QTS HTTP API – Authentication v5.1.0

Version	Date	Description of Changes
5.1.0	2024/1/9	Initial Release

Table of Contents

1. System API 2

1.1 Authentication 2

1.2 Authentication with qtoken 5

1.3 Authentication with 2 step verification 7

1.3.1 1st verification 7

1.3.2 2nd verification 9

1.3.3 Send Emergency E-mail 12

1.3.4 Get Security Question 13

1.3.5 Get Security Question for Mobile App 14

1.3.6 Security Question Authentication 16

1.4 Login with sid 18

1.5 Logout 21

1. System API

1.1 Authentication

Description: Get sid for authentication by password.

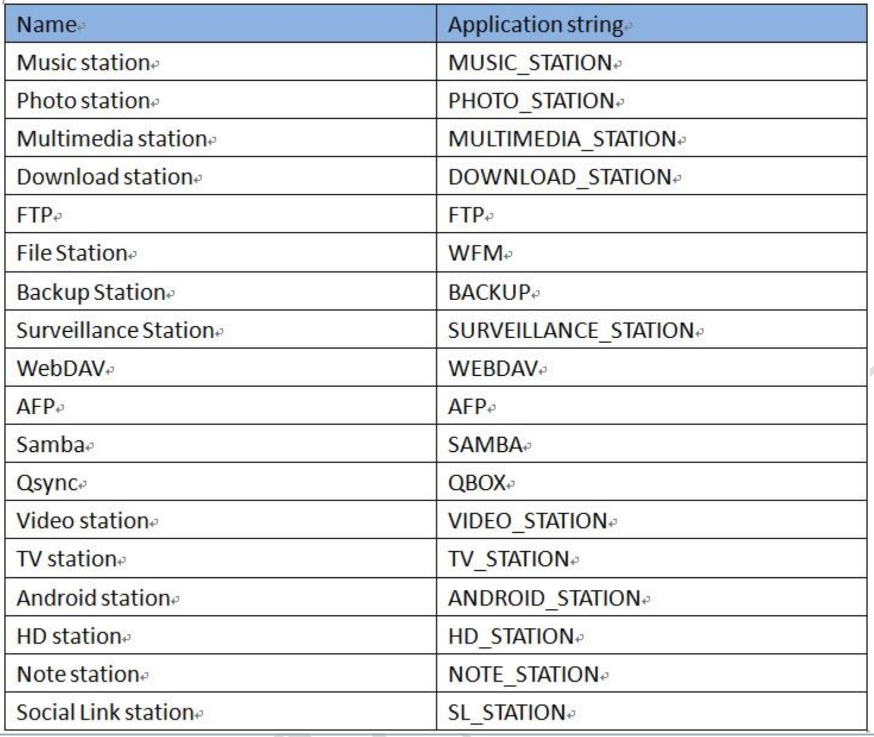
(use Remember me, get qtoken for authentication)

Command:

```
http://IP:8080/cgi-bin/authLogin.cgi?user=${username}&pwd=${encode_string}&remme=${remme}&service=${service}&remote_ip=${remote_ip}&device=${device}&force_to_check_2sv={force_to_check_2sv}&client_id=${client_id}&client_app=${client_app}&client_agent=${client_agent}&gen_client_id=${gen_client_id}&duration=${duration}
```

```
http://IP:8080/cgi-bin/authLogin.cgi?user=${username}&plain_pwd=${pwd_in_plain_text}&remme=${remme}&service=${service}&remote_ip=${remote_ip}&device=${device}&force_to_check_2sv={force_to_check_2sv}&client_id=${client_id}&client_app=${client_app}&client_agent=${client_agent}&gen_client_id=${gen_client_id}&duration=${duration}
```

Variable	Description
${username}	Login user name
${encode_string}	Password (Encode using Base64)
${pwd_in_plain_text}	Password in plain text
${remme}	Optional: 1:return qtoken 0:clean qtoken
${renew}	1:re-generate qtoken and return new qtoken 0:without return qtoken
${force_to_check_2sv}	Optional: 1: Force to check 2-step verification including query from 127.0.0.1 Others: no force
${service}	Username/ Password authentication by specifying service name. When the value>=100, sid will NOT be generated. 5: Qsync 101: Photo Station 102: Music Station 103: Video Station 100: Others 99: Force to check 2-step verification including query from 127.0.0.1
${remote_ip}	Optional; Only for localhost(127.0.0.1), specify the remote address
${device}	Optional; Specify the client' device name
${check_privilege}	Optional;

