

# Using the JavaScript REST client

**OpencgaClient** is the Javascript client library for OpenCGA RESTful Web Services, all the web services are accessible through this client, and it offers a quick way to query OpenCGA projects through web interface. In the same way than in [Using RESTful Web Services URL](#) tutorial, we will focus on **those end points more interesting for HGVA users**. In order to make it easy to follow we will use the same examples used in [Using RESTful Web Services URL](#).

## Configuring and initializing the client

There are two javascript files that need to be imported - [opencga-client-config.js](#) and [opencga-client.js](#).

OpenCGA Client expects an OpenCGA Client Configuration to initialise and create the client. The configuration object is of the following structure

### OpenCGA Client Configuration

```
this.opencgaClientConfig = new OpenCGAClientConfig("bioinfo.hpc.cam.ac.uk/hgva", "v1", false, "hgva");
```

### Arguments of OpencgaClientConfig

- Host - The host name is given here. In our case, It is the HGVA host.
- Version - Version number
- UseCookies - Cookies are used to store user ID and session ID. They are not needed in HGVA as The data is open to public. Hence It is set to **false**.
- CookiePrefix - Prefix is used in case we are using cookies. This can be set anything as the user wishes.

OpenCGA Client can be created with the above opencgaClientConfig.

### OpenCGA Client

```
this.opencgaClient = new OpenCGAClient(this.opencgaClientConfig);
```

## Getting information about genomic variants

The **then** and **catch** clauses are used as opencgaClient returns responses as Javascript **Promises**.

```
this.opencgaClient.studies().getVariants(id, params, options)
    .then(function(response) {
        Correct response is captured here
    })
    .catch(function(response) {
        Error response is captured here
    });
```

## Getting information about projects

Method to call for getting information about projects

```
this.opencgaClient.projects().info(id, params, options)
    .then(function(response) {
        Correct response is captured here
    })
    .catch(function(response) {
        Error response is captured here
    });
```

Method to call for getting studies belonging to the project

### Table of Contents:

- [Configuring and initializing the client](#)
  - [Arguments of OpencgaClientConfig](#)
- [Getting information about genomic variants](#)
- [Getting information about projects](#)
- [Getting information about studies](#)

```
this.opencgaClient.projects().getStudies(id, params, options)
    .then(function(response) {
        Correct response is captured here
    })
    .catch(function(response) {
        Error response is captured here
    });
```

## Getting information about studies

Method to call for getting all metadata for all available studies

```
this.opencgaClient.studies().search(params, options)
    .then(function(response) {
        Correct response is captured here
    })
    .catch(function(response) {
        Error response is captured here
    });
```

Method to call for getting summary data for a study whose ID must be passed as a parameter

```
this.opencgaClient.studies().summary(id, params, options)
    .then(function(response) {
        Correct response is captured here
    })
    .catch(function(response) {
        Error response is captured here
    });
```

Method to call for getting all metadata for a study

```
this.opencgaClient.studies().info(id, params, options)
    .then(function(response) {
        Correct response is captured here
    })
    .catch(function(response) {
        Error response is captured here
    });
```

Method to call for getting all samples metadata for a study

```
this.opencgaClient.studies().getSamples(id, params, options)
    .then(function(response) {
        Correct response is captured here
    })
    .catch(function(response) {
        Error response is captured here
    });
```