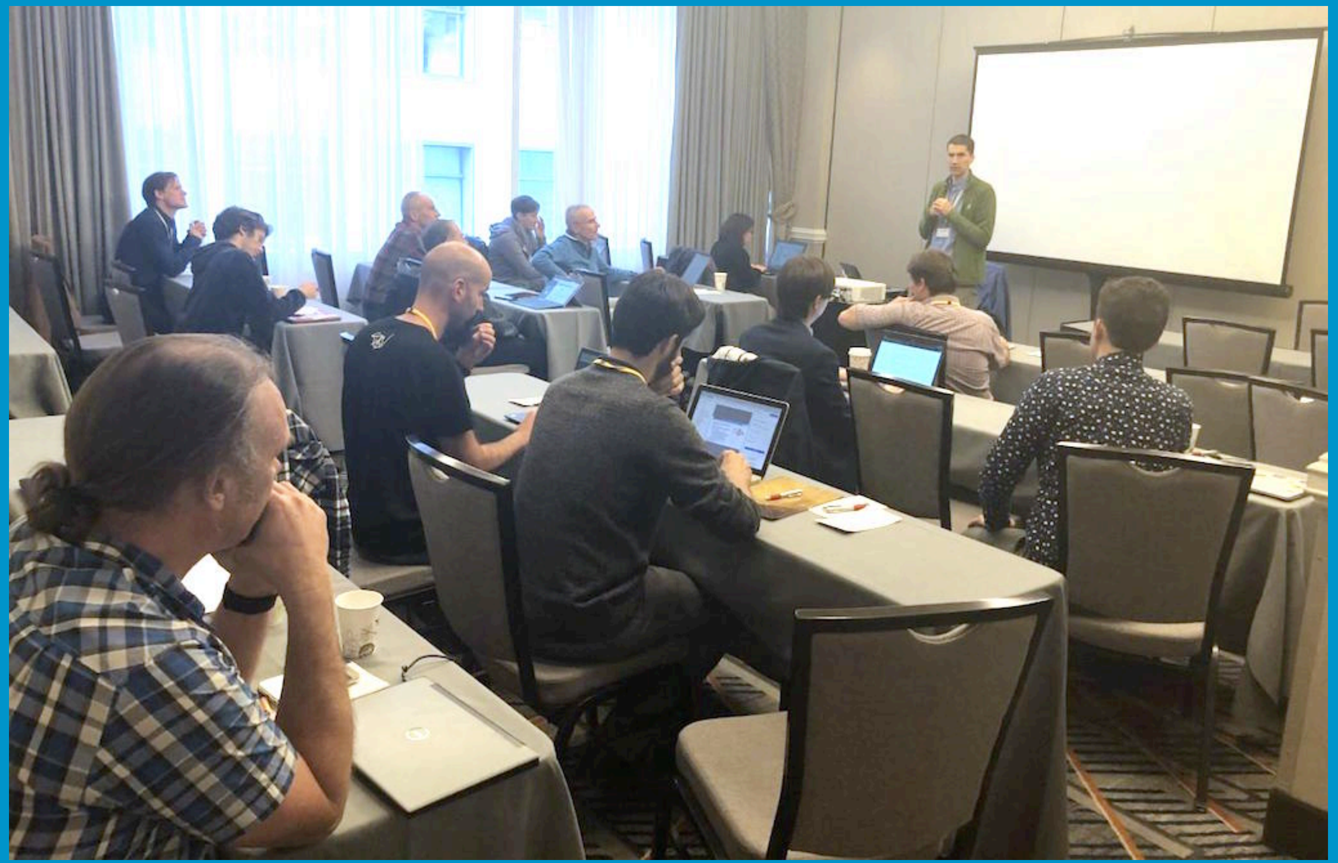
panosc

photon and neutron
open science cloud

EUROPEAN
SPALLATION
SOURCE

WP3 Search API Hackathon @ ESS

Gareth Murphy



SciCat @garethcmurphy

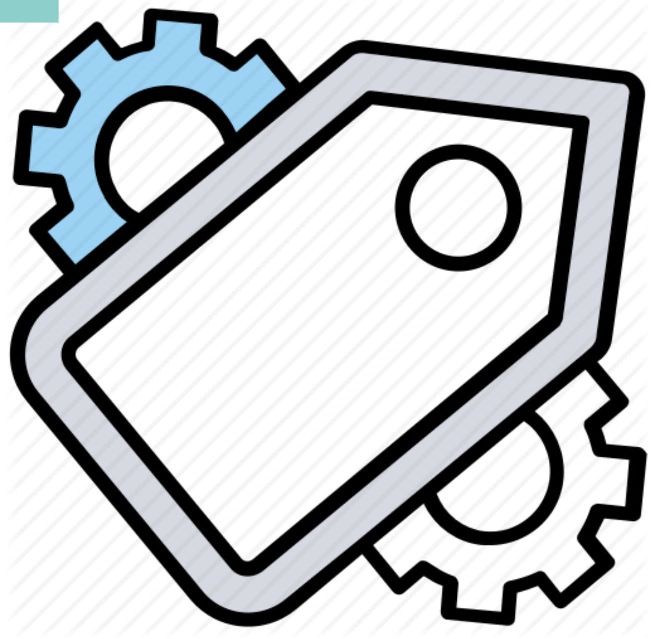
@essneutron

SWAP TEAM

EC/DC

Metadata in EOSC

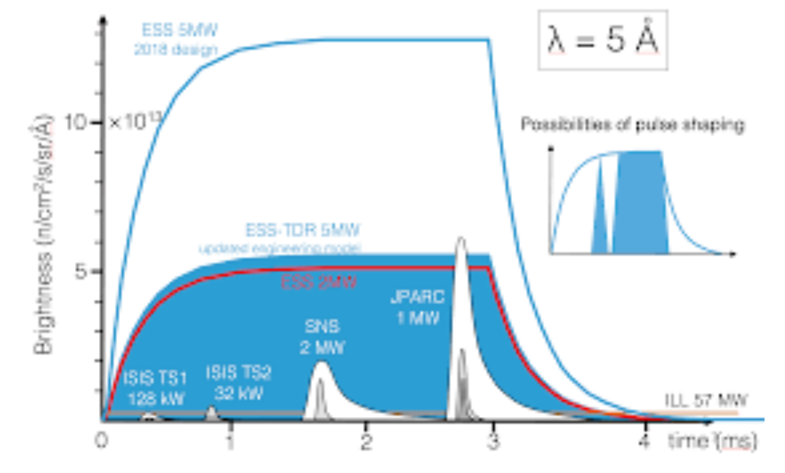
“one of the more important steps will be the possibility to compare the data gathered in a current experiment, to those collected in previous ones, to acquire complementary information for a proper interpretation of the data. This can only be achieved by **proper metadata and properly labelled data.** *It would then be possible to use the same settings in simulation, and directly compare the data collected, with those gathered before“ - Aljosa Hafner, PaNOSC/CERIC*