<table>
<tr><td></td><td>Specify the application string to check

<table>
<tr><th>Name</th><th>Application string</th></tr>
<tr><td>Music station</td><td>MUSIC_STATION</td></tr>
<tr><td>Photo station</td><td>PHOTO_STATION</td></tr>
<tr><td>Multimedia station</td><td>MULTIMEDIA_STATION</td></tr>
<tr><td>Download station</td><td>DOWNLOAD_STATION</td></tr>
<tr><td>FTP</td><td>FTP</td></tr>
<tr><td>File Station</td><td>WFM</td></tr>
<tr><td>Backup Station</td><td>BACKUP</td></tr>
<tr><td>Surveillance Station</td><td>SURVEILLANCE_STATION</td></tr>
<tr><td>WebDAV</td><td>WEBDAV</td></tr>
<tr><td>AFP</td><td>AFP</td></tr>
<tr><td>Samba</td><td>SAMBA</td></tr>
<tr><td>Qsync</td><td>QBOX</td></tr>
<tr><td>Video station</td><td>VIDEO_STATION</td></tr>
<tr><td>TV station</td><td>TV_STATION</td></tr>
<tr><td>Android station</td><td>ANDROID_STATION</td></tr>
<tr><td>HD station</td><td>HD_STATION</td></tr>
<tr><td>Note station</td><td>NOTE_STATION</td></tr>
<tr><td>Social Link station</td><td>SL_STATION</td></tr>
</table></td></tr>
<tr><td>${client_id}</td><td>uuid from client side generated</td></tr>
<tr><td>${client_app}</td><td>client application
Name of the QPKG, 、PC/MAC Utility, Mobile App, etc. The value is "Web Login" when login with the web login screen. e.g. Qmanager</td></tr>
<tr><td>${client_agent}</td><td>client agent
By referring to HTTP header, User-Agent: / (), the format is defined as {app_name}/{app_version} ({OS_name} {OS_version}, {device_name or model_name}) ex. Qmanager/2.18.1 (Android 11, Google Pixel 5)</td></tr>
<tr><td>${gen_client_id}</td><td>server side will response uuid and client need reuse the response uuid</td></tr>
<tr><td>${duration}</td><td>support custom token duration if parameter with client_id
(Unit: day)
default:90, minimum:1, -1:never expire</td></tr>
<tr><td>${vtoken}</td><td>vtoken for don't verify 2sv authentication</td></tr>
</table>

Note: encode_string = ezEncode(utf16to8('${real_password}'))

Example:
http://IP:8080/cgi-bin/authLogin.cgi?user=admin&pwd=YWRtaW4%3D&remme=1
(if real_password is "admin", it will be encoded as "YWRtaW4%3D")

Authentication by service
http://127.0.0.1:8080/cgi-bin/authLogin.cgi?plain_pwd=admin&user=admin&remote_ip=172.17.20.49&device=richardnb

Return value:

Example of successful authentication:

```
<?xml version="1.0" encoding="UTF-8" ?>
      <QDocRoot version="1.0">
      <qtoken><![CDATA[1e29b890910e8135f1692ed4030256fe]]></qtoken>  <= here is 
qtoken, if remme=0 hide
      <authPassed><![CDATA[1]]></authPassed>
      <authSid><![CDATA[ral08opo]]></authSid>           <= here is sid, 
generated when service>=100 was not set
      <isAdmin><![CDATA[1]]></isAdmin>
      </QDocRoot>
```

Example of failed authentication:

```
       <QDocRoot version="1.0">
        <qtoken>1e29b890910e8135f1692ed4030256fe</qtoken>
      <authPassed>0</authPassed>
      <errorValue>-1</errorValue>
      </QDocRoot>
```

Below are the tags inside process list.

Tag name	Description
authPassed	1: Success, 0: fail
en_qrlogin	0/1, user qrcode login enabled
force_qrlogin	0/1, user force qrcode login enabled
user_pw_expiry	0/1, user pw expiry
force_2sv	0/1, need force_2sv

Authorization by service

http://${ip}:8080/cgi-bin/authLogin.cgi?plain_pwd=admin&user=aix&remote_ip=172.17.20.49&service=104&device=aixchou&check_privilege=VIDEO_STATION

Return value:

`Example of permission deny:`

```
<?xml version="1.0" encoding="UTF-8" ?>
        <QDocRoot version="1.0">
        <doQuick><![CDATA[]]></doQuick>
        <is_booting><![CDATA[0]]></is_booting>
        <mediaReady><![CDATA[1]]></mediaReady>
        <SMBFW><![CDATA[0]]></SMBFW>
        <PermissionDeny><![CDATA[1]]></PermissionDeny>    <= permission deny
        <authPassed><![CDATA[0]]></authPassed>
        <errorValue><![CDATA[-1]]></errorValue>
        <username><![CDATA[aix]]></username>
        <ts><![CDATA[85022350]]></ts>
        <fwNotice><![CDATA[0]]></fwNotice>
        <title><![CDATA[]]></title>
        <content><![CDATA[]]></content>
        <psType><![CDATA[1]]></psType>
        <showVersion><![CDATA[0]]></showVersion>
        <show_link><![CDATA[1]]></show_link>
        </QDocRoot>
```

Below are the tags inside process list.

Tag name	Description
authPassed	1: Success, 0: fail
errorValue	0: success -1: Fail -2: not administrator -3 administrator password expired -4 password expired

1.2 Authentication with qtoken

Description: Get sid for authentication by qtoken.

Command:
http://IP:8080/cgi-bin/authLogin.cgi?user=${username}&qtoken=${qtoken}&remme=${remme}&client_id=${client_id}&client_app=${client_app}&client_agent=${client_agent}&gen_client_id=${gen_client_id}&duration=${duration}

Variable	Description
${username}	Login user name
${qtoken}	qtoken
${remme}	Optional: 1:when login by qtoken without return qtoken tag 0:clean qtoken
${client_id}	uuid from client side generated
${client_app}	client application Name of the QPKG,、PC/MAC Utility, Mobile App, etc. The value is "Web Login" when login with the web login screen. e.g. Qmanager
${client_agent}	client agent By referring to HTTP header, User-Agent: <product>/<product-version> (<comment>), the format is defined as {app_name}/{app_version} ({OS_name} {OS_version}, {device_name or model_name}) ex. Qmanager/2.18.1 (Android 11, Google Pixel 5)

${gen_client_id}	server side will response uuid and client need reuse the response uuid
${duration}	support custom token duration if parameter with client_id (Unit: day) default:90, minimum:1, -1:never expire
${vtoken}	token for don't verify 2sv authentication

Example:
http://IP:8080/cgi-bin/authLogin.cgi?user=admin&qtoken=1e29b890910e8135f1692ed4030256fe&remme=1

Return value:

Example of successful authentication:

```
<?xml version="1.0" encoding="UTF-8" ?>
      <QDocRoot version="1.0">
      <authPassed><![CDATA[1]]></authPassed>
      <authSid><![CDATA[ral08opo]]></authSid>           <= here is sid
      <isAdmin><![CDATA[1]]></isAdmin>
      </QDocRoot>
```

Example of failed authentication:

```
       <QDocRoot version="1.0">
      <authPassed>0</authPassed>
      <errorValue>-1</errorValue>
      </QDocRoot>
```

Below are the tags inside process list.

Tag name	Description
user_pw_expiry	0/1, user pw expiry
force_2sv	0/1, need force_2sv
authPassed	1: Success, 0: fail
user_enable	1: enable, 0: disable
user_account_expiry	1: expired, 0: not expired
user_expiry_year	which year to expired
user_expiry_month	which month to expired (1-12)
user_expiry_day	which day to expired (1-31)
errorValue	0: success -1: Fail -2: not administrator -3 administrator password expired -4 password expired

1.3 Authentication with 2 step verification

Description:
If 2-step verification enabled, get sid for 1st authentication by password and 2nd authentication by security code(6 digits) / emergency security code(8 digits) / emergency answer.

In 4.2.0, only support QTS system / File Station login.

HTTP Request from 127.0.0.1 can get sid without 2-step verification.

1.3.1 1st verification

Description: 1st verification

Command:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=${encode_string}&r=0.802557202605028&remme=${remme}&serviceKey=1&user=${username}&client_id=${client_id}&client_app=${client_app}&client_agent=${client_agent}&gen_client_id=${gen_client_id}&duration=${duration}

