name field]	String	128 characters
[Defined name for the custom field 1..5]	String	64 characters

Report column name	Data type	Data length
Tracking ID	String	256 characters
Technician Name	String	128 characters
Technician ID	Integer	unspecified
Technician Email	String	128 characters

Transferred Sessions Report (Report area: 7)

The transferred sessions report contains the following columns:

Report column name	Data type	Data length
Session ID	Integer	unspecified
Time of Transfer	DateTime	unspecified
[Defined name for the name field]	String	128 characters
Transferred by	String	128 characters
Transferred from	String	128 characters
Transferred to	String	128 characters
Transfer Comment	String	128 characters
Time in Transfer	DateTime	unspecified

Technician Survey Report (Report area: 8)

The technician survey report contains the following columns:

Report column name	Data type	Data length
Source	String	128 characters
Session ID	Integer	unspecified
Date	DateTime	unspecified
[Defined name for the name field]	String	128 characters
[Defined questions for the survey]	String	128 characters
Technician Name	String	128 characters
Technician ID	Integer	unspecified
Technician Email	String	128 characters

Collaboration Chatlog Report (Report area: 9)

The collaboration chatlog report contains the following columns:

Report column name	Data type	Data length
Start Time	DateTime	unspecified

Report column name	Data type	Data length
End Time	DateTime	unspecified
Total Time	DateTime	unspecified
Session ID	Integer	unspecified
[Defined name for the name field]	String	128 characters
Technician Name	String	128 characters
Technician ID	Integer	unspecified
Technician Email	String	128 characters
Chat Log	String	2048 characters

Failed Sessions Report (Report area: 10)

The failed sessions report contains the following columns:

Report column name	Data type	Data length
Start Time	DateTime	unspecified
Session ID	Integer	unspecified
Session Type	String	100 characters
Status	String	64 characters
[Defined name for the name field]	String	128 characters
[Defined name for the custom field 1..5]	String	128 characters
Customer IP	String	15 characters
Private Session	String	128 characters
Channel	String	64 characters

Customer Survey Issuance Report (Report area: 12)

The customer survey issuance report contains the following columns:

Report column name	Data type	Data length
Source	String	128 characters
Session ID	Integer	unspecified
Date	DateTime	unspecified
[Name]	String	128 characters
Technician Name	String	128 characters
Technician ID	Integer	unspecified
Technician Email	String	128 characters

Report column name	Data type	Data length
Survey issued by closing technician	String	3 characters

External Technician Session Report (Report area 13)

The external technician session report contains the following columns:

Report column name	Data type	Data length
Start Time	DateTime	unspecified
End Time	DateTime	unspecified
Last Action Time	DateTime	unspecified
External Technician Name	String	128 characters
External Technician Email	String	128 characters
Inviter's Name	String	128 characters
Inviter's ID	String	128 characters
Inviter's Email	String	128 characters
Session ID	Integer	unspecified
Session Type	String	100 characters
Status	String	64 characters
[Name]	String	128 characters
[Custom Fields]	String	64 characters
Tracking ID	String	256 characters
Customer IP	String	15 characters
Device ID	String	128 characters
Incident Tool Used	String	128 characters
Channel ID	Integer	unspecified
Channel Name	String	64 characters
Waiting Time	DateTime	unspecified
Total Time	DateTime	unspecified
Active Time	DateTime	unspecified
Work Time	DateTime	unspecified
Hold Time	DateTime	unspecified
Rebooting Time	DateTime	unspecified
Reconnecting Time	DateTime	unspecified
Platform	String	20 characters
Browser Type	String	unspecified

Rebooting/Reconnecting Report (Report area 15)

The rebooting/reconnecting report contains the following columns:

Report column name	Data type	Data length
Session ID	Integer	unspecified
[Custom Fields]	String	64 characters
Technician Name	String	128 characters
Technician ID	Integer	unspecified
Technician Email	String	128 characters
Channel ID	Integer	unspecified
Channel Name	String	64 characters
Technician Group	String	128 characters
Start Time	DateTime	unspecified
End Time	DateTime	unspecified
Last Action Time	DateTime	unspecified
Event type	String	unspecified
Rebooting/Reconnecting Start Time	DateTime	unspecified
Rebooting/Reconnecting End Time	DateTime	unspecified

Transferred Sessions - Extended Report (Report area 16)

The transferred sessions - extended report contains the following columns:

Report column name	Data type	Data length
Session ID	Integer	unspecified
Time of Transfer	DateTime	unspecified
Waiting Time	DateTime	unspecified
[Defined name for the name field]	String	128 characters
Transferred by - Technician ID	Integer	unspecified
Transferred by	String	128 characters
Transferred from - Technician/Channel ID	Integer	unspecified
Transferred from	String	128 characters
Transferred to - Technician/Channel ID	Integer	unspecified
Transferred to	String	128 characters
Transfer Comment	String	128 characters

Report column name	Data type	Data length
Time in Transfer	DateTime	unspecified
Transferred from - Technician Group ID	Integer	unspecified
Technician from - Technician Group	String	128 characters
Transferred to - Technician Group ID	Integer	unspecified
Technician to - Technician Group	String	128 characters
Chatlog	String	unspecified

Failed Sessions - Extended Report (Report area 17)

The failed sessions - extended report contains the following columns:

Report column name	Data type	Data length
Start Time	DateTime	unspecified
Session ID	Integer	unspecified
Session Type	String	100 characters
Status	String	64 characters
[Defined name for the name field]	String	128 characters
[Defined name for the custom field 1..5]	String	128 characters
Customer IP	String	15 characters
Private Session Technician Name	String	128 characters
Channel	String	64 characters
Technician Email	String	128 characters
Technician Group ID	Integer	unspecified
Technician Group	String	128 characters

Audit report (Report area 18)

The audit report contains the following columns

Report column name	Data type	Data length
Requested by	String	128 characters
Entity type	String	128 characters
Entity ID	Integer	unspecified
Entity name	String	256 characters

Report column name	Data type	Data length
Change type	String	128 characters
Last changed	DateTime	unspecified
Section	String	unspecified
Field	String	unspecified
Old Value	String	unspecified
Old Action	String	unspecified
New Value	String	unspecified
New Action	String	unspecified

getReport_v2

Retrieves the report based on previously defined parameters. The output of the report depends on the parameters defined with the `setReportArea`, `setDelimiter`, `setOutput`, `setReportDate`, `setReportTime`, and `setTimeZone` methods. You can retrieve a report in the Rescue Administration Center from the **Reports** tab. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 2 of getReport

The following values have been introduced to the `nodetype` input parameter in the `getReport_v2` method:

EXTERNALROOT A report will be generated on the **External Technicians** organizational category that contains all external collaborators.

EXTERNALTECH A report will be generated on a specific external technician.

EXTERNALGROUP report will be generated on a group of external technicians.



Important: You can retrieve reports every 60 seconds.

Input Parameters

Element	Description
node	The report is generated based on this node ID. Required.

Element	Description
nodetype	<p>The reference of the node. Optional, default is NODE.</p> <p>The following node types are available:</p> <ul style="list-style-type: none"> • NODE - Use when the node is a Technician • CHANNEL - Use when the node is a channel • EXTERNALROOT - Use when the node is the root element of external technicians. In the Administration Center, this is the External Technicians label on the organization tree. • EXTERNALTECH - Use when the node is an external technician • EXTERNALGROUP - Use when the node is an external technician group
authcode	<p>The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.</p>

Output

```
OK Start Time|End Time|Last Action Time|Technician Name|Technician ID|
Session ID|Session Type|Status|Name|Custom field 1|Custom field 2|Custom
field 3|
Custom field 4|Custom field 5|Tracking ID|Customer IP|Device ID|Incident
Tools Used|
Resolved Unresolved|Channel ID|Channel Name|Calling Card|Connecting Time|
Waiting Time|
Total Time|Active Time|Work Time|Hold Time|Time in Transfer|Rebooting Time|
Reconnecting Time|Platform| 16/7/2012 8:17:23 AM|16/7/2012 8:18:07 AM|
16/7/2012
8:18:06 AM|Justin Case|1028224|1595374|Applet|Closed by technician|
sadcac|||||
|192.168.5.74|7945e345a43a29b5074df0ba189f9cb8|RV PS SP|||||00:00:06|
00:00:01
|00:00:44|00:00:44|00:00:38|00:00:00|00:00:00|00:00:00|00:00:00|Windows 7|
```

The session report has the following header columns:

- Start Time
- End Time
- Last Action Time
- Technician Name
- Technician ID
- Session ID
- Session Type
- Status
- [Defined name for the name field]
- [Defined name for the custom field 1..5]
- Tracking ID
- Customer IP
- Device ID
- Incident Tools Used

- Resolved Unresolved
- Channel ID
- Channel Name
- Calling Card
- Connecting Time
- Waiting Time
- Total Time
- Active Time
- Work Time
- Hold Time
- Time in Transfer
- Rebooting Time
- Reconnecting Time
- Platform

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving a report succeeded.
NOTLOGGEDIN	Retrieving a report failed because the current user is no longer logged in.
POLLRATEEXCEEDED	The specified interval of requesting a report is too short.
INVALIDPARAM_NODE	The specified ID is not the ID of an existing node.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReport_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReport_v2.aspx?
node=-2&nodetype=EXTERNALROOT
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReport_v2.aspx">
  <input name="node" value="-2">
  <input name="nodetype" value="EXTERNALROOT">
```

```
<input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getReport_v2.

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
        NODE_REF eNodeRef = NODE_REF.EXTERNALTECH;

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
        Response.Write("Cookie:" + "<br />");
        Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br />");
        Response.Write("Domain: " + cook.Domain + "<br />");
        Response.Write("Path: " + cook.Path + "<br />");
        Response.Write("Port: " + cook.Port + "<br />");
        Response.Write("Secure: " + cook.Secure + "<br />");

        Response.Write("When issued: " + cook.TimeStamp + "<br />");
        Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
        cook.Expired + "<br />");
        Response.Write("Don't save: " + cook.Discard + "<br />");
        Response.Write("Comment: " + cook.Comment + "<br />");
    }
    */
```

```

Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Report info
//remember to tag your intended node to the URL
string sNode = "";
HttpRequest oReqReport =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
    + "getReport_v2.aspx?node=" + sNode + "&noderef=" + eNodeRef +
    "&authcode=" + sAuthCode);
oReqReport.CookieContainer = sessioncookie;

HttpResponse oRespReport = (HttpResponse)oReqReport.GetResponse();
string sRespReport = new
    StreamReader(oRespReport.GetResponseStream()).ReadToEnd();
Response.Write("getReport_v2 result: " + sRespReport + "<br />"); //You
    can customize the
    response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getReport_v2 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
            }
        }
    }
}

```

```

        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
        int iNodeID = 337366;
        NODE_REF eNodeRef = NODE_REF.EXTERNALTECH;
        string sReport = "";
        DateTime dBeginDate = new DateTime(2010, 7, 1);
        DateTime dEndDate = new DateTime(2010, 7, 8);

        WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

        Response.Write(oLogin + "<br />");

        WebServiceClients.setReportAreaRet oSetReportArea =
proxy.setReportArea
(REPORT.LOGIN, sAuthCode);
        Response.Write(oSetReportArea + "<br />");

