

Atomation S2S APIs

Contents

Atomation S2S APIs.....	1
Top level view:	1
Request details:.....	2
Response details:	3
Testing the API endpoints:	4
Authentication:	6

Atomation REST API documentation and testing is available online on <https://atapi.atomation.net/s2s-api-docs>

It is built as follows:

Top level view:

Atomation 3rd party Server-to-Server API v1.0
[Base URL: test.atomation.net/ap1/v1/s2s/v1_0]

Introduction

Set of APIs for 3rd parties, allowing them to automatically extract Atomation data in JSON format, and use that information for data analysis or integrate it in their BI applications.

sections (points to 'data' and 'authentication' sections)

REST endpoints (points to the endpoint list items)

Authentication settings (points to the 'Authorize' button)

data APIs for extracting data generated by Atomation devices and sensors

- POST **/sensors_readings** Returns sensors samples
- POST **/sensors_events** Returns sensors events

authentication Authentication and token management endpoints

- POST **/auth/login** s2s login
- POST **/auth/token** s2s token generation

Here we see the endpoints that are available, without the details of each endpoint.

Request details:

After clicking and opening an endpoint, details with an example of the request body values is presented:

data APIs for extracting data generated by Atomation devices and sensors

POST `/sensors_readings` Returns sensors samples

This API returns samples read by sensors, filtered by:

- i. devices, defined by an array of 0 or more "mac" addresses
- ii. Start and end date

Parameters

Name	Description
request_body (body)	Example Value Model <pre>{ "filters": { "start_date": "10-04-2019T20:56:47.000Z", "end_date": "20-03-2020T11:57:58.000Z", "mac": ["C8:7B:EB:AD:29:13", "DC:27:1C:75:B2:17"] }, "limit": { "page": 1, "page_size": 100 } }</pre> Parameter content type application/json

[Try it out](#)

Request body Example Tab

Request body Model Tab

Option to test the API call

But after clicking the **request body "Model" tab** there, the request's JSON structure will be presented, with details about request body fields, their types, possible values and descriptions:

POST `/sensors_readings` Returns sensors samples

This API returns samples read by sensors, filtered by:

- i. devices, defined by an array of 0 or more "mac" addresses
- ii. Start and end date

Parameters

Name	Description
request_body (body)	Example Value Model <pre>Data_Req_Body { filters limit page page_size }</pre> Filters > {...} Limit > { description: page: number page_size: number } samples are returned in pages: the entire set of elements that match the filter are divided into pages, with a constant size for each page. so for example, if we have a total on N samples that match the filter, then sample in position K, will be available on page round(K/page_size)+1, and the total number of pages will be Round(N/page_size)+1 Along with page_size, returns all filtered elements that fit inside this page number. Numbering starts at 1. a. The element count used when creating the pagination b. the maximum number of elements that will be returned with each page request

[Try it out](#)

Model tab with option to drill down the body structure

A÷O M A÷I O N

Response details:

The various endpoint response options are specified as follows:

Response
JSON Model
description

Successful response

Example JSON response

[illegible]

Testing the API endpoints:

It is possible to perform actual API calls to the server and check the responses:

After clicking the “Try it out” button of the desired endpoint, the request “Example Values” tab will be become editable (as well as other request parameters in case such are available):

The screenshot shows the API testing interface for the `POST /sensors_readings` endpoint. The interface includes a description of the endpoint, a parameters table, and an example request body. Annotations highlight key features:

- Editable request body:** Points to the JSON example value in the "Example Value" tab.
- Grey lock means that Authorization is not set yet:** Points to a lock icon in the top right corner.
- Cancel API test:** Points to a "Cancel" button below the example value.
- Perform the endpoint call with given request:** Points to the "Execute" button at the bottom.