Variable	Description
${username}	Login user name
${encode_string}	Password
${remme}	Optional: 1:return qtoken 0:clean qtoken
${client_id}	uuid from client side generated
${client_app}	client application Name of the QPKG, PC/MAC Utility, Mobile App, etc. The value is "Web Login" when login with the web login screen. e.g. Qmanager
${client_agent}	client agent By referring to HTTP header, User-Agent: <product>/<product-version> (<comment>), the format is defined as {app_name}/{app_version} ({OS_name} {OS_version}, {device_name or model_name}) ex. Qmanager/2.18.1 (Android 11, Google Pixel 5)
${gen_client_id}	server side will response uuid and client need reuse the response uuid
${duration}	support custom token duration if parameter with client_id (Unit: day) default:90, minimum:1, -1:never expire
${vtoken}	token for don't verify 2sv authentication

Note: encode_string = ezEncode(utf16to8('${real_password}'))

Example:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=YWRtaW4%3D&r=0.022977258475666645&remme=1&serviceKey=1&user=admin
(if real_password is "admin", it will be encoded as "YWRtaW4%3D")

Return value:
Example of successful for 1st authentication:

```
<QDocRoot version="1.0">
	<doQuick></doQuick>
	<is_booting>0</is_booting>
	<mediaReady>1</mediaReady>
	<SMBFW>0</SMBFW>
	<authPassed>0</authPassed>
	<need_2sv>1</need_2sv>
	<lost_phone>1</lost_phone>
	<emergency_try_count>0</emergency_try_count>
	<emergency_try_limit>5</emergency_try_limit>
	<username>admin</username>
	<groupname>administrators</groupname>
	<ts>88323841</ts>
	<fwNotice>0</fwNotice>
	<title></title>
	<content></content>
	<psType>0</psType>
	<showVersion>0</showVersion>
	<show_link>1</show_link>
</QDocRoot>
```

Below are the tags inside process list.

Tag name	Type	Description
authPassed	int	authentication result 0:fail 1:success
errorValue	int	0: success -1: Fail -2: not administrator -3 administrator password expired -4 password expired
need_2sv	int	whether user need 2-step verification 1:need 2-step verification if need_2sv not return, please refer general Authentication

need_2_step_verification	int	user need new 2-step verification 1:need 2-step verification New API refer New 2sv API
lost_phone	int	click "Verify another way" will lead to the following Optional: 1:send emergency e-mail that contains emergency security code(8 digits) 2:emergency question handler
emergency_try_count	int	if "lost_phone" is 1, it means the number of send emergency mail if "lost_phone" is 2, it means the number of answer incorrectly
emergency_try_limit	int	if "lost_phone" is 1, it means the maximum number of send emergency mail if "lost_phone" is 2, it means the maximum number of answer incorrectly so far, we define "emergency_try_limit" is 5

1.3.2 2nd verification

Description: verify security code(6 digits) or emergency security code(8 digits)

Command:

```
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=${encode_string}&r=0.3938051044582034&remme=${remme}&security_code=${security_code}&serviceKey=1&user=${username}&client_id=${client_id}&client_app=${client_app}&client_agent=${client_agent}&gen_client_id=${gen_client_id}&duration=${duration}
```

Variable	Description
${username}	Login user name
${encode_string}	Password (Please refer to attached 「get_sid.js」 to get ezEncode function)
${remme}	Optional: 1:return qtoken 0:clean qtoken
${dont_verify_2sv_again }	0 for disable, 1 for enable. you must provide this parameter
${security_code}	security code(6 digits) • while emergency handler is "Send emergency e-mail", it can be emergency security code(8 digits)
${client_id}	uuid from client side generated
${client_app}	client application Name of the QPKG, 、PC/MAC Utility, Mobile App, etc. The value is "Web Login" when login with the web login screen. e.g. Qmanager

${client_agent}	client agent By referring to HTTP header, User-Agent: <product>/<product-version> (<comment>), the format is defined as {app_name}/{app_version} ({OS_name} {OS_version}, {device_name or model_name}) ex. Qmanager/2.18.1 (Android 11, Google Pixel 5)
${gen_client_id}	server side will response uuid and client need reuse the response uuid
${duration}	support custom token duration if parameter with client_id (Unit: day) default:90, minimum:1, -1:never expire

Note: encode_string = ezEncode(utf16to8('${real_password}'))

Example:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=YWRtaW4%3D&r=0.3938051044582034&security_code=215238&serviceKey=1&user=admin
(if real_password is "admin", it will be encoded as "YWRtaW4%3D")

Return value:

Example of successful for 2nd authentication:

```
<QDocRoot version="1.0">
      <doQuick></doQuick>
      <is_booting>0</is_booting>
      <mediaReady>1</mediaReady>
      <SMBFW>0</SMBFW>
      <authPassed>1</authPassed>
      <authSid>mxz01een</authSid>
      <need_2sv>1</need_2sv>
      <lost_phone>1</lost_phone>
      <emergency_try_count>0</emergency_try_count>
      <emergency_try_limit>5</emergency_try_limit>
      <isAdmin>1</isAdmin>
      <username>admin</username>
      <groupname>administrators</groupname>
      <ts>88323841</ts>
      <fwNotice>0</fwNotice>
      <title></title>
      <content></content>
      <psType>0</psType>
      <showVersion>0</showVersion>
      <show_link>1</show_link>
</QDocRoot>
```

Example of failed for 2nd authentication:

```
<QDocRoot version="1.0">
```

```
        <doQuick></doQuick>
        <is_booting>0</is_booting>
        <mediaReady>1</mediaReady>
        <SMBFW>0</SMBFW>
        <authPassed>0</authPassed>
        <need_2sv>1</need_2sv>
        <lost_phone>1</lost_phone>
        <emergency_try_count>0</emergency_try_count>
        <emergency_try_limit>5</emergency_try_limit>
        <date_time>
                <timezone>(GMT+08:00) Taipei</timezone>
                <timestamp>1432803710</timestamp>
                <date_format_index>1</date_format_index>
                <time_format>24</time_format>
        </date_time>
        <username>admin</username>
        <groupname>administrators</groupname>
        <ts>97469902</ts>
        <fwNotice>0</fwNotice>
        <SUID>6801bd1901459a79a9a39eb6c24da8fb</SUID>
        <title></title>
        <content></content>
        <psType>1</psType>
        <showVersion>0</showVersion>
        <show_link>1</show_link>
</QDocRoot>
```

Most parameters can refer 1st verification

Below are the other tags inside process list.

Tag name	Type	Description
authPassed	int	1: Success, 0: Fail
need_2sv	int	1: need 2-step verification 0: no need 2-step verification
emergency_try_count	int	emergency try count, it should not exceed emergency_try_limit
emergency_try_limit	int	emergency try limit. It will reply authentication fail if exceed the limit.
timezone	string	Time Zone, ex."(GMT+08:00) Taipei"
timestamp	int	seconds since Jan 01 1970
date_format_index	int	date format index 1 : year/month/day, 2 : year.month.day, 3 : year-month-day,

		4 : month/day/year, 5 : month.day.year, 6 : month-day-year, 7 : day/month/year, 8 : day.month.year, 9 : day-month-year
time_format	int	time format : 24 / 12
vtoken	string	token for don't verify 2sv

1.3.3 Send Emergency E-mail

Description: Send emergency e-mail request (while 1st verification or 2nd verification return value "lost_phone" is 1)

Command:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=${encode_string}&r=0.3938051044582034&remme=${remme}&send_mail=1&serviceKey=1&user=${username}

Variable	Description
${username}	Login user name
${encode_string}	Password (Please refer to attached 「get_sid.js」 to get ezEncode function)
${remme}	Optional: 1:return qtoken 0:clean qtoken

Note: encode_string = ezEncode(utf16to8('${real_password}'))

Example:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=YWRtaW4%3D&r=0.3938051044582034&send_mail=1&serviceKey=1&user=admin
(if real_password is "admin", it will be encoded as "YWRtaW4%3D")

Return value:

```
<QDocRoot version="1.0">
      <doQuick></doQuick>
      <is_booting>0</is_booting>
      <mediaReady>1</mediaReady>
      <SMBFW>0</SMBFW>
      <send_result>1</send_result>
      <emergency_try_count>3</emergency_try_count>
      <emergency_try_limit>5</emergency_try_limit>
      <username>admin</username>
      <groupname>administrators</groupname>
```

```
       <ts>88323841</ts>
       <fwNotice>0</fwNotice>
       <title></title>
       <content></content>
       <psType>0</psType>
       <showVersion>0</showVersion>
       <show_link>1</show_link>
</QDocRoot>
```

Below are the tags inside process list.