        WebServiceClients.setReportDateRet oSetReportDate =
proxy.setReportDate_v2
(dBeginDate, dEndDate, sAuthCode);

        WebServiceClients.getReportRet oGetReport_v2 =
proxy.getReport_v2(iNodeID,
eNodeRef, sAuthCode, out sReport);
        Response.Write(oGetReport_v2 + "<br />");

        Response.Write(sReport);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

getReportArea

Displays the report type that is currently set for retrieval. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK AREA:0
```

AREA: Report area

The following report areas are available:

- 0 - Session report
- 1 - Customer Survey report
- 2 - Login report
- 3 - Missed Sessions report
- 4 - Performance report
- 5 - Chat Log report
- 6 - Custom Fields report
- 7 - Transferred Sessions report
- 8 - Technician Survey report
- 9 - Collaboration Chatlog report
- 10 - Failed Sessions report

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the report type succeeded.
NOTLOGGEDIN	Retrieving the report type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportArea` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportArea.aspx?
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReportArea.aspx">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getReportArea>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//define parameters
$reportarams = array (
'sAuthCode' => $sAuthCode
);

//login
$getReportAreaResult = $soapclient->getReportArea($reportareaparams);

//print the result
echo "<b>getReportArea full response.</b><br />"; //formatting
print_r($getReportAreaResult);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
actionName.aspx?
for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "4ahx...80u0";

    //set up the request
    HttpWebRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);
```

```

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

        string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
        Response.Write("Login result: " + sResp + "<br />"); //You can customize
        the response

        /*
        //debug cookies
        foreach (Cookie cook in oResp.Cookies)
        {
            Response.Write("Cookie:" + "<br />");
            Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br />");
            Response.Write("Domain: " + cook.Domain + "<br />");
            Response.Write("Path: " + cook.Path + "<br />");
            Response.Write("Port: " + cook.Port + "<br />");
            Response.Write("Secure: " + cook.Secure + "<br />");

            Response.Write("When issued: " + cook.TimeStamp + "<br />");
            Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
                cook.Expired + "<br />");
            Response.Write("Don't save: " + cook.Discard + "<br />");
            Response.Write("Comment: " + cook.Comment + "<br />");
            Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
            Response.Write("Version: RFC " + cook.Version + "<br />");

            // Show the string representation of the cookie.
            Response.Write("String: " + cook.ToString());
        }*/

        //add cookies to cookie container
        CookieContainer sessioncookie = oReq.CookieContainer;

        //get the ReportArea info
        HttpRequest oReqReportArea =
            (HttpRequest)System.Net.WebRequest.Create(sEndpoint
                + "getReportArea.aspx?authcode=" + sAuthCode);
        oReqReportArea.CookieContainer = sessioncookie;

        HttpWebResponse oRespReportArea =
            (HttpWebResponse)oReqReportArea.GetResponse();
        string sRespReportArea = new
            StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
        Response.Write("getReportArea result: " + sRespReportArea + "<br />"); //
        You can
        customize the response

        %>

        <html xmlns="http://www.w3.org/1999/xhtml" >
        <head id="Head1" runat="server">
            <title>Rescue API getReportArea Test</title>
        </head>
        <body>

        </body>
        </html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT eReportArea;

                WebServiceClients.getReportAreaRet oGetReportArea =
                proxy.getReportArea
                (sAuthCode, out eReportArea);
                Response.Write(oGetReportArea + "<br />");

                Response.Write(eReportArea);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

getReportArea_v2

Displays the report type that is currently set for retrieval. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 2 of getReportArea

The following report area has been introduced in the `getReportArea_v2` method:

External Collaboration Chatlog report This API method can now retrieve reports containing the chatlog between a Rescue technician and an external technician.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK AREA:12
```

AREA: Report area

The following report areas are available:

- 0 - Session report
- 1 - Customer Survey report
- 2 - Login report
- 3 - Missed Sessions report
- 4 - Performance report
- 5 - Chat Log report
- 6 - Custom Fields report
- 7 - Transferred Sessions report
- 8 - Technician Survey report
- 9 - Collaboration Chatlog report
- 10 - Failed Sessions report
- 12 - External Collaboration Chatlog report

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the report type succeeded.
NOTLOGGEDIN	Retrieving the report type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportArea_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportArea_v2.aspx?  
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/getReportArea_v2.aspx">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getReportArea_v2.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//define parameters
$reportarams = array (
'sAuthCode' => $sAuthCode
);

//login
$getReportArea_v2Result = $soapclient->getReportArea($reportareaparams);

//print the result
echo "<b>getReportArea_v2 full response.</b><br />"; //formatting
print_r($getReportArea_v2Result);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
HttpWebRequest oReqReportArea =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
    + "getReportArea_v2.aspx?authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpWebResponse oRespReportArea =
    (HttpWebResponse) oReqReportArea.GetResponse();
string sRespReportArea = new
    StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();

```

```

Response.Write("getReportArea_v2 result: " + sRespReportArea + "<br />");
//You can
customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getReportArea_v2 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT_V2 eReportArea;

                WebServiceClients.getReportAreaRet oGetReportArea =
                proxy.getReportArea_v2
                (sAuthCode, out eReportArea);
                Response.Write(oGetReportArea + "<br />");

                Response.Write(eReportArea);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

getReportArea_v3

Displays the report type that is currently set for retrieval. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 3 of getReportArea

The following report area has been introduced in the `getReportArea_v3` method:

Customer Survey Issuance report This API method retrieves reports about whether the closing technician issued the customer survey at the end of the Rescue session. To change the issuing technician in the Administration Center, go to **Global Settings > Customer Survey Issuance > Survey issued by**.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK AREA:13
```

AREA: Report area

The following report areas are available:

- 0 - Session report
- 1 - Customer Survey report
- 2 - Login report
- 3 - Missed Sessions report
- 4 - Performance report
- 5 - Chat Log report
- 6 - Custom Fields report
- 7 - Transferred Sessions report
- 8 - Technician Survey report
- 9 - Collaboration Chatlog report
- 10 - Failed Sessions report
- 12 - External Collaboration Chatlog report
- 13 - Customer Survey Issuance report

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the report type succeeded.

Displayed Return Value	Description
NOTLOGGEDIN	Retrieving the report type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportArea_v3` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportArea_v3.aspx?
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReportArea_v3.aspx">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getReportArea_v3.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";
```

```

//define parameters
$reportarams = array (
'sAuthCode' => $sAuthCode
);

//login
$getReportArea_v3Result = $soapclient->getReportArea($reportareaparams);

//print the result
echo "<b>getReportArea_v3 full response.</b><br />"; //formatting
print_r($getReportArea_v3Result);
echo "<br /><br />"; //formatting

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "4ahx...80u0";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?
    email="
    + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");

```



```

Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
HttpRequest oReqReportArea =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
+ "getReportArea_v3.aspx?authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
    (HttpResponse) oReqReportArea.GetResponse();
string sRespReportArea = new
    StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("getReportArea_v3 result: " + sRespReportArea + "<br />");
//You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getReportArea_v3 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();
            }
        }
    }
}

```

```

        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
        REPORT_V3 eReportArea;

        WebServiceClients.getReportAreaRet oGetReportArea =
proxy.getReportArea_v3(sAuthCode, out eReportArea);
        Response.Write(oGetReportArea + "<br />");

        Response.Write(eReportArea);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}

```

getReportArea_v5

Displays the report type that is currently set for retrieval. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 5 of getReportArea

The following report area has been introduced in the `getReportArea_v5` method:

**Rebooting/
Reconnecting
report** This API method retrieves reports about reconnecting or rebooting events.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK AREA:15
```

AREA: Report area

The following report areas are available:

- 0 - Session report
- 1 - Customer Survey report
- 2 - Login report
- 3 - Missed Sessions report
- 4 - Performance report
- 5 - Chat Log report
- 6 - Custom Fields report

- 7 - Transferred Sessions report
- 8 - Technician Survey report
- 9 - Collaboration Chatlog report
- 10 - Failed Sessions report
- 12 - External Collaboration Chatlog report
- 13 - Customer Survey Issuance report
- 15 - Rebooting/Reconnecting report

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the report type succeeded.
NOTLOGGEDIN	Retrieving the report type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportArea_v5` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportArea_v5.aspx?
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReportArea_v5.aspx">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getReportArea_v5.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");
```

```

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//define parameters
$reportarams = array (
'sAuthCode' => $sAuthCode
);

//login
$getReportArea_v5Result = $soapclient->getReportArea($reportareaparams);

//print the result
echo "<b>getReportArea_v5 full response.</b><br />"; //formatting
print_r($getReportArea_v5Result);
echo "<br /><br />"; //formatting

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "4ahx...80u0";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?
    email="
    + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();

```

```

Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
HttpRequest oReqReportArea =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
+ "getReportArea_v5.aspx?authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
    (HttpResponse)oReqReportArea.GetResponse();
string sRespReportArea = new
    StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("getReportArea_v5 result: " + sRespReportArea + "<br />");
    //You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getReportArea_v5 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;

```

```

using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT_V5 eReportArea;

                WebServiceClients.getReportAreaRet oGetReportArea =
                proxy.getReportArea_v5(sAuthCode, out eReportArea);
                Response.Write(oGetReportArea + "<br />");

                Response.Write(eReportArea);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

getReportArea_v6

Displays the report type that is currently set for retrieval. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 6 of getReportArea

The following report area has been introduced in the `getReportArea_v6` method:

Transferred Sessions - Extended report This API method retrieves reports about transferred sessions.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK AREA:16
```

AREA: Report area

The following report areas are available:

- 0 - Session report
- 1 - Customer Survey report
- 2 - Login report
- 3 - Missed Sessions report
- 4 - Performance report
- 5 - Chat Log report
- 6 - Custom Fields report
- 7 - Transferred Sessions report
- 8 - Technician Survey report
- 9 - Collaboration Chatlog report
- 10 - Failed Sessions report
- 12 - External Collaboration Chatlog report
- 13 - Customer Survey Issuance report
- 15 - Rebooting/Reconnecting report
- 16 - Transferred Sessions - Extended report

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the report type succeeded.
NOTLOGGEDIN	Retrieving the report type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportArea_v6` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportArea_v6.aspx?  
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
getReportArea_v6.aspx">
```

```
<input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getReportArea_v6.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//define parameters
$reportarams = array (
'sAuthCode' => $sAuthCode
);

//login
$getReportArea_v6Result = $soapclient->getReportArea($reportareaparams);

//print the result
echo "<b>getReportArea_v6 full response.</b><br />"; //formatting
print_r($getReportArea_v6Result);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
```



```

        string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
        actionName.aspx? for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?"
        email="
        + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
HttpWebRequest oReqReportArea =
(HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
+ "getReportArea_v6.aspx?authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpWebResponse oRespReportArea =
(HttpWebResponse) oReqReportArea.GetResponse();
string sRespReportArea = new
StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("getReportArea_v6 result: " + sRespReportArea + "<br />");
//You can customize the response

```

```

%>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getReportArea_v6 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT_V6 eReportArea;

                WebServiceClients.getReportAreaRet oGetReportArea =
                proxy.getReportArea_v6(sAuthCode, out eReportArea);
                Response.Write(oGetReportArea + "<br />");

                Response.Write(eReportArea);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

getReportArea_v7

Displays the report type that is currently set for retrieval. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the *Rescue Administration Center User Guide*.

