

# REST API for Developers

- SAYMON API
- What can I do with the SAYMON API?
- What API commands are allowed?
- What about the access rights?
- Authentication
  - Basic Access Authorization
  - Session ID (Cookie)
  - API Key (API Token)

## SAYMON API

SAYMON API belongs to the REpresentational State Transfer (REST) category. This allows you to perform "RESTful" operations like read, modify, add or delete data from your cloud account or on-site installation.

New methods and their description are constantly added here: <https://saas.saymon.info/apidoc/>

## What can I do with the SAYMON API?

Every operation you can perform in SAYMON [web interface](#) can be implemented with API, e.g.:

- get list of [objects](#);
- get [current state of objects and their state history](#);
- [write data](#) to objects without [agents](#);
- modify [properties](#) of objects;
- work with [incidents](#);
- create [classes](#) with [probes](#) addition;
- and many more.

## What API commands are allowed?

SAYMON API has been implemented as JSON over HTTP and allows the following REST Commands:

Command	Purpose
GET	Fetch information about existing items
POST	Create new items
PUT	Update and modify existing items
PATCH	Update and modify existing items
DELETE	Remove existing items

## What about the access rights?

Your ability to access data or perform actions depends on the permissions available for your SAYMON user account.

## Authentication

Before you start you may wish to learn more about authentication. There are several options described below.

### Basic Access Authorization

Pretend your login is *login* and the password is *password*. You can use basic auth to get info about your account like this:

```
curl -v -u login:password -H "Content-Type: application/json" -X GET
https://your.server.name/node/api/users/current
```

## Session ID (Cookie)

Pretend your login is *login* and the password is *password*. You can use cookies to get info about your account like this:

### curl

```
curl -v --cookie-jar cookies.txt -H "Content-Type: application/json" -X
POST --data '{"login": "login", "password": "password"}' https://your.
server.name/node/api/users/session
curl -v --cookie cookies.txt -H "Content-Type: application/json" -X GET
https://your.server.name/node/api/users/current
```

### ajax

```
$.ajax
({
  type: "POST",
  url: "https://your.server.name/node/api/users/session",
  dataType: 'json',
  async: true,
  data:
  {
    "login": "login",
    "password": "password"
  },
  success: function (res){
    console.log('Thanks for your login!');
    console.log('Session ID:'+res);
  },
  error: function () {
    console.log('Login failed!');
  }
});
```

## API Key (API Token)

Pretend your login is *login* and the password is *password*:

- login with these credentials to the web-interface of the system;
- click on your login on the top right corner;
- go to Configuration page.

Your API token will be available in the Authorization Link field -<https://your.server.name/?auth-token=52cc80e1-e2d8-4cff-a242-dcfb538b585>. Use it in each request like this:

```
curl -v -H "Content-Type: application/json" -X GET https://your.server.name  
/node/api/users/current?api-token=52cc80e1-e2d8-4cff-a242-dfcfb538b585
```