

Ashly AquaControl API Documentation

All Ashly AquaControl products serve up an interactive API document. You can access the API server in Chrome, Edge, or Safari web browser on your PC or mobile device. Make sure the AquaControl product is on and connected to the same network as your PC or mobile device. Enter the following address in your web browser's address bar:

[AquaControl-Product's-IP-Address]:8000/documentation

For example:

192.168.111.247:8000/documentation

or

http://[product-hostname].local:8000/documentation

The product-hostname format is "model_(*mac address*)", with an underscore character between the model and mac address. For example:

http://AQM1208_0014AA036024.local:8000/documentation

http://AQM408_0014AA03605C.local:8000/documentation

http://MXA1502_0014AAF00031.local:8000/documentation

You will see a page load that looks similar to this:

The screenshot shows a Swagger API documentation page for an 'event' API. The page is titled 'API documentation 1.0-beta' and includes a search bar with '/swagger.json' and an 'Explore' button. The base URL is 'aqm1208_0019ffffff24.local:8000/v1.0-beta'. A dropdown menu shows 'Schemes' with 'HTTP' selected. The 'event' section is expanded, showing a list of endpoints:

Method	Endpoint	Description
GET	/event	get event records
DELETE	/event/{id}	get action type records
GET	/event/actions/parameter/types	get all action parameter type records
GET	/event/actions/types	get action type records
GET	/event/getCancellationStatus	get status of event cancellation
GET	/event/operation/parameter/type	get all operation parameter type records
GET	/event/operation/type	get all operation type records
POST	/event/resumeSchedule	used to run a test event
POST	/event/scheduled	Create a new scheduled event

In order to interact with the APIs, you must first use the session/login API to establish an admin login connection to the unit from this Swagger document page. To do this, scroll down to the **session** section. Click on the down arrow at the far-right side of session/login to expand it, and click on "Try it out":

The screenshot shows a REST client interface with the following details:

- Method: **POST**
- Endpoint: `/session/login`
- Parameters: A **Try it out** button is highlighted with a red box.
- Request Body: A JSON object with the following structure:

```
{  "username": "string",  "password": "string",  "keepLoggedIn": true}
```
- Parameter content type: `application/json`
- Response content type: `application/json`
- Response Code: `default`
- Response Description: `Successful`
- Response Example Value: `"string"`

Enter your login username and password for the product. If this is a new product then use the factory default credentials and click Execute. Factory default credentials are (case sensitive):