Changes in Version 7 of getReportArea

The following report area has been introduced in the `getReportArea_v7` method:

Failed Sessions - Extended report This API method retrieves reports about failed sessions.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK AREA:17
```

AREA: Report area

The following report areas are available:

- 0 - Session report
- 1 - Customer Survey report
- 2 - Login report
- 3 - Missed Sessions report
- 4 - Performance report
- 5 - Chat Log report
- 6 - Custom Fields report
- 7 - Transferred Sessions report
- 8 - Technician Survey report
- 9 - Collaboration Chatlog report
- 10 - Failed Sessions report
- 12 - External Collaboration Chatlog report
- 13 - Customer Survey Issuance report
- 15 - Rebooting/Reconnecting report
- 16 - Transferred Sessions - Extended report
- 17 - Failed Sessions - Extended report

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.

Displayed Return Value	Description
OK	Retrieving the report type succeeded.
NOTLOGGEDIN	Retrieving the report type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportArea_v7` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportArea_v7.aspx?
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReportArea_v7.aspx">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getReportArea_v7.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting
```

```

$sAuthCode = "4ahx...80u0";

//define parameters
$reportarams = array (
'sAuthCode' => $sAuthCode
);

//login
$getReportArea_v7Result = $soapclient->getReportArea($reportareaparams);

//print the result
echo "<b>getReportArea_v7 full response.</b><br />"; //formatting
print_r($getReportArea_v7Result);
echo "<br /><br />"; //formatting

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "4ahx...80u0";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?"
    email="
    + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");
}

```

```

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
  cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
HttpRequest oReqReportArea =
  (HttpRequest) System.Net.WebRequest.Create(sEndpoint
+ "getReportArea_v7.aspx?authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
  (HttpResponse) oReqReportArea.GetResponse();
string sRespReportArea = new
  StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("getReportArea_v7 result: " + sRespReportArea + "<br />");
  //You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Rescue API getReportArea_v7 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
  public partial class SOAP : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      try
      {
        WebServiceClients.API proxy = new WebServiceClients.API();

```

```

        proxy.CookieContainer = new CookieContainer();

        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
        REPORT_V7 eReportArea;

        WebServiceClients.getReportAreaRet oGetReportArea =
proxy.getReportArea_v7(sAuthCode, out eReportArea);
        Response.Write(oGetReportArea + "<br />");

        Response.Write(eReportArea);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

getReportArea_v8

Displays the report type that is currently set for retrieval. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 8 of getReportArea

The following report area has been introduced in the `getReportArea_v8` method:

Audit report This API method retrieves reports about actions taken by Administrators on items of the Organization Tree.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK AREA:18
```

AREA: Report area

The following report areas are available:

- 0 - Session report
- 1 - Customer Survey report
- 2 - Login report
- 3 - Missed Sessions report
- 4 - Performance report
- 5 - Chat Log report

- 6 - Custom Fields report
- 7 - Transferred Sessions report
- 8 - Technician Survey report
- 9 - Collaboration Chatlog report
- 10 - Failed Sessions report
- 12 - External Collaboration Chatlog report
- 13 - Customer Survey Issuance report
- 15 - Rebooting/Reconnecting report
- 16 - Transferred Sessions - Extended report
- 17 - Failed Sessions - Extended report
- 18 - Audit report

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the report type succeeded.
NOTLOGGEDIN	Retrieving the report type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportArea_v8` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportArea_v8.aspx?
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReportArea_v8.aspx">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getReportArea_v8.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//define parameters
$reportarams = array (
'sAuthCode' => $sAuthCode
);

//login
$getReportArea_v8Result = $soapclient->getReportArea($reportareaparams);

//print the result
echo "<b>getReportArea_v8 full response.</b><br />"; //formatting
print_r($getReportArea_v8Result);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "4ahx...80u0";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?
email="
+ sEmail + "&pwd=" + sPwd);
```

```

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

        string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
        Response.Write("Login result: " + sResp + "<br />"); //You can customize
        the response

        /*
        //debug cookies
        foreach (Cookie cook in oResp.Cookies)
        {
            Response.Write("Cookie:" + "<br />");
            Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br />");
            Response.Write("Domain: " + cook.Domain + "<br />");
            Response.Write("Path: " + cook.Path + "<br />");
            Response.Write("Port: " + cook.Port + "<br />");
            Response.Write("Secure: " + cook.Secure + "<br />");

            Response.Write("When issued: " + cook.TimeStamp + "<br />");
            Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
                cook.Expired + "<br />");
            Response.Write("Don't save: " + cook.Discard + "<br />");
            Response.Write("Comment: " + cook.Comment + "<br />");
            Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
            Response.Write("Version: RFC " + cook.Version + "<br />");

            // Show the string representation of the cookie.
            Response.Write("String: " + cook.ToString());
        }*/

        //add cookies to cookie container
        CookieContainer sessioncookie = oReq.CookieContainer;

        //get the ReportArea info
        HttpRequest oReqReportArea =
            (HttpRequest)System.Net.WebRequest.Create(sEndpoint
            + "getReportArea_v8.aspx?authcode=" + sAuthCode);
        oReqReportArea.CookieContainer = sessioncookie;

        HttpWebResponse oRespReportArea =
            (HttpWebResponse)oReqReportArea.GetResponse();
        string sRespReportArea = new
            StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
        Response.Write("getReportArea_v8 result: " + sRespReportArea + "<br />");
        //You can customize the response

        %>

        <html xmlns="http://www.w3.org/1999/xhtml" >
        <head id="Head1" runat="server">
            <title>Rescue API getReportArea_v8 Test</title>
        </head>
        <body>

        </body>
        </html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT_V8 eReportArea;

                WebServiceClients.getReportAreaRet oGetReportArea =
                proxy.getReportArea_v8(sAuthCode, out eReportArea);
                Response.Write(oGetReportArea + "<br />");

                Response.Write(eReportArea);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

setReportArea

Sets the type of report to retrieve with the `getReport` method. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
area	The report type that you want to retrieve. Required. The following report areas are available: <ul style="list-style-type: none">• 0 - Session report• 1 - Customer Survey report• 2 - Login report• 3 - Missed Sessions report• 4 - Performance report• 5 - Chat Log report• 6 - Custom Fields report• 7 - Transferred Sessions report• 8 - Technician Survey report• 9 - Collaboration Chatlog report• 10 - Failed Sessions report
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the report type succeeded.
NOTLOGGEDIN	Setting the report type failed because the current user is no longer logged in.
INVALIDPARAM_REPORTAREA	The specified ID is not the ID of an existing report type.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportArea` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportArea.aspx?area=3
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea.aspx">
    <input name="area" value="3">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setReportArea>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'someemail@domain.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//define parameters
$reportarams = array (
'eReportArea' => '3',
'sAuthCode' => $sAuthCode
);

//login
$setReportAreaResult = $soapclient->setReportArea($reportareaparams);

//print the result
echo "<b>setReportArea full response.</b><br />"; //formatting
print_r($setReportAreaResult);
echo "<br /><br />"; //formatting

?>
```

C# with HttpRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "";

        //set up the request
        HttpRequest oReq =
        (HttpRequest)System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpResponse oResp = (HttpResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
        Response.Write("Cookie:" + "<br />");
        Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
        >");
        Response.Write("Domain: " + cook.Domain + "<br />");
        Response.Write("Path: " + cook.Path + "<br />");
        Response.Write("Port: " + cook.Port + "<br />");
        Response.Write("Secure: " + cook.Secure + "<br />");

        Response.Write("When issued: " + cook.TimeStamp + "<br />");
        Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
        cook.Expired + "<br />");
        Response.Write("Don't save: " + cook.Discard + "<br />");
        Response.Write("Comment: " + cook.Comment + "<br />");
        Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
        Response.Write("Version: RFC " + cook.Version + "<br />");

        // Show the string representation of the cookie.
        Response.Write("String: " + cook.ToString());
    }*/

    //add cookies to cookie container
    CookieContainer sessioncookie = oReq.CookieContainer;
```

```

//get the ReportArea info
string sReportArea = "3";
HttpRequest oReqReportArea =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
    + "setReportArea.aspx?area=" + sReportArea + "&authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
    (HttpResponse) oReqReportArea.GetResponse();
string sRespReportArea = new
    StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("setReportArea result: " + sRespReportArea + "<br />"); //
You can
customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setReportArea Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Net;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT eReportArea = REPORT.CHAT_LOG;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
                sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.setReportAreaRet oSetArea =
                proxy.setReportArea
                (eReportArea, sAuthCode);
                Response.Write(oSetArea);
            }
            catch (Exception ex)
            {

```


Displayed Return Value	Description
NOTLOGGEDIN	Setting the report type failed because the current user is no longer logged in.
INVALIDPARAM_REPORTAREA	The specified ID is not the ID of an existing report type.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportArea_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportArea_v2.aspx?area=12
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea_v2.aspx">
  <input name="area" value="12">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=setReportArea_v2.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'someemail@domain.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
```

```

echo "<br /><br />"; //formatting

$sAuthCode = "";

//define parameters
$reportarams = array (
'eReportArea' => '12',
'sAuthCode' => $sAuthCode
);

//login
$setReportArea_v2Result = $soapclient->setReportArea_v2($reportareaparams);

//print the result
echo "<b>setReportArea full response.</b><br />"; //formatting
print_r($setReportArea_v2Result);
echo "<br /><br />"; //formatting
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    }
    */

```

```

Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
string sReportArea = "12";
HttpRequest oReqReportArea =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
        + "setReportArea_v2.aspx?area=" + sReportArea + "&authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
    (HttpResponse) oReqReportArea.GetResponse();
string sRespReportArea = new
    StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("setReportArea result: " + sRespReportArea + "<br />"); //
You can
customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setReportArea_v2 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Net;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {

```

```

WebServiceClients.API proxy = new WebServiceClients.API();
proxy.CookieContainer = new CookieContainer();

string sEmail = "some@email.com";
string sPwd = "secretPassword";
string sAuthCode = "4ahx...80u0";
REPORT_V2 eReportArea = REPORT_V2.CHAT_LOG;

WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

Response.Write(oLogin + "<br />");

WebServiceClients.setReportAreaRet oSetArea =
proxy.setReportArea_v2
(eReportArea, sAuthCode);
    Response.Write(oSetArea);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

setReportArea_v3

Sets the type of report to retrieve with the `getReport` method. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 3 of setReportArea

The following report area has been introduced in the `setReportArea_v3` method:

Customer Survey Issuance report	Sets to retrieve a report about whether the closing technician issued the customer survey at the end of the Rescue session.
--	---

Input Parameters

Element	Description
area	<p>The report type that you want to retrieve. Required.</p> <p>The following report areas are available:</p> <ul style="list-style-type: none">• 0 - Session report• 1 - Customer Survey report• 2 - Login report• 3 - Missed Sessions report• 4 - Performance report• 5 - Chat Log report• 6 - Custom Fields report• 7 - Transferred Sessions report• 8 - Technician Survey report• 9 - Collaboration Chatlog report• 10 - Failed Sessions report• 12 - External Collaboration Chatlog report• 13 - Customer Survey Issuance report
authcode	<p>The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.</p>