Tag name	Type	Description
send_result	int	send mail result 1:success 0:fail -1:the email notification service is not enabled
emergency_try_count	int	it means the number of send emergency mail
emergency_try_limit	int	it means the maximum number of send emergency mail so far, we define "emergency_try_limit" is 5

1.3.4 Get Security Question

Description: Get security question (while 1st verification or 2nd verification return value "lost_phone" is 2)

Command:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=${encode_string}&r=0.3938051044582034&remme=${remme}&get_question=1&serviceKey=1&user=${username}

Variable	Description
${username}	Login user name
${encode_string}	Password (Please refer to attached 「get_sid.js」 to get ezEncode function)
${remme}	Optional: 1:return qtoken 0:clean qtoken

Note: encode_string = ezEncode(utf16to8('${real_password}'))

Example:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?get_question=1&pwd=YWRtaW4%3D&r=0.3938051044582034&serviceKey=1&user=admin
(if real_password is "admin", it will be encoded as "YWRtaW4%3D")

Return value:

```
<QDocRoot version="1.0">
	<doQuick></doQuick>
	<is_booting>0</is_booting>
	<mediaReady>1</mediaReady>
	<SMBFW>0</SMBFW>
	<security_question_no>4</security_question_no>
	<security_question_text>how are you?</security_question_text>
	<username>admin</username>
	<groupname>administrators</groupname>
	<ts>88323841</ts>
	<fwNotice>0</fwNotice>
	<title></title>
	<content></content>
	<psType>0</psType>
	<showVersion>0</showVersion>
	<show_link>1</show_link>
</QDocRoot>
```

Below are the tags inside process list.

Tag name	Type	Description
security_question_no	int	question no 1: frontend will show "What is your pet's name?" 2: frontend will show "What is your favorite sport?" 3: frontend will show "What is your favorite color?" 4: Custom question
security_question_text	string	Custom question (only for "security_question_no" is 4)

1.3.5 Get Security Question for Mobile App

Description: Get security question for Mobile App (while 1st verification or 2nd verification return value "lost_phone" is 2)

Command:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=${encode_string}&r=0.3938051044582034&remme=${remme}&get_question=1&serviceKey=1&user=${username}&q_lang=${q_lang}

Variable	Description
${username}	Login user name
${encode_string}	Password (Please refer to attached 「get_sid.js」 to get ezEncode function)

${remme}	Optional: 1:return qtoken 0:clean qtoken
${q_lang}	Languages for seucruty question {CZE, DAN, DUT, ENG, ESM, FIN, FRE, GER, GRK, HUN, ITA, JPN, KOR, NOR, POL, POR, ROM, RUS, SCH, SPA, SWE, TCH, THA, TUR}

Note: encode_string = ezEncode(utf16to8('${real_password}'))

Example:
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?get_question=1&pwd=YWRtaW4%3D&r=0.3938051044582034&serviceKey=1&user=admin&q_lang=ENG
(if real_password is "admin", it will be encoded as "YWRtaW4%3D")

Return value:

```
<QDocRoot version="1.0">
      <doQuick></doQuick>
      <is_booting>0</is_booting>
      <mediaReady>1</mediaReady>
      <SMBFW>0</SMBFW>
      <security_question_no>4</security_question_no>
      <system_question_text>how are you?</system_question_text>
      <security_question_text>how are you?</security_question_text>
      <username>admin</username>
      <groupname>administrators</groupname>
      <ts>88323841</ts>
      <fwNotice>0</fwNotice>
      <title></title>
      <content></content>
      <psType>0</psType>
      <showVersion>0</showVersion>
      <show_link>1</show_link>
</QDocRoot>
```

Below are the tags inside process list.

Tag name	Type	Description
security_question_no	int	question no 1: frontend will show "What is your pet's name?" 2: frontend will show "What is your favorite sport?" 3: frontend will show "What is your favorite color?" 4: Custom question
system_question_text	string	security question text (for mobile app)
security_question_text	string	Custom question (only for "security_question_no" is 4)

1.3.6 Security Question Authentication

Description:
verify security answer (while 1st verification or 2nd verification return value "lost_phone" is 2)

Command:
```
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=${encode_string}&r=0.3938051044582034&remme=${remme}&security_answer=${security_answer}&serviceKey=1&user=${username}&client_id=${client_id}&client_app=${client_app}&client_agent=${client_agent}&gen_client_id=${gen_client_id}&duration=${duration}
```

Variable	Description
${username}	Login user name
${encode_string}	Password (Please refer to attached 「get_sid.js」 to get ezEncode function)
${remme}	Optional: 1:return qtoken 0:clean qtoken
${dont_verify_2sv_again}	Optional: 0 for disable, 1 for enable. you must provide this parameter
${security_answer}	string
${client_id}	uuid from client side generated
${client_app}	client application Name of the QPKG,、PC/MAC Utility, Mobile App, etc. The value is "Web Login" when login with the web login screen. e.g. Qmanager
${client_agent}	client agent By referring to HTTP header, User-Agent: <product>/<product-version> (<comment>), the format is defined as {app_name}/{app_version} ({OS_name} {OS_version}, {device_name or model_name}) ex. Qmanager/2.18.1 (Android 11, Google Pixel 5)
${gen_client_id}	server side will response uuid and client need reuse the response uuid
${duration}	support custom token duration if parameter with client_id (Unit: day) default:90, minimum:1, -1:never expire