`"username": "admin",`

`"password": "secret",`

← → ↻ Not secure aqm1208_0019ffffff24.local:8000/documentation#/

session

POST /session/authorizedlogout

GET /session/checkCurrentLogin

POST /session/login

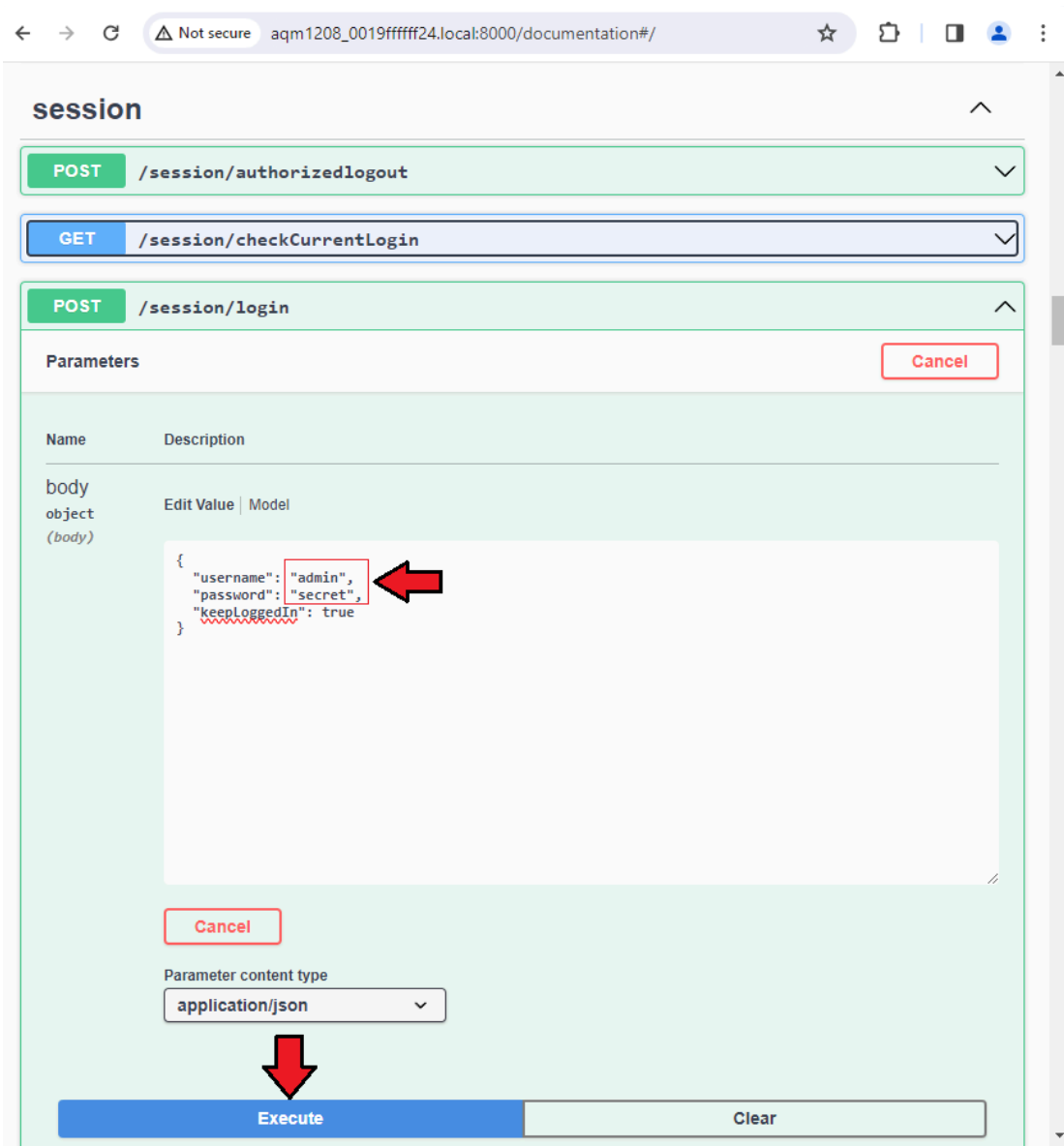
Parameters Cancel

Name	Description
body	
object	Edit Value Model
(body)	<pre>{ "username": "admin", "password": "secret", "keepLoggedIn": true }</pre>

Cancel

Parameter content type
application/json

Execute Clear



Below it you will see the Post message that went out and the response from the AQM1208:

The screenshot shows a web browser window displaying a REST client interface. The address bar shows the URL: `http://aqm1208_0019ffffff24.local:8000/documentation#/session/postSessi...`. The interface is titled "Responses" and shows a successful POST request and response.

Response content type: application/json

Curl:

```
curl -X 'POST' \
  'http://aqm1208_0019ffffff24.local:8000/v1.0-beta/session/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "username": "admin",
  "password": "secret",
  "keepLoggedIn": true
}'
```

Request URL: `http://aqm1208_0019ffffff24.local:8000/v1.0-beta/session/login`

Server response:

Code	Details
200	<p>Response body:</p> <pre>{ "success": true, "data": [{ "loggedIn": true, "enabledPermissions": ["View Accounts", "Edit Accounts", "Remote Logout", "View Device Misc.", "Edit Device Misc.", "View All DSP settings", "Event Log View", "Event Log Clear", "Event Scheduler View", "Event Scheduler Edit", "Front Panels Control View", "Front Panels Control Edit", "Network Settings View", "Network Settings Edit", "Paging Ducking View", "Paging Ducking Edit", "Preset View", "Preset Edit", "Preset Recall", "DCA Remote", "Mixer 1 Remote", "Mixer 2 Remote",] }] }</pre> <p>Response headers:</p> <pre>cache-control: no-cache connection: keep-alive content-encoding: gzip content-type: application/json; charset=utf-8 date: Mon, 19 Feb 2024 16:23:16 GMT keep-alive: timeout=5 transfer-encoding: chunked vary: accept-encoding</pre>

NOTE: Use GET message APIs to read data from the unit and see valid parameters / id values that can be used for POST messages.