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the report type succeeded.
NOTLOGGEDIN	Setting the report type failed because the current user is no longer logged in.
INVALIDPARAM_REPORTAREA	The specified ID is not the ID of an existing report type.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportArea_v3` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportArea_v3.aspx?area=13&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea_v3.aspx">
  <input name="area" value="13">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=setReportArea_v3.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'someemail@domain.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//define parameters
$reportarams = array (
'eReportArea' => '13',
'sAuthCode' => $sAuthCode
);

//login
$setReportArea_v3Result = $soapclient->setReportArea_v3($reportareaparams);

//print the result
echo "<b>setReportArea full response.</b><br />"; //formatting
print_r($setReportArea_v3Result);
echo "<br /><br />"; //formatting
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>
```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?"
    email="
    + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
string sReportArea = "13";
HttpWebRequest oReqReportArea =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
    + "setReportArea_v3.aspx?area=" + sReportArea + "&authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

```

```

HttpWebResponse oRespReportArea =
    (HttpWebResponse)oReqReportArea.GetResponse();
string sRespReportArea = new
    StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("setReportArea result: " + sRespReportArea + "<br />"); //
You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setReportArea_v3 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Net;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT_V3 eReportArea = REPORT_V3.CHAT_LOG;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.setReportAreaRet oSetArea =
proxy.setReportArea_v3(eReportArea, sAuthCode);
                Response.Write(oSetArea);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

setReportArea_v5

Sets the type of report to retrieve with the `getReport` method. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the *Rescue Administration Center User Guide*.

Changes in Version 5 of setReportArea

The following report area has been introduced in the `setReportArea_v5` method:

**Rebooting/
Reconnecting
report** Sets to retrieve a report about rebooting/reconnecting events.

Input Parameters

Element	Description
area	The report type that you want to retrieve. Required. The following report areas are available: <ul style="list-style-type: none">• 0 - Session report• 1 - Customer Survey report• 2 - Login report• 3 - Missed Sessions report• 4 - Performance report• 5 - Chat Log report• 6 - Custom Fields report• 7 - Transferred Sessions report• 8 - Technician Survey report• 9 - Collaboration Chatlog report• 10 - Failed Sessions report• 12 - External Collaboration Chatlog report• 13 - Customer Survey Issuance report• 14 - Daily Performance report• 15 - Rebooting/Reconnecting report
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the report type succeeded.
NOTLOGGEDIN	Setting the report type failed because the current user is no longer logged in.

Displayed Return Value	Description
INVALIDPARAM_REPORTAREA	The specified ID is not the ID of an existing report type.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportArea_v5` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportArea_v5.aspx?area=15
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea_v5.aspx">
  <input name="area" value="15">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=setReportArea_v5.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'someemail@domain.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";
```

```

//define parameters
$reportarams = array (
'eReportArea' => '15',
'sAuthCode' => $sAuthCode
);

//login
$setReportArea_v5Result = $soapclient->setReportArea_v5($reportareaparams);

//print the result
echo "<b>setReportArea full response.</b><br />"; //formatting
print_r($setReportArea_v5Result);
echo "<br /><br />"; //formatting
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?"
    email="
    + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");
    }
    */

```

```

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
  cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
string sReportArea = "15";
HttpRequest oReqReportArea =
  (HttpRequest) System.Net.WebRequest.Create(sEndpoint
+ "setReportArea_v5.aspx?area=" + sReportArea + "&authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
  (HttpResponse) oReqReportArea.GetResponse();
string sRespReportArea = new
  StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("setReportArea result: " + sRespReportArea + "<br />"); //
You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Rescue API setReportArea_v5 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Net;
using apiSamples.WebServiceClients;

namespace apiSamples
{
  public partial class SOAP : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      try
      {
        WebServiceClients.API proxy = new WebServiceClients.API();
        proxy.CookieContainer = new CookieContainer();

        string sEmail = "some@email.com";

```

```

        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
        REPORT_V5 eReportArea = REPORT_V5.CHAT_LOG;

        WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

        Response.Write(oLogin + "<br />");

        WebServiceClients.setReportAreaRet oSetArea =
proxy.setReportArea_v5(eReportArea, sAuthCode);
        Response.Write(oSetArea);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

setReportArea_v6

Sets the type of report to retrieve with the `getReport` method. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 6 of setReportArea

The following report area has been introduced in the `setReportArea_v6` method:

Transferred Sessions - Extended report Sets to retrieve a report about transferred sessions.

Input Parameters

Element	Description
area	<p>The report type that you want to retrieve. Required.</p> <p>The following report areas are available:</p> <ul style="list-style-type: none">• 0 - Session report• 1 - Customer Survey report• 2 - Login report• 3 - Missed Sessions report• 4 - Performance report• 5 - Chat Log report• 6 - Custom Fields report• 7 - Transferred Sessions report• 8 - Technician Survey report• 9 - Collaboration Chatlog report• 10 - Failed Sessions report• 12 - External Collaboration Chatlog report• 13 - Customer Survey Issuance report• 14 - Daily Performance report• 15 - Rebooting/Reconnecting report• 16 - Transferred Sessions - Extended report
authcode	<p>The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.</p>

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the report type succeeded.
NOTLOGGEDIN	Setting the report type failed because the current user is no longer logged in.
INVALIDPARAM_REPORTAREA	The specified ID is not the ID of an existing report type.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportArea_v6` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportArea_v6.aspx?area=16
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea_v6.aspx">
    <input name="area" value="16">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=setReportArea_v6.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'someemail@domain.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//define parameters
$reportareparams = array (
'eReportArea' => '16',
'sAuthCode' => $sAuthCode
);

//login
$setReportArea_v6Result = $soapclient->setReportArea_v6($reportareparams);

//print the result
echo "<b>setReportArea full response.</b><br />"; //formatting
print_r($setReportArea_v6Result);
echo "<br /><br />"; //formatting
?>
```

C# with HttpRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "";

    //set up the request
    HttpRequest oReq =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?
    email="
    + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpResponse oResp = (HttpResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");

    Response.Write("When issued: " + cook.TimeStamp + "<br />");
    Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
    Response.Write("Don't save: " + cook.Discard + "<br />");
    Response.Write("Comment: " + cook.Comment + "<br />");
    Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
    Response.Write("Version: RFC " + cook.Version + "<br />");

    // Show the string representation of the cookie.
    Response.Write("String: " + cook.ToString());
    }*/

    //add cookies to cookie container
    CookieContainer sessioncookie = oReq.CookieContainer;
```



```

//get the ReportArea info
string sReportArea = "16";
HttpRequest oReqReportArea =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
+ "setReportArea_v6.aspx?area=" + sReportArea + "&authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
    (HttpResponse)oReqReportArea.GetResponse();
string sRespReportArea = new
    StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("setReportArea result: " + sRespReportArea + "<br />"); //
You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setReportArea_v6 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Net;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT_V6 eReportArea = REPORT_V6.CHAT_LOG;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.setReportAreaRet oSetArea =
proxy.setReportArea_v6(eReportArea, sAuthCode);
                Response.Write(oSetArea);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

```

    }
  }
}

```

setReportArea_v7

Sets the type of report to retrieve with the `getReport` method. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 7 of setReportArea

The following report area has been introduced in the `setReportArea_v7` method:

Failed Sessions - Extended report Sets to retrieve a report about failed sessions.

Input Parameters

Element	Description
area	<p>The report type that you want to retrieve. Required.</p> <p>The following report areas are available:</p> <ul style="list-style-type: none"> • 0 - Session report • 1 - Customer Survey report • 2 - Login report • 3 - Missed Sessions report • 4 - Performance report • 5 - Chat Log report • 6 - Custom Fields report • 7 - Transferred Sessions report • 8 - Technician Survey report • 9 - Collaboration Chatlog report • 10 - Failed Sessions report • 12 - External Collaboration Chatlog report • 13 - Customer Survey Issuance report • 14 - Daily Performance report • 15 - Rebooting/Reconnecting report • 16 - Transferred Sessions - Extended report • 17 - Failed Sessions - Extended report
authcode	<p>The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.</p>

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the report type succeeded.
NOTLOGGEDIN	Setting the report type failed because the current user is no longer logged in.
INVALIDPARAM_REPORTAREA	The specified ID is not the ID of an existing report type.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportArea_v7` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportArea_v7.aspx?area=17
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea_v7.aspx">
  <input name="area" value="17">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=setReportArea_v7.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'someemail@domain.com',
'sPassword' => 'secretPassword'
);

//login
```

```

$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//define parameters
$reportarams = array (
'eReportArea' => '17',
'sAuthCode' => $sAuthCode
);

//login
$setReportArea_v7Result = $soapclient->setReportArea_v7($reportareaparams);

//print the result
echo "<b>setReportArea full response.</b><br />"; //formatting
print_r($setReportArea_v7Result);
echo "<br /><br />"; //formatting
?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "";

    //set up the request
    HttpWebRequest oReq =
    (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?
    email="
    + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)

```

```

{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportArea info
string sReportArea = "17";
HttpRequest oReqReportArea =
(HttpRequest)System.Net.WebRequest.Create(sEndpoint
+ "setReportArea_v7.aspx?area=" + sReportArea + "&authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
(HttpResponse)oReqReportArea.GetResponse();
string sRespReportArea = new
StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("setReportArea result: " + sRespReportArea + "<br />"); //
You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
<title>Rescue API setReportArea_v7 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Net;
using apiSamples.WebServiceClients;

namespace apiSamples
{
public partial class SOAP : System.Web.UI.Page
{

```

```

protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        WebServiceClients.API proxy = new WebServiceClients.API();
        proxy.CookieContainer = new CookieContainer();

        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
        REPORT_V7 eReportArea = REPORT_V7.CHAT_LOG;

        WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
        Response.Write(oLogin + "<br />");

        WebServiceClients.setReportAreaRet oSetArea =
proxy.setReportArea_v7(eReportArea, sAuthCode);
        Response.Write(oSetArea);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}

```

setReportArea_v8

Sets the type of report to retrieve with the `getReport` method. You can set the report area in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Changes in Version 7 of setReportArea

The following report area has been introduced in the `setReportArea_v8` method:

Failed Sessions Sets to retrieve a report about failed sessions.

**- Extended
report**

Input Parameters

Element	Description
area	<p>The report type that you want to retrieve. Required.</p> <p>The following report areas are available:</p> <ul style="list-style-type: none">• 0 - Session report• 1 - Customer Survey report• 2 - Login report• 3 - Missed Sessions report• 4 - Performance report• 5 - Chat Log report• 6 - Custom Fields report• 7 - Transferred Sessions report• 8 - Technician Survey report• 9 - Collaboration Chatlog report• 10 - Failed Sessions report• 12 - External Collaboration Chatlog report• 13 - Customer Survey Issuance report• 14 - Daily Performance report• 15 - Rebooting/Reconnecting report• 16 - Transferred Sessions - Extended report• 17 - Failed Sessions - Extended report• 18 - Audit report
authcode	<p>The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.</p>

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the report type succeeded.
NOTLOGGEDIN	Setting the report type failed because the current user is no longer logged in.
INVALIDPARAM_REPORTAREA	The specified ID is not the ID of an existing report type.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportArea_v8` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportArea_v8.aspx?area=18
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportArea_v8.aspx">
  <input name="area" value="18">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=setReportArea_v8.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'someemail@domain.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//define parameters
$reportarams = array (
'eReportArea' => '18',
'sAuthCode' => $sAuthCode
);

//login
$setReportArea_v8Result = $soapclient->setReportArea_v8($reportareaparams);

//print the result
echo "<b>setReportArea full response.</b><br />"; //formatting
print_r($setReportArea_v8Result);
echo "<br /><br />"; //formatting
?>
```