Parameters Table:

Name	Description
request_body <i>(body)</i>	

Example Value (JSON):

```
{  "filters": {    "start_date": "10-04-2019T20:56:47.000Z",    "end_date": "20-03-2020T11:57:58.000Z",    "mac": [      "C9:7B:EB:AD:29:13",      "DC:27:1C:75:B2:17"    ]  },  "limit": {    "page": 1,    "page_size": 100  }}
```

Parameter content type: application/json

After pressing the “Execute” button, the following information will be displayed:

The actual HTTP request that is sent to the server, as a curl command

Full endpoint URL

[illegible]

Actual Server response body

Actual Server response Header

The curl command that is generated on execution, can be used to generate the API calls on the actual client application, or on other HTTP test tools (such as Postman).

It contains the exact HTTP request that is sent to the server.

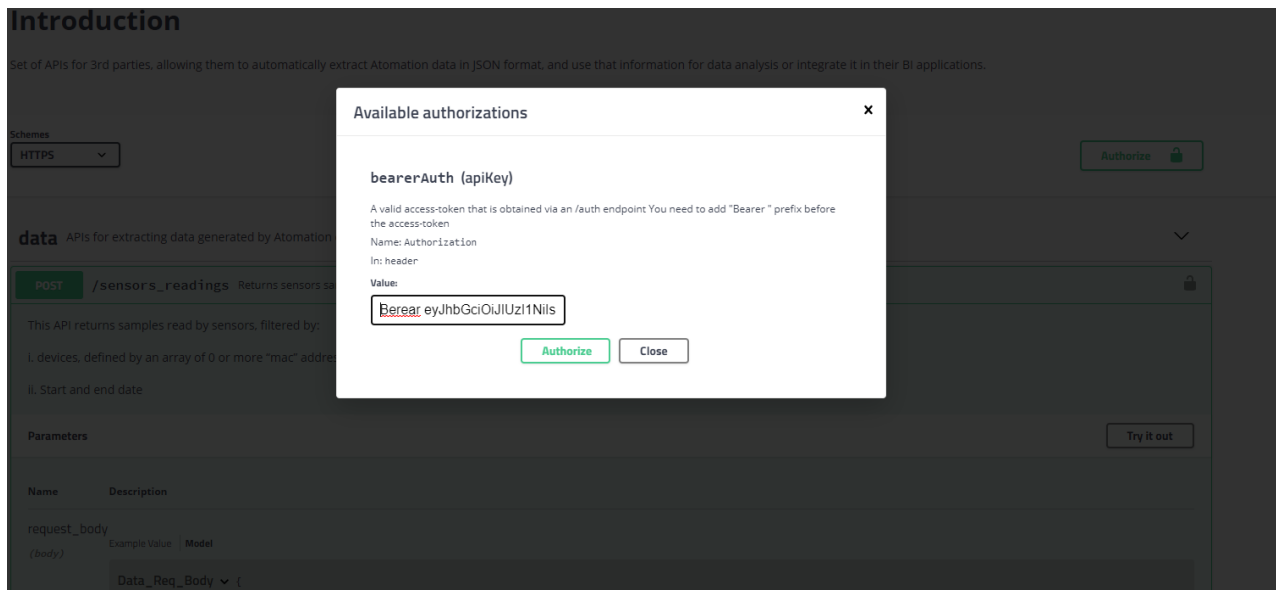
Pressing the “Clear” button will hide all the information that is related to the endpoint execution (request + response).

Authentication:

Endpoints that require authentication have the lock icon next to them.
The icon is grey when no authentication settings were given.

In order to provide authentication information:

1. Execute auth/login endpoint call (or auth/token endpoint in case you already have an expired token)
2. Provide the response' token to the API test mechanism by clicking the "Authorize" button (on top of the page)
3. It will open the following dialog:



4. In the edit box type "Bearer " and paste the token
5. press "Authorize" and then "Close"

At this point the lock icon next to the endpoint should turn from grey to black, and executing the endpoint will now add the token information in the HTTP header as requested.
This information will be visible of course in the curl command output.