Now that you have successfully established a login session into the Swagger API, you can try out other API messages on the page. The login session created a security cookie in your browser that is required for most AquaControl API message types. If you are using an old control system application that does not support security cookies then you will not be able to access most of the AquaControl APIs. For such cases, there is a special set of APIs under the section called "simplecontrol" that do not require security cookies. However, in order for these "simplecontrol" APIs to work with the product, you must create a user account in the product with the username **SimpleControl**, (which is case-sensitive and requires capital S and C). It can be of any role type and does not require any permissions. When the product receives a "simplecontrol" API message type, it will only accept it if a SimpleControl user account is present. You can create the SimpleControl user account by logging into the product's AquaControl web server via an admin account and navigate to the Settings → Security page, or you can use the Swagger API documentation to create a user by using the "security" POST security/users API.

The simplecontrol APIs include preset recall, power on/off, channel level control, channel muting, source selection (mixer source mute), mix levels, and DVCA group level control & muting:

simplecontrol

Show/Hide | List Operations | Expand Operations

GET	/simplecontrol/preset/progress	check if preset recall is in progress
POST	/simplecontrol/preset/recall/{id}	Copy info from preset to working settings
GET	/simplecontrol/system/frontPanel/info/power	get system power state
POST	/simplecontrol/system/frontPanel/info/power	Turn on or off system power
GET	/simplecontrol/workingsettings/dsp/block/gain/level	get all gain block level parameters
POST	/simplecontrol/workingsettings/dsp/block/gain/level/{id}	set specific gain block level parameter
GET	/simplecontrol/workingsettings/dsp/chain/mute/	get all channel mute status
POST	/simplecontrol/workingsettings/dsp/chain/mute/{id}	set the dsp channel mute
GET	/simplecontrol/workingsettings/dsp/mixer/sourceMutesAndLevels	get all of the dsp mixer config source mutes and levels
POST	/simplecontrol/workingsettings/dsp/mixer/sourcelevel/{id}	set specific DSP mixer config input channel level
POST	/simplecontrol/workingsettings/dsp/mixer/sourcemute/{id}	Set specific DSP mixer input channel mute
POST	/simplecontrol/workingsettings/virtualDVCA/parameters/level/{id}	Modify the virtual DCA channel parameter
POST	/simplecontrol/workingsettings/virtualDVCA/parameters/mute/{id}	Modify the virtual DCA channel parameter
GET	/simplecontrol/workingsettings/virtualDVCA/parameters/mutesAndLevels	get the virtual DCA parameter settings

The SimpleControl account can be created in AquaControl Settings → Security:

The screenshot shows the AquaControl Settings interface. The 'Settings' menu item is circled in red and labeled '1'. The 'Security' tab is selected and circled in red and labeled '2'. A 'New Profile' button is circled in blue and labeled '3'. The user profile table is as follows:

User	Role	Last login
admin	Admin	12/18/2024 10:09 AM
SimpleControl	Operator	Never
RMAdmin	Admin	Never
Prova	Operator	04/22/2024 04:43 AM
Patio	Operator	Never
Mikeup	Operator	11/21/2024 10:04 AM
Manager	Operator	03/15/2022 10:18 AM

Or, on the Swagger server API documentation page, use the security/users POST API to create the account with the message body containing username SimpleControl, role Operator, any password of your choice, and make the permission fields blank, as shown below. All items are case sensitive:

security

GET /security/users Get a list of all the user records

POST /security/users add new user record

add new user record

Parameters

Cancel

Name Description

body object (body)

Edit Value | Model

```
{
  "username": "SimpleControl",
  "roleId": "Operator",
  "password": "secret",
  "permissions": [
  ],
  "remotePermissions": [
  ]
}
```

Cancel

Parameter content type

application/json



Execute Clear