C# with HttpRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx? for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "";

    //set up the request
    HttpRequest oReq =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint + "login.aspx" + "?"
    email="
    + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
    oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpResponse oResp = (HttpResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br />");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");

    Response.Write("When issued: " + cook.Timestamp + "<br />");
    Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
    Response.Write("Don't save: " + cook.Discard + "<br />");
    Response.Write("Comment: " + cook.Comment + "<br />");
    Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
    Response.Write("Version: RFC " + cook.Version + "<br />");

    // Show the string representation of the cookie.
    Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;
```

```

//get the ReportArea info
string sReportArea = "18";
HttpRequest oReqReportArea =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
+ "setReportArea_v8.aspx?area=" + sReportArea + "&authcode=" + sAuthCode);
oReqReportArea.CookieContainer = sessioncookie;

HttpResponse oRespReportArea =
    (HttpResponse) oReqReportArea.GetResponse();
string sRespReportArea = new
    StreamReader(oRespReportArea.GetResponseStream()).ReadToEnd();
Response.Write("setReportArea result: " + sRespReportArea + "<br />"); //
You can customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setReportArea_v8 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Net;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORT_V8 eReportArea = REPORT_V8.CHAT_LOG;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.setReportAreaRet oSetArea =
proxy.setReportArea_v8(eReportArea, sAuthCode);
                Response.Write(oSetArea);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

```
}  
}  
}
```

getDateFormat

Retrieves the date format for reporting. You can set the report date format in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the *Rescue Administration Center User Guide*.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK DATEFORMAT:MMDDYY
```

DATEFORMAT: Possible values of the date format are MMDDYY and DDMMYY

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the date format succeeded.
NOTLOGGEDIN	Retrieving the date format failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getDateFormat` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getDateFormat.aspx?  
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getDateFormat.aspx">
  <input name="authcode" value="4ahx...80u0">
</form>
```

C# with SOAP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";

                //Login
                WebServiceClients.loginRet loginResult =
                proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                DATEFORMAT dateformat;

                //Get date format
                WebServiceClients.getDateFormatRet getDateFormatResult =
                proxy.getDateFormat(sAuthCode, out dateformat);

                Response.Write(dateformat);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}
```

setDateFormat

Sets the date format for reporting. You can set the report date format in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
dateformat: "MMDDYY" "DDMMYY"	Defines the date format that can be MMDDYY or DDMMYY. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the date format succeeded.
NOTLOGGEDIN	Setting the date format failed because the current user is no longer logged in.
INVALIDPARAM_DATEFORMAT	The specified date format is incorrect.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setDateFormat` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setDateFormat.aspx?dateformat=MMDDYY
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setDateFormat.aspx">
  <input name="dateformat" value="MMDDYY">
  <input name="authcode" value="4ahx...80u0">
</form>
```

C# with SOAP

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                //Set up proxy
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";

                //Login
                WebServiceClients.loginRet loginResult =
                proxy.login(sEmail, sPwd);
                Response.Write(loginResult + "<br />");

                DATEFORMAT dateformat = DATEFORMAT.DDMMYY;

                //Get result
                WebServiceClients.setDateFormatRet setDateFormatResult =
                proxy.setDateFormat(dateformat, sAuthCode);

                Response.Write(setDateFormatResult);
            }
            catch (Exception ex)
            {
                lblError.Text = ex.Message;
            }
        }
    }
}
```

getDelimiter

Retrieves the delimiter characters for text-based output. You cannot define a delimiter character in the Rescue Administration Center.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK DELIMITER: |
```

DELIMITER: Defines the delimiter used in reports that are generated with the getReport method

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the delimiter character succeeded.
NOTLOGGEDIN	Retrieving the delimiter character failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the getDelimiter method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getDelimiter.aspx?authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/getDelimiter.aspx">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getDelimiter>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//set up array for request
$delimiterparams = array(
'sAuthCode' => $sAuthCode
);

$getDelimiterResponse = $soapclient->getDelimiter($delimiterparams);

//print the delimiter results
print_r("<b>getDelimiter full response.</b><br />");
print_r($getDelimiterResponse);

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
```



```

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Delimiter info
HttpWebRequest oReqDelimiter =
    (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
    + "getDelimiter.aspx?authcode=" + sAuthCode);
oReqDelimiter.CookieContainer = sessioncookie;

HttpWebResponse oRespDelimiter =
    (HttpWebResponse)oReqDelimiter.GetResponse();
string sRespDelimiter = new
    StreamReader(oRespDelimiter.GetResponseStream()).ReadToEnd();
Response.Write("getDelimiter result: " + sRespDelimiter + "<br />"); //You
can customize
the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getDelimiter Test</title>

```

```
</head>
<body>

</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                string sDelimiter = "";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.getDelimiterRet oGetDelimiter =
proxy.getDelimiter
(sAuthCode, out sDelimiter);
                Response.Write(oGetDelimiter + "<br />");

                Response.Write(sDelimiter);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

setDelimiter

Sets the delimiter characters for text-based output which cannot be of zero length. You cannot define a delimiter character in the Rescue Administration Center.

Input Parameters

Element	Description
delimiter	The delimiter character sequence. Required, default is ' ' and it cannot be of zero length.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the delimiter character succeeded.
NOTLOGGEDIN	Setting the delimiter character failed because the current user is no longer logged in.
INVALIDPARAM_DELIMITER	The specified delimiter character is invalid.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setDelimiter` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setDelimiter.aspx?delimiter=-&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/setDelimiter.aspx">  
  <input name="delimiter" value="-">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setDelimiter>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//define parameters
$delimiterparams = array (
'sDelimiter' => '-',
'sAuthCode' => $sAuthCode
);

//set the delimiter
$setDelimiterResult = $soapclient->setDelimiter($delimiterparams);

//print the result
echo "<b>setDelimiter full response.</b><br />"; //formatting
print_r($setDelimiterResult);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
```

```

string sPwd = "secretPassword";
string sAuthCode = "4ahx...80u0";

//set up the request
HttpWebRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

//create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

//get the response
HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br />");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Delimiter info
string sDelimiter = "-";
HttpWebRequest oReqDelimiter =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "setDelimiter.aspx?delimiter=" + sDelimiter + "&authcode=" + sAuthCode);
oReqDelimiter.CookieContainer = sessioncookie;

HttpWebResponse oRespDelimiter =
(HttpWebResponse)oReqDelimiter.GetResponse();
string sRespDelimiter = new
StreamReader(oRespDelimiter.GetResponseStream()).ReadToEnd();
Response.Write("setDelimiter result: " + sRespDelimiter + "<br />"); //You
can
customize the response

```

```

%>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Rescue API setDelimiter Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                string sDelimiter = "-";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
                Response.Write(oLogin + "<br />");

                WebServiceClients.setDelimiterRet oSetDelimiter =
proxy.setDelimiter
(sDelimiter, sAuthCode);
                Response.Write(oSetDelimiter);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

getOutput

Retrieves the output type currently set for reporting. You can set the report output type in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the *Rescue Administration Center User Guide*.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK OUTPUT:TEXT
```

OUTPUT: The output of your generated reports. Possible values are TEXT and XML

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the report output type succeeded.
NOTLOGGEDIN	Retrieving the report output type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getOutput` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getOutput.aspx?authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/getOutput.aspx">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getOutput>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//set up array for getOutput request
$getOutputParams = array(
'sAuthCode' => $sAuthCode
);

$getOutputResult = $soapclient->getOutput($getOutputParams);

//print out the crrent output format
print_r("<b>getOutput full response.</b><br />");
print_r($getOutputResult);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
```



```

string sAuthCode = "";

//set up the request
HttpRequest oReq =
(HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

//create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

//get the response
HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Output info
HttpRequest oReqOutput =
(HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
+ "getOutput.aspx?authcode=" + sAuthCode);
oReqOutput.CookieContainer = sessioncookie;

HttpWebResponse oRespOutput = (HttpWebResponse) oReqOutput.GetResponse();
string sRespOutput = new
StreamReader(oRespOutput.GetResponseStream()).ReadToEnd();
Response.Write("getOutput result: " + sRespOutput + "<br />"); //You can
customize the
response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >

```

```
<head id="Head1" runat="server">
  <title>Rescue API getOutput Test</title>
</head>
<body>

</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                OUTPUT eOutput;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
                Response.Write(oLogin + "<br />");

                WebServiceClients.getOutputRet oGetOutput =
proxy.getOutput(sAuthCode,
out eOutput);
                Response.Write(oGetOutput + "<br />");

                Response.Write(eOutput);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

setOutput

Sets the output type for reporting. You can set the report output type in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
output	The output of the report that can be TEXT or XML. Required, default output type is TEXT.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the report output type succeeded.
NOTLOGGEDIN	Setting the report output type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the setOutput method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setOutput.aspx?  
output=XML&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
setOutput.aspx">  
  <input name="output" value="XML">  
  <input name="authcode" value="4ahx...80u0">  
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setOutput>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define variables
$email = "some@email.com";
$password = "secretPassword";

//define parameters
$loginparams = array (
'sEmail' => $email,
'sPassword' => $password);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//switch to XML for easier formatting of output
$output = array(
'eOutput' => 'XML',
'sAuthCode' => $sAuthCode
);

$outputResponse = $soapclient->setOutput($output);
print_r("<b>setOutput Full response.</b> <br />");
print_r($outputResponse);
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";
```

```

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Output info
string sOutput = "XML";
HttpWebRequest oReqOutput =
(HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
+ "setOutput.aspx?output=" + sOutput + "&authcode=" + sAuthCode);
oReqOutput.CookieContainer = sessioncookie;

HttpWebResponse oRespOutput = (HttpWebResponse)oReqOutput.GetResponse();
string sRespOutput = new
StreamReader(oRespOutput.GetResponseStream()).ReadToEnd();
Response.Write("setOutput result: " + sRespOutput + "<br />"); //You can
customize the
response

%>
<html xmlns="http://www.w3.org/1999/xhtml" >

```

```
<head id="Head1" runat="server">
  <title>Rescue API setOutput Test</title>
</head>
<body>

</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                OUTPUT eOutput = OUTPUT.XML;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
                Response.Write(oLogin + "<br />");

                WebServiceClients.setOutputRet oSetOutput =
proxy.setOutput(eOutput,
sAuthCode);
                Response.Write(oSetOutput + "<br />");
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

getReportDate

Retrieves the start and end date of reports. You can set the report date range in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the *Rescue Administration Center User Guide*.

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK BEGINDATE:7/15/2012 ENDDATE:8/15/2012
```

BEGINDATE: The start date of the reporting period in M/D/YYYY format
ENDDATE: The end date of the reporting period in M/D/YYYY format

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the start and end dates of the report succeeded.
NOTLOGGEDIN	Retrieving the start and end dates of the report failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportDate` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportDate.aspx?  
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
getReportDate.aspx">
```

```
<input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getReportDate>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//set up array for request
$reportdateparams = array(
'sAuthCode' => $sAuthCode
);

$getReportDateResponse = $soapclient->getReportDate($reportdateparams);

//print the report date results
print_r("<b>getReportDate full response.</b><br />");
print_r($getReportDateResponse);

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
```



```

string sEmail = "some@email.com";
string sPwd = "secretPassword";
string sAuthCode = "4ahx...80u0";

//set up the request
HttpWebRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

//create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

//get the response
HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportDate info
HttpWebRequest oReqReportDate =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "getReportDate.aspx?authcode=" + sAuthCode);
oReqReportDate.CookieContainer = sessioncookie;

HttpWebResponse oRespReportDate =
(HttpWebResponse)oReqReportDate.GetResponse();
string sRespReportDate = new
StreamReader(oRespReportDate.GetResponseStream()).ReadToEnd();
Response.Write("getReportDate result: " + sRespReportDate + "<br />"); //
You can
customize the response

```

```

%>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getReportDate Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                string sBeginDate = "";
                string sEndDate = "";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.getReportDateRet oGetReportDate =
proxy.getReportDate
(sAuthCode, out sBeginDate, out sEndDate);
                Response.Write(oGetReportDate + "<br />");

                Response.Write("Begin date: " + sBeginDate + "<br />End
date: " + sEndDate);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

getReportDate_v2

Retrieves the start and end date of reports. You can set the report date range in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the *Rescue Administration Center User Guide*.

