API documentation for Sequencescape

The sequencescape API is designed to manage projects/studies and manipulating lab ware. It's designed to enable PSD and third parties to build applications for displaying information and manipulating data stored in sequencescape.

Quick start

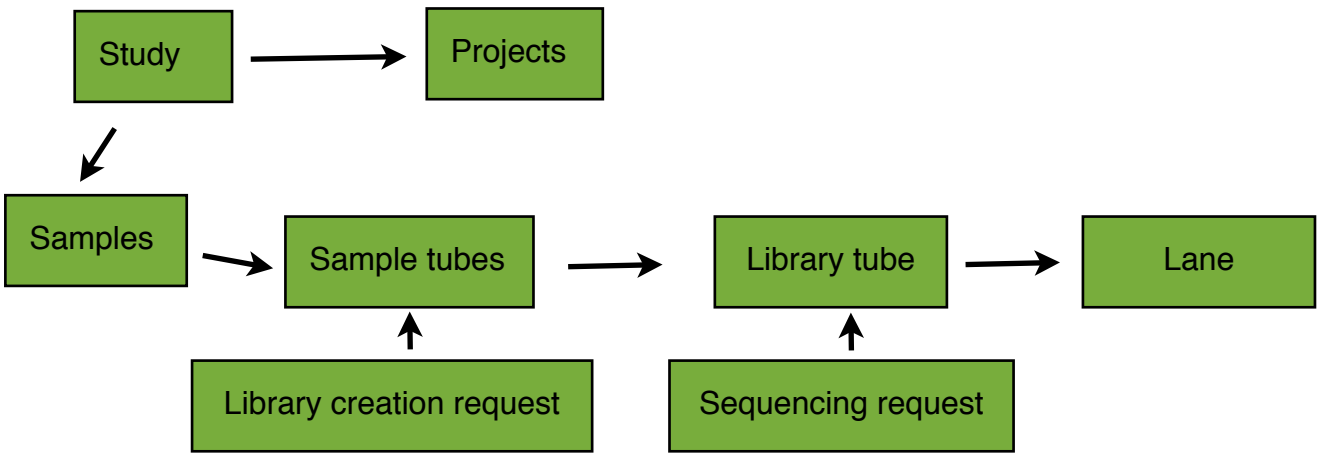
Below are a few simple steps to how you can create requests in Sequencescape using the API. The prerequisites you need is a HTTP client and a Sanger SSO key.

We recommend using curl or RESTClient for Firefox. In the API you can currently use either the internal IDs of projects or the UUID. We recommend that you move the UUID.

1. Log into Sanger SSO.
2. Get your WTSISignOn cookie (in this example AbC..d3)
3. Using curl execute the following command (for project 384 and asset 59408)

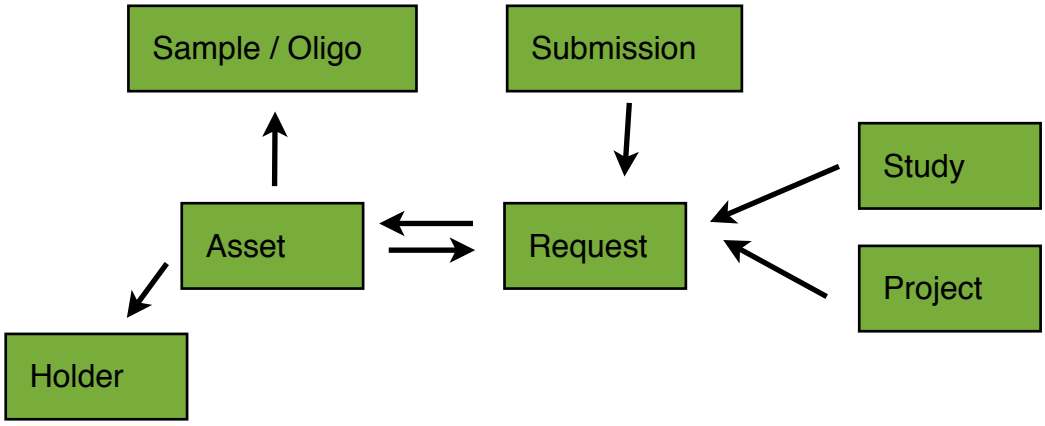
```
curl -H 'Content-Type:application/json' -H 'Accept:application/json' \
-H 'Cookie: WTSISignOn=AbC..d3' \
-d ' {
      "project" : { id : 384},
      "asset" : { "id" :  59408},
      "request_type" : {
              "single_ended_sequencing": {
                      "read_length": 108
              }
      },
      "comments": "This is a Single Ended Sequencing  request created via
WAPITI"} '  http://wapiti.com/0_0/requests
```

Navigating the API



Data model

Sequencescape includes as number of types of data.



Object type	Description
Study	A study describes the scientific details of what needs to be done. A study is related to a ENA/EGA/ArrayExpress study.
Project	A project describes the financial details.
Asset	An asset is a general term for anything which is a lab asset
Request	Requests are based on assets and creates assets. An example of this would be to aliquote a well from from one well to another well.
Sample	A sample is the source starting point. This will eventually be merged with the StockTube it lives in.
SampleTube	A sample tube is an asset that contains a DNA sample.
1D/2D Barcode	When referring to a barcode we talk about the data.
Holder	A holder is the container of assets. For example Rack or Plate.
Material	A material can be a sample or oligo

Data model

Wells and tubes are related via requests. Every request has a source and a target asset. This means that any asset has parents and children. Apart from this relationship there is also a holder relationship. A holder is an asset that contains other assets.