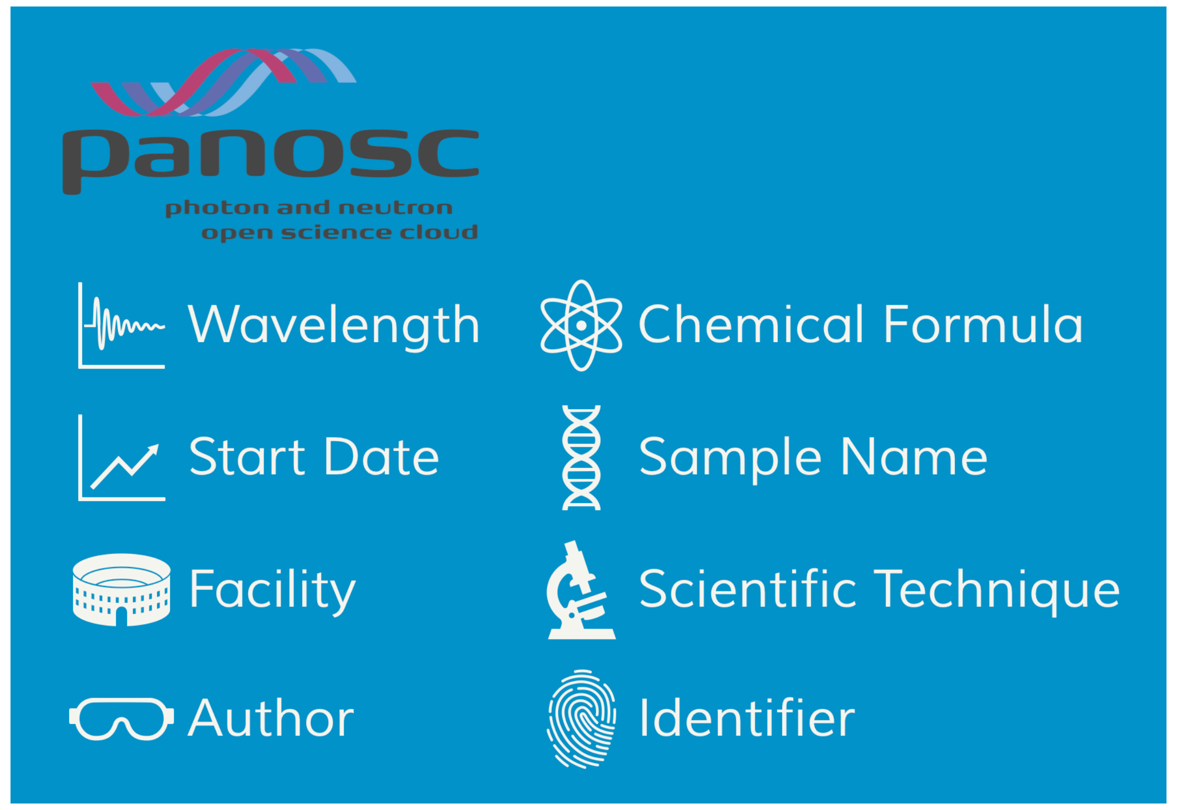


https://www.panosc.eu/news/interview-with-panosc-computational-physicist-aljosa-hafner-on-the-use-and-benefits-of-the-eosc/

Hackathon Goals 🥅

- Add sample datasets for ESRF, CERIC, ILL, ESS, ELI and XFEL
- Set up test cases for common api




panosc photon and neutron open science cloud

Wavelength	Chemical Formula
Start Date	Sample Name
Facility	Scientific Technique
Author	Identifier

Install Node.js and Git

For Mac use brew, for Linux use apt or yum, for windows, can install with chocolatey

https://brew.sh/

https://chocolatey.org

```
brew install git node
choco install git.install
apt install nodejs git
```