Changes in Version 2 of getReportDate

The following has been updated in the `getReportDate_v2` method:

- The data type of the output parameters has been changed from `string` to `dateTime`.
- The `begindate` and `enddate` values of the output have been changed to `begindatetime` and `enddatetime` respectively.

Input Parameters

Element	Description
<code>authcode</code>	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK  BEGINDATETIME:7/15/2012 2:23:45 PM ENDDATETIME:8/15/2012 11:59:59 PM
```

`BEGINDATETIME`: The start date and time of the report in M/D/YYYY H:MM:SS format

`ENDDATETIME`: The end date and time of the report in M/D/YYYY H:MM:SS format

Return Values

Displayed Return Value	Description
<code>ERROR</code>	An unspecified error occurred, such as timeout.
<code>OK</code>	Retrieving the start and end dates of the report succeeded.
<code>NOTLOGGEDIN</code>	Retrieving the start and end dates of the report failed because the current user is no longer logged in.
<code>INVALID_SECRETAUTHCODE</code>	The secret authentication code for the user is invalid.
<code>USER_DELETED_OR_DISABLED</code>	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportDate_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportDate_v2.aspx?
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReportDate_v2.aspx">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=getReportDate_v2.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
    $soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

    $loginparams = array (
        'sEmail' => 'some@email.com',
        'sPassword' => 'secretPassword'
    );

    $getreportdate_v2params = array (
        'sAuthCode' => '4ahx...80u0'
    );

    $loginResult = $soapclient->login($loginparams);

    print_r($loginResult);

    $getReportDate_v2Result = $soapclient-
>getReportDate_v2($getreportdate_v2params);

    print_r($getReportDate_v2Result);
?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
```

```

string sEmail = "some@email.com";
string sPwd = "secretPassword";
string sAuthCode = "4ahx...80u0";

//set up the request
HttpWebRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

//create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

//get the response
HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportDate info
HttpWebRequest oReqReportDate =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "getReportDate_v2.aspx?authcode=" + sAuthCode);
oReqReportDate.CookieContainer = sessioncookie;

HttpWebResponse oRespReportDate =
(HttpWebResponse)oReqReportDate.GetResponse();
string sRespReportDate = new
StreamReader(oRespReportDate.GetResponseStream()).ReadToEnd();
Response.Write("getReportDate result: " + sRespReportDate + "<br />"); //
You can
customize the response

```

```

%>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Rescue API getReportDate_v2 Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                DateTime dBeginDate = new DateTime();
                DateTime dEndDate = new DateTime();

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.getReportDateRet oGetReportDate =
proxy.getReportDate_v2
(sAuthCode, out dBeginDate, out dEndDate);
                Response.Write(oGetReportDate + "<br />");

                Response.Write("Begin date: " + dBeginDate + "<br />End
date: " + dEndDate);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

setReportDate

Sets the start and end date of reports. You can set the report date range in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the *Rescue Administration Center User Guide*.

Input Parameters

Element	Description
bdate	The start date of the report in M/D/YYYY format. Required.
edate	The end date of the report in M/D/YYYY format. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the start and end dates of the report succeeded.
NOTLOGGEDIN	Setting the start and end dates of the report failed because the current user is no longer logged in.
INVALIDFORMAT	The date format is invalid.
INVALIDDATERANGE	The end date is earlier than the start date of the report.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportDate` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportDate.aspx?bdate=12/1/2011&edate=12/31/2011&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportDate.aspx">
  <input name="bdate" value="12/1/2011">
  <input name="edate" value="12/31/2011">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setReportDate>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//define parameters
$dBeginDate = new DateTime('01/13/2015 08:00:00');
$dEndDate = new DateTime('01/16/2015 07:59:59');

$reportdateparams = array (
'dBeginDate' => $dBeginDate->format('c'),
'dEndDate' => $dEndDate->format('c'),
'sAuthCode' => $sAuthCode
);

//set the report date
$setReportDateResult = $soapclient->setReportDate($reportdateparams);

//print the result
echo "<b>setReportDate full response.</b><br />"; //formatting
print_r($setReportDateResult);
echo "<br /><br />"; //formatting

?>
```

C# with HttpRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

        //set up the request
        HttpRequest oReq =
        (HttpRequest) System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpResponse oResp = (HttpResponse) oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
        Response.Write("Cookie:" + "<br />");
        Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br />");
        Response.Write("Domain: " + cook.Domain + "<br />");
        Response.Write("Path: " + cook.Path + "<br />");
        Response.Write("Port: " + cook.Port + "<br />");
        Response.Write("Secure: " + cook.Secure + "<br />");

        Response.Write("When issued: " + cook.Timestamp + "<br />");
        Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
        cook.Expired + "<br />");
        Response.Write("Don't save: " + cook.Discard + "<br />");
        Response.Write("Comment: " + cook.Comment + "<br />");
        Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
        Response.Write("Version: RFC " + cook.Version + "<br />");

        // Show the string representation of the cookie.
        Response.Write("String: " + cook.ToString());
    }*/

    //add cookies to cookie container
    CookieContainer sessioncookie = oReq.CookieContainer;
```

```

//get the ReportDate info
string sBDate = "12/1/2011";
string sEDate = "12/31/2011";
HttpRequest oReqReportDate =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
    + "setReportDate.aspx?bdate=" + sBDate + "&edate=" + sEDate + "&authcode="
    + sAuthCode);
oReqReportDate.CookieContainer = sessioncookie;

HttpResponse oRespReportDate =
    (HttpResponse)oReqReportDate.GetResponse();
string sRespReportDate = new
    StreamReader(oRespReportDate.GetResponseStream()).ReadToEnd();
Response.Write("setReportDate result: " + sRespReportDate + "<br />"); //
You can
customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setReportDate Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretpassword";
                string sAuthCode = "";
                string sBeginDate = "12/1/2011";
                string sEndDate = "12/31/2011";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
                sPwd);

                Response.Write(oLogin + "<br />");
            }
        }
    }
}

```


Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the start and end dates of the report succeeded.
NOTLOGGEDIN	Setting the start and end dates of the report failed because the current user is no longer logged in.
INVALIDFORMAT	The date format is invalid.
INVALIDDATERANGE	The end date is earlier than the start date of the report.
INVALID_SECRETAUTHCODE	The authentication code is invalid.
USER_DELETED_OR_DISABLED	The user whose authentication code is provided is already deleted or disabled.

Sample Code

The following are examples for using the `setReportDate_v2` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportDate_v2.aspx?bdate=12/1/2011
8:00:00
&edate=12/31/2011 19:59:59&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setReportDate_v2.aspx">
  <input name="bdate" value="12/1/2011 8:00:00">
  <input name="edate" value="12/31/2011 19:59:59">
  <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit https://secure.logmeinrescue.com/api/API.asmx?op=setReportDate_v2.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
```

```

$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//define parameters
$dBeginDate = new DateTime('01/13/2015 08:00:00');
$dEndDate = new DateTime('01/16/2015 07:59:59');

$reportdateparams = array (
'dBeginDate' => $dBeginDate->format('c'),
'dEndDate' => $dEndDate->format('c'),
'sAuthCode' => $sAuthCode
);

//set the report date
$setReportDateResult = $soapclient->setReportDate_v2($reportdateparams);

//print the result
echo "<b>setReportDate full response.</b><br />"; //formatting
print_r($setReportDateResult);
echo "<br /><br />"; //formatting

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

```

```

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportDate info
DateTime dBDate = new DateTime(2011, 12, 1, 8, 0, 0);
DateTime dEDate = new DateTime(2011, 12, 31, 19, 59, 59);
HttpRequest oReqReportDate =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "setReportDate_v2.aspx?bdate=" + dBDate + "&edate=" + dEDate +
"&authcode=" + sAuthCode);
oReqReportDate.CookieContainer = sessioncookie;

HttpWebResponse oRespReportDate =
(HttpWebResponse)oReqReportDate.GetResponse();
string sRespReportDate = new
StreamReader(oRespReportDate.GetResponseStream()).ReadToEnd();
Response.Write("setReportDate result: " + sRespReportDate + "<br />"); //
You can
customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
<title>Rescue API setReportDate Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                DateTime dBeginDate = new DateTime(2011, 12, 1, 8, 0, 0);
                DateTime dEndDate = new DateTime(2011, 12, 31, 19, 59, 59);

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.setReportDateRet oSetDate =
proxy.setReportDate_v2
(dBeginDate, dEndDate, sAuthCode);
                Response.Write(oSetDate);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}
```

getReportTime

Retrieves the daily time range of reports. You can set the report time range in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK BEGINTIME:09:00 ENDTIME:18:00
```

BEGINTIME: The start time of the reporting period

ENDTIME: The end time of the reporting period

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the daily time range of the report succeeded.
NOTLOGGEDIN	Retrieving the daily time range of the report failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportTime` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportTime.aspx?  
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
getReportTime.aspx">
```



```
<input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getReportTime>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//set up array for request
$reporttimeparams = array(
'sAuthCode' => $sAuthCode
);

$getReportTimeResponse = $soapclient->getReportTime($reporttimeparams);

//print the report time results
print_r("<b>getReportTime full response.</b><br />");
print_r($getReportTimeResponse);

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
```

```

string sEmail = "some@email.com";
string sPwd = "secretPassword";
string sAuthCode = "4ahx...80u0";

//set up the request
HttpRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

//create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

//get the response
HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportTime info
HttpRequest oReqReportTime =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "getReportTime.aspx?authcode=" + sAuthCode);
oReqReportTime.CookieContainer = sessioncookie;

HttpWebResponse oRespReportTime =
(HttpWebResponse)oReqReportTime.GetResponse();
string sRespReportTime = new
StreamReader(oRespReportTime.GetResponseStream()).ReadToEnd();
Response.Write("getReportTime result: " + sRespReportTime + "<br />"); //
You can
customize the response

```

```

%>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Rescue API getReportTime Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                string sBeginTime = "";
                string sEndTime = "";

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

                Response.Write(oLogin + "<br />");

                WebServiceClients.getReportTimeRet oGetReportTime =
proxy.getReportTime
(sAuthCode, out sBeginTime, out sEndTime);
                Response.Write(oGetReportTime + "<br />");

                Response.Write("Begin time: " + sBeginTime + "<br />End
time: " + sEndTime);
            }
            catch (Exception ex)
            {
                Response.Write(ex.Message);
            }
        }
    }
}

```

setReportTime

Sets the daily time range of reports. You can set the report time range in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).