Technical overview

This API use a standard RESTful HTTP routing scheme and JSON to describe data.

There is only one level of nested resource and there are no nested objects. Instead, a link to the nested object is provided in response body. We use UUID (Universal Unique Identifier) instead of IDs that are specific to the underlying applications. This will provide a unique identifier for a given object.

Accessing an object via its specific system ID (e.g. an id in Sequencescape) will be available initially but we strongly recommend to only use and store the UUID.

Routing

the basic routes are :

```
GET :version/:resource/:id => returns the description of the
specified resource.
POST :version/:resource => creates one (or many) items of the
resource
PUT :version/:resource/:id => updates the specified resource.
DELETE :version/:resource/:id => removes the specified resource.
```

Nested resources routes are :

```
GET :version/:resource/:id/nested_resources => returns the list of
all the nested resources of a particular type belonging to the
specified resource.
GET :version/:resource/:id/nested_resources => returns the
description of a singular nested resource
DELETE :version/:resource/:id/nested_resources/:nested_resource_id
=> detaches the nested resource from its resource
```

```
e.g. 0_1/projects/123/requests
```

/!\ Warning some route might no be implemented in all version. See the version section to see what is exactly implemented in a specific version.

VERSION

version should match the following regular expression :

```
/\d+(_\d+)*/ example 1_0, 12_3, 1_2_3_4_5
```

ID vs UUID

id in the route could be either an standard ID or an UUID. The API will recognize UUIDs from normal IDs.

id :

```
123
3
```

uuid

```
3953ae20-2c2e-11df-84b5-00254ba8dcb4
```

Authentication

The authentication system use SSO. The sso token should be providen for each request as the WTISignOn value of a cookie. For more information on the Sanger SSO see XXXX.

Error

This API use standard HTTP error code but will add some suberror code to specify errors more accurately. The text description is also available.

HTTP response codes

Return values for queries using the HTTP GET call

Return value	Explanation
200 OK	The query was successful. The body of the HTTP response contains the query result.
404 Not Found	The object requested was not found.
413 Request Entity To Large	The request will return a response that is too large for the client.
500 Internal Server Error	An internal error occurred while processing the request.

Return values for queries using the HTTP PUT or POST call

Return value	Explanation
200 OK	The query was successful. The body of the HTTP response contains the query result.
201 Created	The PUT created a new object on the system. The PUT or POST was successful.

Return value	Explanation
400 Bad request	The contents of the PUT or POST were invalid and the data was rejected.
409 Conflict	The PUT or POST attempted to modify an existing object that cannot be modified. The PUT or POST was unsuccessful.
500 Internal Server Error	An internal error occurred while processing the request.

POST requests must have content-type multipart/form-data, and the object(s) being uploaded must be added as input type “file”.

API reference

All URLs should be prefixed with a version. Example /samples becomes /0_5/samples.

Creating a study

You need to create a study do sequencing

URL	/studies
HTTP Method	POST

Input Format	`{"study":`
	`{`
	`"name":"Viral discovery pilot",`
	`"sac_sponsor":"Julian Parkhill",`
	`"study_description":"Identification of novel viruses in encephalopathy of unknown cause"`
	`}`
	`}`

Result	Creation of a study

Creating a project

URL	/projects
HTTP Method	POST

Input Format	`{"project":`
	`{`
	`“name”: “Test project”,`
	`"project_cost_code”: “000/S000”`
	`}`
	`}`

Result	Creation of a project

Creating a samples

URL	/samples
HTTP Method	POST

Input Format	<pre>{
	  "sample": {
	  "gc_content": "neutral",
	  "organism": "Mycobacterium tuberculosis",
	  "name": "TB N0126",
	  "public_name": "",
	  "common_name": "Mycobacterium tuberculosis",
	  "strain": "TB N0126",
	  "scientific_rationale": "",
	  "ebi_accession_number": "ERS003762",
	  "taxon_id": "1773",
	  "concentration": "",
	  "description": "my description",
	  "ena_visibility": "hold"
	  }
	}</pre>

Result	Creation of a sample

Requests/Submission

If you want to request sequencing or genotyping via the API. Then you need to create a submission.

There are a few different types of requests you can do.

Request library creation and paired end sequencing

URL	/submissions
HTTP Method	POST
Input Format	{ “project_id”: “233”, “study_id”: “233”, “sample_tubes”: [“23”, “2332”, “2323”], “type”: “paired_end_sequencing”, “comments”: “Comments”, “number_of_lanes”: “3” }
Result	Creates a submission

Request multiplexed library creation and paired end sequencing

URL	/submissions
HTTP Method	POST
Input Format	{ “project_id”: “233”, “study_id”: “233”, “sample_tubes”: [“23”, “2332”, “2323”], “type”: “multiplexed_paired_end_sequencing”, “comments”: “Comments”, “number_of_lanes”: “2” }
Result	Creates a submission

Retrieving submission types

URL	/submissions/types
HTTP Method	GET
Result	Lists types of available submissions

Find batch by flow cell barcode

URL	/batches/find_by_flowcell_barcode/R234324
HTTP Method	GET
Result	Retrieves the batch for a given flow cell barcode

Creating Events

URL	/events
HTTP Method	POST

Input Format	{
	  "event": {
	  "message": "",
	  "eventful_id": "234",
	  "eventful_type": "Request",
	  "family": "pass",
	  "identifier": "",
	  "location": ""
	  }
	}

Result	A pass event for Request and calls the event mechanisms.
Description	Events are used to drive updates in Sequencescape. A pass event to a Request will cause the Request to become passed.