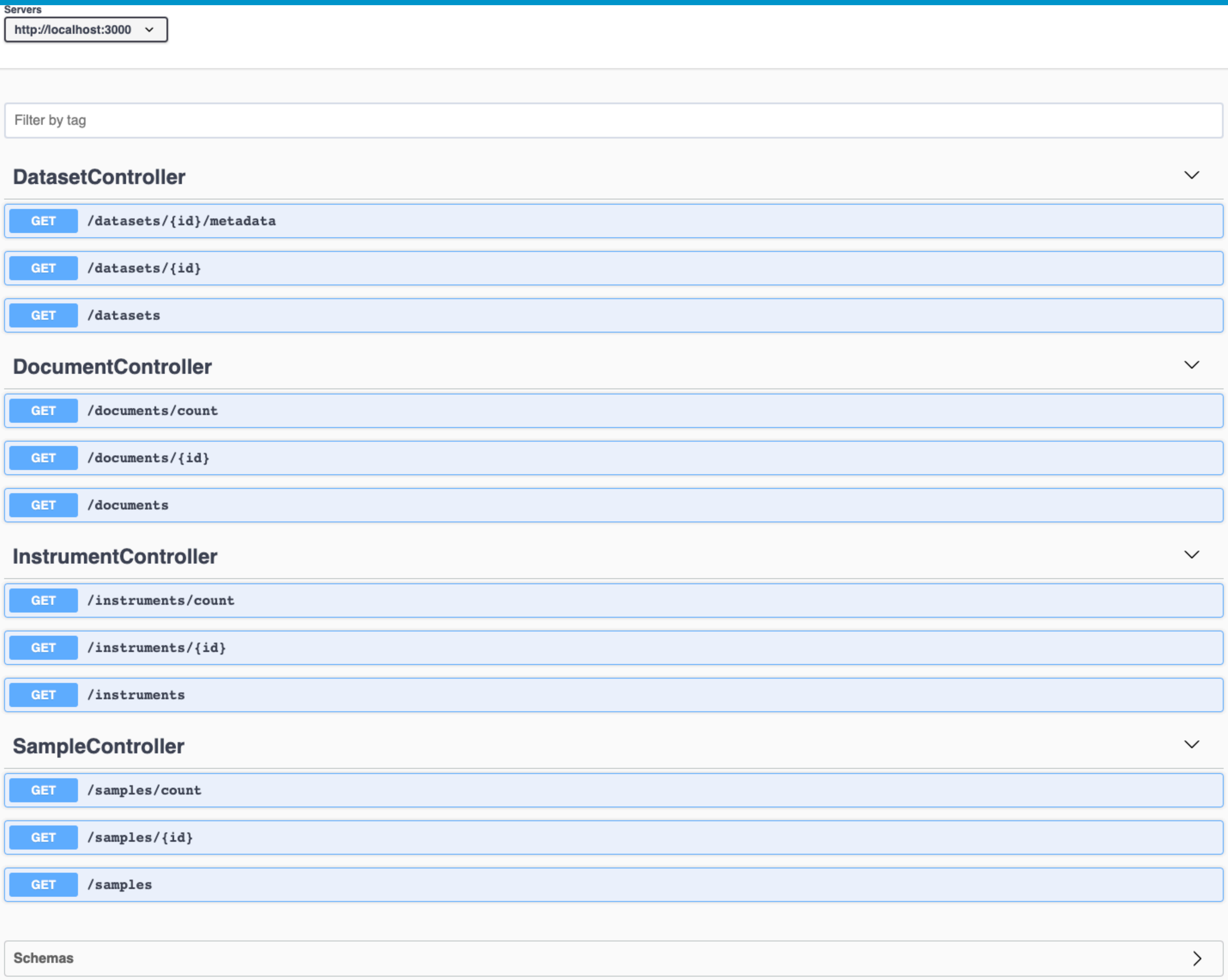
Install search-api

```
git clone https://github.com/panosc-eu/search-api.git
cd search-api
npm install
npm start
```

View explorer

- Go to
- http://localhost:3000/explorer
- Click DatasetController -> Get -> Try it Out
- Change limit to 10
- Click Execute