Note: Both `setReportTime` and `setReportDate_v2` can set the time of your report period. The difference is that `setReportDate_v2` sets the start and end time of your whole report period, whereas `setReportTime` sets a daily time range for every day of your report period. For example, you can define a report period between 12/1/2011 0:00:00 and 5/31/2012 23:59:59 with the `setReportDate_v2` method. Then restrict reporting every day from 9:00 to 17:00 with the `setReportTime` method.

Input Parameters

Element	Description
<code>btime</code>	The start time of the report in H:MM:SS format. For example, 8:30:00. Required.
<code>etime</code>	The end time of the report in H:MM:SS format. For example, 18:30:00. Required.
<code>authcode</code>	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the daily time range of the report succeeded.
NOTLOGGEDIN	Setting the daily time range of the report failed because the current user is no longer logged in.
INVALIDFORMAT	The time format is invalid.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setReportTime` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportTime.aspx?  
btime=8:30:00&etime=18:30:00  
&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
setReportTime.aspx">  
    <input name="btime" value="8:30:00">  
    <input name="etime" value="18:30:00">  
    <input name="authcode" value="4ahx...80u0">  
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setReportTime>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php  
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/  
api.asmx?wsdl");  
  
//define parameters  
$loginparams = array (  
    'sEmail' => 'some@email.com',  
    'sPassword' => 'secretPassword'  
);  
  
//login  
$loginResult = $soapclient->login($loginparams);  
  
//print the result  
echo "<b>Login full response.</b><br />"; //formatting  
print_r($loginResult);  
echo "<br /><br />"; //formatting  
  
$sAuthCode = "4ahx...80u0";  
  
//define parameters  
$reporttimeparams = array (  
    'sBeginTime' => '8:30:00',  
    'sEndTime' => '18:30:00',  
    'sAuthCode' => $sAuthCode  
);  
  
//set the report time  
$setReportTimeResult = $soapclient->setReportTime($reporttimeparams);  
  
//print the result  
echo "<b>setReportTime full response.</b><br />"; //formatting  
print_r($setReportTimeResult);  
echo "<br /><br />"; //formatting
```

?>

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");

    Response.Write("When issued: " + cook.TimeStamp + "<br />");
    Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
    Response.Write("Don't save: " + cook.Discard + "<br />");
    Response.Write("Comment: " + cook.Comment + "<br />");
    Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
    Response.Write("Version: RFC " + cook.Version + "<br />");

    // Show the string representation of the cookie.
    Response.Write("String: " + cook.ToString());
    }*/

    //add cookies to cookie container
```

```

CookieContainer sessioncookie = oReq.CookieContainer;

//get the ReportTime info
string sBTime = "8:30:00";
string sETime = "18:30:00";
HttpRequest oReqReportTime =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
    + "setReportTime.aspx?btime=" + sBTime + "&etime=" + sETime + "&authcode="
    + sAuthCode);
oReqReportTime.CookieContainer = sessioncookie;

HttpResponse oRespReportTime =
    (HttpResponse)oReqReportTime.GetResponse();
string sRespReportTime = new
    StreamReader(oRespReportTime.GetResponseStream()).ReadToEnd();
Response.Write("setReportTime result: " + sRespReportTime + "<br />"); //
You can
customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setReportTime Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "";
                string sBeginTime = "8:30:00";
                string sEndTime = "18:30:00";
            }
        }
    }
}

```

```

        WebServiceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);
        Response.Write(oLogin + "<br />");

        WebServiceClients.setReportTimeRet oSetTime =
proxy.setReportTime
(sBeginTime, sEndTime, sAuthCode);
        Response.Write(oSetTime);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

getTimeZone

Retrieves the time zone set for the report. This is calculated as the deviation, in minutes, from UTC for reporting. You can set the report time zone in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK TIMEZONE:0
```

TIMEZONE: The timezone calculated as the deviation, in minutes, from UTC for reporting.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving the specified time zone of the report succeeded.
NOTLOGGEDIN	Retrieving the specified time zone of the report failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getTimeZone` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getTimezone.aspx?authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/getTimezone.aspx">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getTimezone>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/api.asmx?wsdl");

//define parameters
$loginparams = array (
    'sEmail' => 'some@email.com',
    'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "4ahx...80u0";

//get current reporting timezone
$getTimeZoneParams = array(
    'sAuthCode' => $sAuthCode
);

$getTimeZoneResult = $soapclient->getTimeZone($getTimeZoneParams);

//print out the crrent time zone output
print_r("<b>setTimeZone full response.</b><br />");
print_r($getTimeZoneResult);
echo "<br /><br />"; //formatting
```

?>

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
    Response.Write("Cookie:" + "<br />");
    Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
    >");
    Response.Write("Domain: " + cook.Domain + "<br />");
    Response.Write("Path: " + cook.Path + "<br />");
    Response.Write("Port: " + cook.Port + "<br />");
    Response.Write("Secure: " + cook.Secure + "<br />");

    Response.Write("When issued: " + cook.TimeStamp + "<br />");
    Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
    cook.Expired + "<br />");
    Response.Write("Don't save: " + cook.Discard + "<br />");
    Response.Write("Comment: " + cook.Comment + "<br />");
    Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
    Response.Write("Version: RFC " + cook.Version + "<br />");

    // Show the string representation of the cookie.
    Response.Write("String: " + cook.ToString());
}*/
```

```
//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Timezone info
HttpRequest oReqTimezone =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
    + "getTimezone.aspx?authcode=" + sAuthCode);
oReqTimezone.CookieContainer = sessioncookie;

HttpResponse oRespTimezone =
    (HttpResponse)oReqTimezone.GetResponse();
string sRespTimezone = new
    StreamReader(oRespTimezone.GetResponseStream()).ReadToEnd();
Response.Write("getTimezone result: " + sRespTimezone + "<br />"); //You
    can customize
    the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getTimezone Test</title>
</head>
<body>

</body>
</html>
```

C# with SOAP

The example values shown must be replaced with actual values.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                short sTimezone;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
                sPwd);

                Response.Write(oLogin + "<br />");
            }
        }
    }
}
```

```

        WebServiceClients.getTimezoneRet oGetTimezone =
        proxy.getTimezone
        (sAuthCode, out sTimezone);
        Response.Write(oGetTimezone + "<br />");

        Response.Write(sTimezone);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

setTimeZone

Sets the time zone for the report. This is calculated as the deviation, in minutes, from UTC for reporting. You can set the report time zone in the **Reports** tab of the Rescue Administration Center. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
timezone	The time zone calculated as the deviation, in minutes, from UTC for reporting. Required.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the specified time zone of the report succeeded.
NOTLOGGEDIN	Setting the specified time zone of the report failed because the current user is no longer logged in.
INVALIDPARAM_TIMEZONE	The time zone value does not specify a real time zone.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `setTimeZone` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setTimezone.aspx?
timezone=-240&authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
setTimezone.aspx">
    <input name="timezone" value="-240">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setTimezone>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

//set up array for SOAP request

$timezone = -240; //UTC -4 hours = -240 minutes (EST during Daylight
Savings)
$sAuthCode = "";
$setTimezoneParams = array(
'sTimezone' => $timezone,
'sAuthCode' => $sAuthCode
);

$setTimezoneResult = $soapclient->setTimezone($setTimezoneParams);
```

```

//print out the setTimeZone full response
print_r("<b>setTimeZone full response.</b><br />");
print_r($setTimeZoneResult);
echo "<br /><br />"; //formatting

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
    actionName.aspx?
    for each action called
        string sEmail = "some@email.com";
        string sPwd = "secretPassword";
        string sAuthCode = "4ahx...80u0";

        //set up the request
        HttpWebRequest oReq =
        (HttpWebRequest) System.Net.WebRequest.Create(sEndpoint
        + "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

        //create a cookie container to store the cookies for this session
        oReq.CookieContainer = new CookieContainer();

        //get the response
        HttpWebResponse oResp = (HttpWebResponse) oReq.GetResponse();

    string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
    Response.Write("Login result: " + sResp + "<br />"); //You can customize
    the response

    /*
    //debug cookies
    foreach (Cookie cook in oResp.Cookies)
    {
        Response.Write("Cookie:" + "<br />");
        Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
        >");
        Response.Write("Domain: " + cook.Domain + "<br />");
        Response.Write("Path: " + cook.Path + "<br />");
        Response.Write("Port: " + cook.Port + "<br />");
        Response.Write("Secure: " + cook.Secure + "<br />");

        Response.Write("When issued: " + cook.TimeStamp + "<br />");
        Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
        cook.Expired + "<br />");
        Response.Write("Don't save: " + cook.Discard + "<br />");
        Response.Write("Comment: " + cook.Comment + "<br />");
        Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
        Response.Write("Version: RFC " + cook.Version + "<br />");
    }
    */

```

```

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Timezone info
string sTimezone = "-240";
HttpRequest oReqTimezone =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
    + "setTimezone.aspx?Timezone=" + sTimezone + "&authcode=" + sAuthCode);
oReqTimezone.CookieContainer = sessioncookie;

HttpResponse oRespTimezone =
    (HttpResponse)oReqTimezone.GetResponse();
string sRespTimezone = new
    StreamReader(oRespTimezone.GetResponseStream()).ReadToEnd();
Response.Write("setTimezone result: " + sRespTimezone + "<br />"); //You
    can
    customize the response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setTimezone Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
            }
        }
    }
}

```

```

        short sTimezone = -240;

        WebserviceClients.loginRet oLogin = proxy.login(sEmail,
sPwd);

        Response.Write(oLogin + "<br />");

        WebserviceClients.setTimezoneRet oSetTimezone =
proxy.setTimezone
(sTimezone, sAuthCode);
        Response.Write(oSetTimezone);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

getReportType

Retrieves the type of reporting currently set. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Output

```
OK REPORTTYPE:LISTALL
```

REPORTTYPE: The listing type of your generated reports. Possible values are LISTALL and SUMMARY.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Retrieving a report succeeded.
NOTLOGGEDIN	Retrieving a report failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.