Note: encode_string = ezEncode(utf16to8('${real_password}'))

Example:
```
http://172.17.20.20:8080/cgi-bin/authLogin.cgi?pwd=YWRtaW4%3D&r=0.4000929836850201&security_answer=fine&serviceKey=1&user=admin
```
(if real_password is "admin", it will be encoded as "YWRtaW4%3D")

Return value:

Example of successful for security question authentication:

```
<QDocRoot version="1.0">
        <doQuick></doQuick>
        <is_booting>0</is_booting>
        <mediaReady>1</mediaReady>
        <SMBFW>0</SMBFW>
        <authPassed>1</authPassed>
        <authSid>m9x71gxw</authSid>
        <emergency_try_count>1</emergency_try_count>
        <emergency_try_limit>5</emergency_try_limit>
        <isAdmin>1</isAdmin>
        <username>admin</username>
        <groupname>administrators</groupname>
        <ts>88323841</ts>
        <fwNotice>0</fwNotice>
        <title></title>
        <content></content>
        <psType>0</psType>
        <showVersion>0</showVersion>
        <show_link>1</show_link>
</QDocRoot>
```

Example of failed for security question authentication:

```
<QDocRoot version="1.0">
        <doQuick></doQuick>
        <is_booting>0</is_booting>
        <mediaReady>1</mediaReady>
        <SMBFW>0</SMBFW>
        <authPassed>0</authPassed>
        <emergency_try_count>1</emergency_try_count>
        <emergency_try_limit>5</emergency_try_limit>
        <username>admin</username>
        <groupname>administrators</groupname>
        <ts>88323841</ts>
        <fwNotice>0</fwNotice>
        <title></title>
        <content></content>
        <psType>0</psType>
        <showVersion>0</showVersion>
        <show_link>1</show_link>
</QDocRoot>
```