OpenAPI Explorer



Servers
http://localhost:3000

Filter by tag

DatasetController
GET /datasets/{id}/metadata
GET /datasets/{id}
GET /datasets

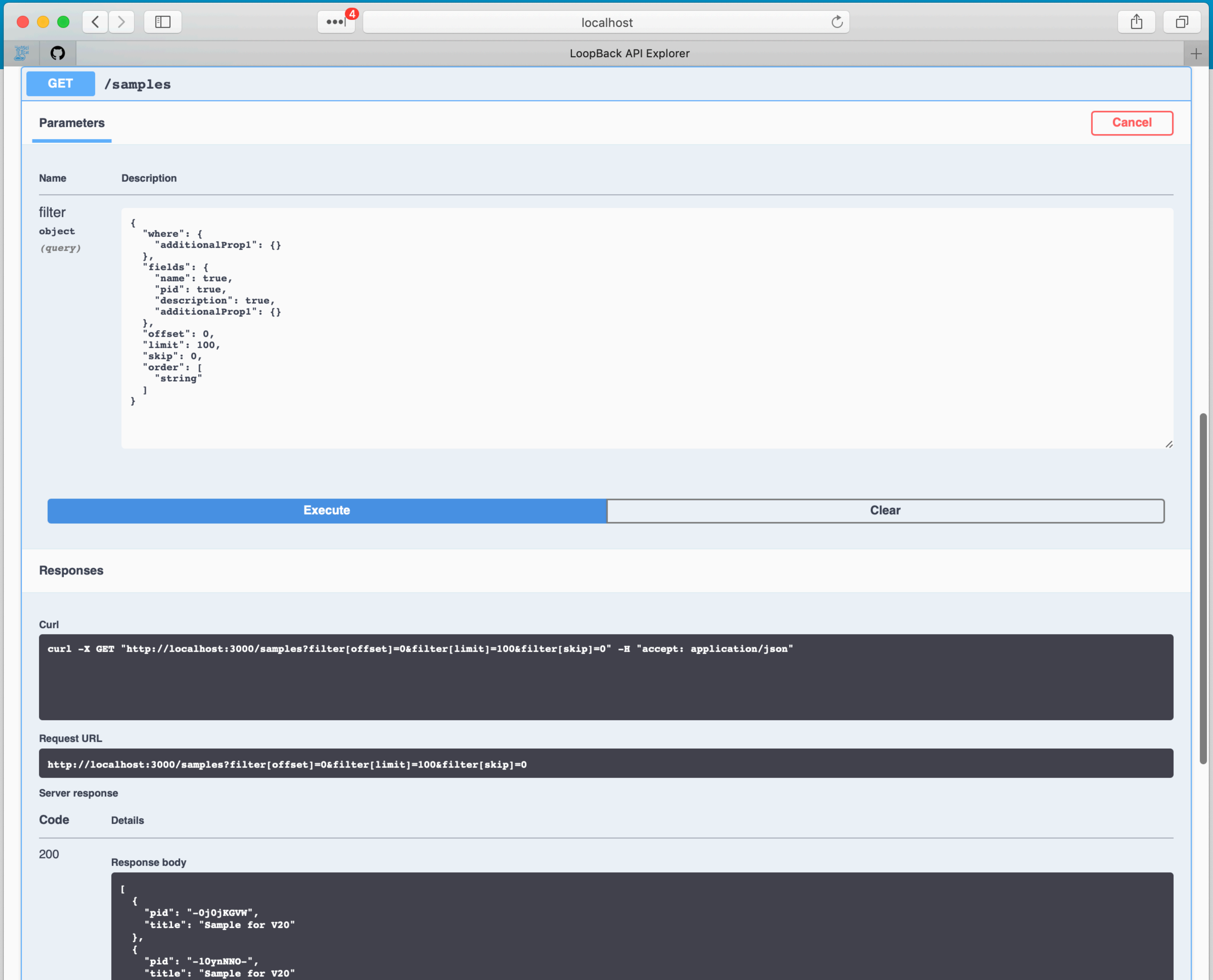
DocumentController
GET /documents/count
GET /documents/{id}
GET /documents

InstrumentController
GET /instruments/count
GET /instruments/{id}
GET /instruments

SampleController
GET /samples/count
GET /samples/{id}
GET /samples

Schemas

Fetch the samples using the GET api call



`GET /samples`

Parameters

Name	Description
filter object *(query)*	(see below)

```
{
  "where": {
    "additionalProp1": {}
  },
  "fields": {
    "name": true,
    "pid": true,
    "description": true,
    "additionalProp1": {}
  },
  "offset": 0,
  "limit": 100,
  "skip": 0,
  "order": [
    "string"
  ]
}
```

Responses

Curl

```
curl -X GET "http://localhost:3000/samples?filter[offset]=0&filter[limit]=100&filter[skip]=0" -H "accept: application/json"
```