Sample Code

The following are examples for using the `getReportType` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/getReportType.aspx?
authcode=4ahx...80u0
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/
getReportType.aspx">
    <input name="authcode" value="4ahx...80u0">
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=getReportType>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/
api.asmx?wsdl");

//define parameters
$loginparams = array (
'sEmail' => 'some@email.com',
'sPassword' => 'secretPassword'
);

//login
$loginResult = $soapclient->login($loginparams);

//print the result
echo "<b>Login full response.</b><br />"; //formatting
print_r($loginResult);
echo "<br /><br />"; //formatting

$sAuthCode = "";

//set up array for getReportType request
$getReportTypeParams = array(
'sAuthCode' => $sAuthCode
);

$getReportTypeResult = $soapclient->getReportType($getReportTypeParams);

//print out the currently set ReportType format
print_r("<b>getReportType full response.</b><br />");
print_r($getReportTypeResult);
```

```
echo "<br /><br />"; //formatting

?>
```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
actionName.aspx?
for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "";

    //set up the request
    HttpWebRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());
```

```

}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//get the Output info
HttpRequest oReqGetReportType =
    (HttpRequest) System.Net.WebRequest.Create(sEndpoint
    + "getReportType.aspx?authcode=" + sAuthCode);
oReqGetReportType.CookieContainer = sessioncookie;

HttpWebResponse oRespGetReportType = (HttpWebResponse)
    oReqGetReportType.GetResponse();
string sRespGetReportType = new
    StreamReader(oRespGetReportType.GetResponseStream()).ReadToEnd();
Response.Write("getReportType result: " + sRespGetReportType + "<br />");
//You can customize the
response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API getReportType Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                REPORTTYPE eReportType;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
                sPwd);
            }
        }
    }
}

```

```

        Response.Write(oLogin + "<br />");

        WebServiceClients.getRepotyTypeRet oGetReportType =
proxy.getReportType(sAuthCode,
out eReportType);
        Response.Write(oGetReportType + "<br />");

        Response.Write(eReportType);
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
}

```

setReportType

Sets the report type for reporting. For information on reports, see *Generating Reports* in the [Rescue Administration Center User Guide](#).

Input Parameters

Element	Description
reporttype	The type of the report that can be LISTALL or SUMMARY. Required, default report type is LISTALL.
authcode	The secret authentication code that is used to authenticate the user without logging in to Rescue. Optional.

Return Values

Displayed Return Value	Description
ERROR	An unspecified error occurred, such as timeout.
OK	Setting the report output type succeeded.
NOTLOGGEDIN	Setting the report output type failed because the current user is no longer logged in.
INVALID_SECRETAUTHCODE	The secret authentication code for the user is invalid.
USER_DELETED_OR_DISABLED	The user is deleted or disabled.
NOTSUPPORTED	The given report type is not supported.

Sample Code

The following are examples for using the `setReportType` method that you can call in your environment.

HTTP GET

```
https://secure.logmeinrescue.com/API/setReportType.aspx?  
type=LISTALL&authcode=4ahx.
```

HTTP POST

```
<form method="post" action="https://secure.logmeinrescue.com/API/  
setReportType.aspx">  
    <input name="type" value="SUMMARY">  
    <input name="authcode" value="4ahx...80u0">  
</form>
```

SOAP

For sample SOAP 1.1 and SOAP 1.2 request and response messages, visit <https://secure.logmeinrescue.com/api/API.asmx?op=setReportType>.

PHP with SOAP

The example values shown must be replaced with actual values.

```
<?php  
$soapclient = new SoapClient("https://secure.logmeinrescue.com/api/  
api.asmx?wsdl");  
  
//define variables  
$sEmail = "some@email.com";  
$sPassword = "secretPassword";  
  
//define parameters  
$loginparams = array (  
'sEmail' => $sEmail,  
'sPassword' => $sPassword);  
  
//login  
$loginResult = $soapclient->login($loginparams);  
  
//print the result  
echo "<b>Login full response.</b><br />"; //formatting  
print_r($loginResult);  
echo "<br /><br />"; //formatting  
  
//switch to Summay list type reports  
$setType = array(  
'eReportType' => 'SUMMARY',  
'sAuthCode' => $sAuthCode  
);  
  
$Response = $soapclient->setReportType($setType);  
print_r("<b> setReportType Full response.</b> <br />");
```

```

print_r($Response);
echo "<br /><br />"; //formatting

?>

```

C# with HttpWebRequest

The example values shown must be replaced with actual values.

```

<%@ Page Language="C#" %>
<%@ Import Namespace="System.Net" %>
<%@ Import Namespace="System.IO" %>
<%@ Import Namespace="System" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<%
    string sEndpoint = "https://secure.logmeinrescue.com/API/"; //add
actionName.aspx?
for each action called
    string sEmail = "some@email.com";
    string sPwd = "secretPassword";
    string sAuthCode = "4ahx...80u0";

    //set up the request
    HttpWebRequest oReq =
(HttpWebRequest)System.Net.WebRequest.Create(sEndpoint
+ "login.aspx" + "?email=" + sEmail + "&pwd=" + sPwd);

    //create a cookie container to store the cookies for this session
oReq.CookieContainer = new CookieContainer();

    //get the response
    HttpWebResponse oResp = (HttpWebResponse)oReq.GetResponse();

string sResp = new StreamReader(oResp.GetResponseStream()).ReadToEnd();
Response.Write("Login result: " + sResp + "<br />"); //You can customize
the response

/*
//debug cookies
foreach (Cookie cook in oResp.Cookies)
{
Response.Write("Cookie:" + "<br />");
Response.Write("Name: " + cook.Name + " " + "Value: " + cook.Value + "<br /
>");
Response.Write("Domain: " + cook.Domain + "<br />");
Response.Write("Path: " + cook.Path + "<br />");
Response.Write("Port: " + cook.Port + "<br />");
Response.Write("Secure: " + cook.Secure + "<br />");

Response.Write("When issued: " + cook.TimeStamp + "<br />");
Response.Write("Expires: " + cook.Expires + " " + "Expired? " +
cook.Expired + "<br />");
Response.Write("Don't save: " + cook.Discard + "<br />");
Response.Write("Comment: " + cook.Comment + "<br />");
Response.Write("Uri for comments: " + cook.CommentUri + "<br />");
Response.Write("Version: RFC " + cook.Version + "<br />");

// Show the string representation of the cookie.
Response.Write("String: " + cook.ToString());

```

```

}*/

//add cookies to cookie container
CookieContainer sessioncookie = oReq.CookieContainer;

//set the Report type
string sReportType = "LISTALL";
HttpRequest oReqSetType =
    (HttpRequest)System.Net.WebRequest.Create(sEndpoint
    + "setReportType.aspx?type=" + sReportType + "&authcode=" + sAuthCode);
oReqSetType.CookieContainer = sessioncookie;

HttpWebResponse oRespSetType = (HttpWebResponse) oReqSetType.GetResponse();
string sRespSetType = new
    StreamReader(oRespSetType.GetResponseStream()).ReadToEnd();
Response.Write("setReportType result: " + sRespSetType + "<br />"); //You
    can customize the
    response

%>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Rescue API setReportType Test</title>
</head>
<body>

</body>
</html>

```

C# with SOAP

The example values shown must be replaced with actual values.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using apiSamples.WebServiceClients;

namespace apiSamples
{
    public partial class SOAP : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
            {
                WebServiceClients.API proxy = new WebServiceClients.API();
                proxy.CookieContainer = new CookieContainer();

                string sEmail = "some@email.com";
                string sPwd = "secretPassword";
                string sAuthCode = "4ahx...80u0";
                ReportType oReportType = ReportType.Summary;

                WebServiceClients.loginRet oLogin = proxy.login(sEmail,
                sPwd);
            }
        }
    }
}

```

```
        Response.Write(oLogin + "<br />");

        WebServiceClients.setReportRet oSetReportRet = proxy.
setReportType(oReportType,
sAuthCode);
        Response.Write(oSetReportRet + "<br />");
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}
}
```


Index

A

- account information [21](#)
 - get account [26](#)
 - get hierarchy [31, 36](#)
 - log in [21](#)
 - log out [24](#)
- account management [42](#)
 - assign channel [110](#)
 - create group [73](#)
 - create user [42](#)
 - create users [46](#)
 - delete node [107](#)
 - get group [76](#)
 - get group v2 [80, 83](#)
 - get user [51, 55, 57](#)
 - move node [103](#)
 - set group [86](#)
 - set group v2 [93](#)
 - set group v3 [98](#)
 - set user [60, 65](#)
 - setStatus_v7_1 [69](#)
 - technician available on channel [203](#)
 - unassign channel [114](#)
- API
 - authentication [7](#)
 - start pro meeting [13, 16, 18, 20](#)
- assign channel [110](#)
- authcode [7](#)

C

- channel form
 - customization [168](#)
- chat log
 - get [207](#)
- create group [73](#)
- create user [42](#)
- create users [46](#)
- CRM systems [117](#)
 - EU [137, 142](#)
 - generate email text [117](#)
 - login ticket [123](#)
 - request authentication code [126, 129](#)
 - request PIN code [132, 137, 142](#)
 - requestLensPinCode [148](#)
 - requestLensPinCode_v2 [153](#)
 - requestLensPinCode_v3 [158](#)
 - requestLensPinCode_v4 [164](#)

D

- date format
 - get [299](#)
 - set [301](#)
- delete node [107](#)

- delimiter character
 - get [303](#)
 - set [307](#)

G

- generate email text [117](#)
- generated reports [223, 224](#)
- get account [26](#)
- get group [76](#)
- get group v2 [80, 83](#)
- get hierarchy [31, 36](#)
- get report [215](#)
- get report_v2 [232](#)
- get session [171](#)
- get session v2 [177](#)
- get session v3 [180](#)
- get user [51, 55, 57](#)

L

- log in [21](#)
- log out [24](#)
- login [7](#)
- login ticket [123, 124](#)

M

- move node [103](#)

N

- nodeID [164](#)
- Nodeld [142](#)

O

- overview [6](#)

Q

- quick starts [7](#)

R

- report
 - columns [224](#)
 - generated [223](#)
 - get [215](#)
- report area
 - get [237](#)
 - get v2 [241](#)
 - get v3 [246](#)
 - get v5 [250](#)

- get v6 [254](#)
- get v7 [259](#)
- get v8 [263](#)
- set [268](#)
- set v2 [272](#)
- set v3 [276](#)
- set v5 [281](#)
- set v6 [285](#)
- set v7 [290](#)
- set v8 [294](#)
- report date
 - get [319](#)
 - get v2 [323](#)
 - set [327](#)
 - set v2 [331](#)
- report output
 - get [311](#)
 - set [315](#)
- report time
 - get [336](#)
 - set [340](#)
- report_v2
 - get [232](#)
- reporting [207](#)
- request authentication code [126](#), [129](#)
- request PIN code [132](#), [137](#), [142](#)
- requestLensPinCode [148](#)
- requestLensPINCode_v2 [153](#)
- requestLensPINCode_v3 [158](#)
- requestLensPINCode_v4 [164](#)

S

- session
 - get [171](#)
 - get v2 [177](#)
 - get v3 [180](#)
- session management [170](#)
 - cancel action [200](#)
 - close session [197](#)
 - hold session [189](#)
 - start session [186](#)
 - transfer session [193](#)
- session notes
 - get [211](#)
- set group [86](#)
- set group v2 [93](#)
- set group v3 [98](#)
- set user [60](#), [65](#)
- setUserStatus_v7_1 [69](#)

T

- technician available on channel channel [203](#)
- time zone
 - get [344](#)
 - set [348](#)

U

- unassign channel [114](#)