These parameters can be referred to 1st verification

1.4 Login with sid

Description: Login with sid

Command:
http://IP:8080/cgi-bin/authLogin.cgi?sid=${sid}

Example:
http://IP:8080/cgi-bin/authLogin.cgi?sid=pr4i5et6

Return value:

```
<?xml version="1.0" encoding="UTF-8"?>
<QDocRoot version="1.0">
   <doQuick />
   <is_booting>0</is_booting>
   <mediaReady>1</mediaReady>
   <SMBFW>0</SMBFW>
   <authPassed>1</authPassed>
   <isAdmin>1</isAdmin>
   <user>admin</user>
   <username>admin</username>
   <groupname>administrators</groupname>
   <userid>0</userid>
   <userType>local</userType>
   <force_change_pw>1</force_change_pw>
   <pw_expiry_date>2017/01/29</pw_expiry_date>
   <model>
      <modelName>TS-670</modelName>
      <internalModelName>TS-670</internalModelName>
      <platform>TS-NASX86</platform>
      <platform_ex>X86_SANDYBRIDGE</platform_ex>
      <customModelName />
      <displayModelName>TS-670 Pro</displayModelName>
      <sas_model>0</sas_model>
      <storage_v2>1</storage_v2>
      <encryptfsSupported>1</encryptfsSupported>
   </model>
   <firmware>
      <version>4.4.0</version>
      <number>0614</number>
      <build>20160614</build>
      <patch>0</patch>
      <buildTime>2016/06/14</buildTime>
   </firmware>
   <rfs_bits>64</rfs_bits>
   <sp>ALPHA</sp>
   <specVersion>1.0</specVersion>
   <hostname>SAMSON-TS670</hostname>
   <DemoSiteSuppurt>no</DemoSiteSuppurt>
   <customLogo>
      <customFrontLogo />
      <customLoginLogo />
   </customLogo>
   <gqMaster>-1</gqMaster>
   <HTTPHost>172.17.20.124</HTTPHost>
   <webAccessPort>8080</webAccessPort>
   <QWebPort>80</QWebPort>
   <webFSEnabled>1</webFSEnabled>
   <QMultimediaEnabled>0</QMultimediaEnabled>
   <MSV2Supported>0</MSV2Supported>
   <MSV2WebEnabled>1</MSV2WebEnabled>
   <MSV2URL>/MSV2/</MSV2URL>
   <QDownloadEnabled>0</QDownloadEnabled>
   <DSV2Supported>0</DSV2Supported>
   <DSV3Supported>1</DSV3Supported>
   <DSV2URL>/downloadstation/?ssid=</DSV2URL>
   <QWebEnabled>1</QWebEnabled>
   <QWebSSLEnabled>1</QWebSSLEnabled>
   <QWebSSLPort>8081</QWebSSLPort>
   <NVREnabled>0</NVREnabled>
   <NVRURL>/cgi-bin/camera_view.cgi</NVRURL>
   <NVRVER>1</NVRVER>
   <WFM2>1</WFM2>
   <wfmPortEnabled>0</wfmPortEnabled>
   <wfmPort>8080</wfmPort>
   <wfmSSLEnabled>0</wfmSSLEnabled>
   <wfmSSLPort>443</wfmSSLPort>
   <wfmURL>/filestation/</wfmURL>
   <QMusicsEnabled>1</QMusicsEnabled>
   <QMusicsURL>/musicstation/</QMusicsURL>
   <QVideosEnabled>1</QVideosEnabled>
   <QVideosURL>/videostation/</QVideosURL>
   <QPhotosEnabled>1</QPhotosEnabled>
   <QPhotosURL>/photo/</QPhotosURL>
   <stunnelEnabled>1</stunnelEnabled>
   <stunnelPort>443</stunnelPort>
   <forceSSL>0</forceSSL>
   <HDAROOT_ALMOST_FULL>0</HDAROOT_ALMOST_FULL>
   <passwdConstraints>
      <passwdConstraint01>0</passwdConstraint01>
      <passwdConstraint02>0</passwdConstraint02>
      <passwdConstraint03>0</passwdConstraint03>
      <passwdConstraint04>0</passwdConstraint04>
   </passwdConstraints>
   <quickStart>1</quickStart>
   <connet_info>
      <connet_ip>172.17.20.90</connet_ip>
   </connet_info>
   <ts>29225615</ts>
   <fwNotice>0</fwNotice>
   <SUID>6801bd1901459a79a9a39eb6c24da8fb</SUID>
   <title />
   <content />
   <psType>1</psType>
   <standard_massage />
   <standard_color>#ffffff</standard_color>
   <standard_size>12px</standard_size>
   <standard_bg_style>fill</standard_bg_style>
   <showVersion>0</showVersion>
   <show_link>1</show_link>
   <role_delegation>
      <role>
         <name>
            <![CDATA[user_group_management]]>
         </name>
         <desc>
            <![CDATA[User and Group Management Desc]]>
         </desc>
         <id>
            <![CDATA[5]]>
         </id>
      </role>
      <role>
         <name>
            <![CDATA[shared_folder_management]]>
         </name>
         <desc>
            <![CDATA[Shared Folder Management Desc]]>
         </desc>
         <id>
            <![CDATA[6]]>
         </id>
      </role>
      <role>
         <name>
```

```
            <![CDATA[backup_management]]>
         </name>
         <desc>
            <![CDATA[Backup Management Management Desc]]>
         </desc>
         <id>
            <![CDATA[7]]>
         </id>
      </role>
   </role_delegation>
</QDocRoot>
```

Response format:

Tag name	Description
authPassed	1: Success, 0: Fail
connet_ip	source ip
role_delegation id	System Management = 1, Application Management = 2, Access Management = 3, System Monitoring = 4, User and Group Management = 5, Shared Folder Management = 6, Backup Management

1.5 Logout

Description: logout with sid

Command:
http://IP:8080/cgi-bin/authLogout.cgi?sid=${sid}

Example:
http://IP:8080/cgi-bin/authLogout.cgi?sid=pr4i5et6

Return value:

```
<QDocRoot version="1.0">
    <authPassed><![CDATA[0]]></authPassed>
</QDocRoot>
```

Parameter:

Tag name	Description
logout	1: do logout
sid	sid to logout

del_user_session	1: delete sid and qtoken belong to user 2: delete sid only 3: delete sid and qtoken except qnap_authenticator
del_client	verify qtoken and remove qtoken from specified user and client_id
qtoken	qtoken to delete (use with del_client)
client_id	qtoken of client_id to delete (use with del_client)
user	username of client_id to delete (use with del_client)

Response format:

Tag name	Description
authPassed	1: auth pass