Request URL

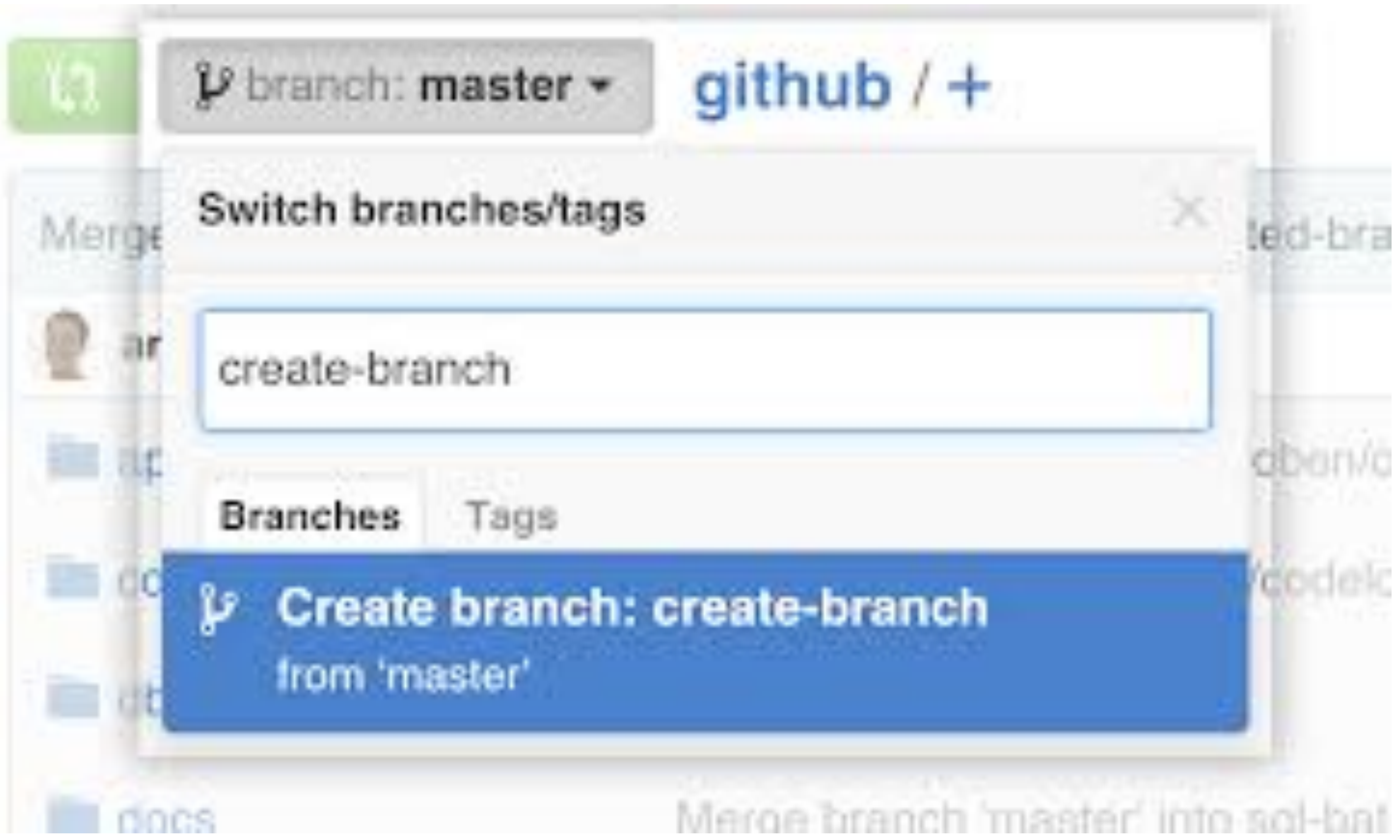
```
http://localhost:3000/samples?filter[offset]=0&filter[limit]=100&filter[skip]=0
```

Server response

Code | Details
200 | Response body

```
[
  {
    "pid": "-0j0jKGVW",
    "title": "Sample for V20"
  },
  {
    "pid": "-10ynNNO-",
    "title": "Sample for V20"
```

Make a new branch in GitHub repository



Add a test

EUROPEAN SPALLATION SOURCE

```
it('retrieves datasets with water and  pressure above 100', async () => {
  find.resolves(aListOfDatasets);
  const details = await controller.find({
    where:
   {and:
     [
      {'pressure.value': {gt: 100}},
      {sample: 'water'}
     ]},
   });
  console.log(details);
  expect(details).to.eql(aListOfDatasets);
  sinon.assert.called(find);
});
```

Goals:

1. Add data to PaNOSC search-api
2. Query data in search-api
3. Implement PaNOSC use cases as unit tests