API Validators

19th Septembre, 2019

Alejandro de Maria,

Data Manager

Data Analysis Unit

ESRF

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 823852

Task 3.1: Develop API (M1-M28)

Leader: ESS
Contributors:

- ESRF,
- ILL,
- XFEL.EU,
- ELI,
- CERIC-ERIC

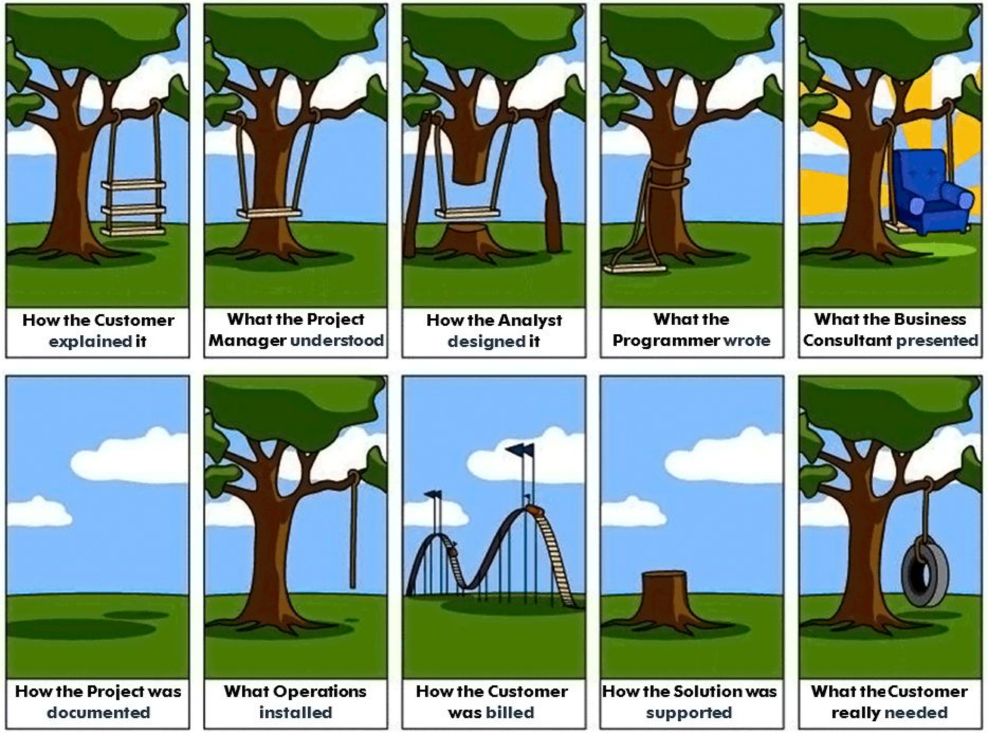
Define an API to be used in the Photon and Neutron community that will allow for FAIR exposure of the data at the individual institutions through a catalogue service.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Task 3.1: Develop API (M1-M28)

[.....] In order to test any implementation at facilities for compliance, **a set of API tests will be developed**. The test harness will be executable against a given site catalogue service and result in a **report stating the status** towards compliance.

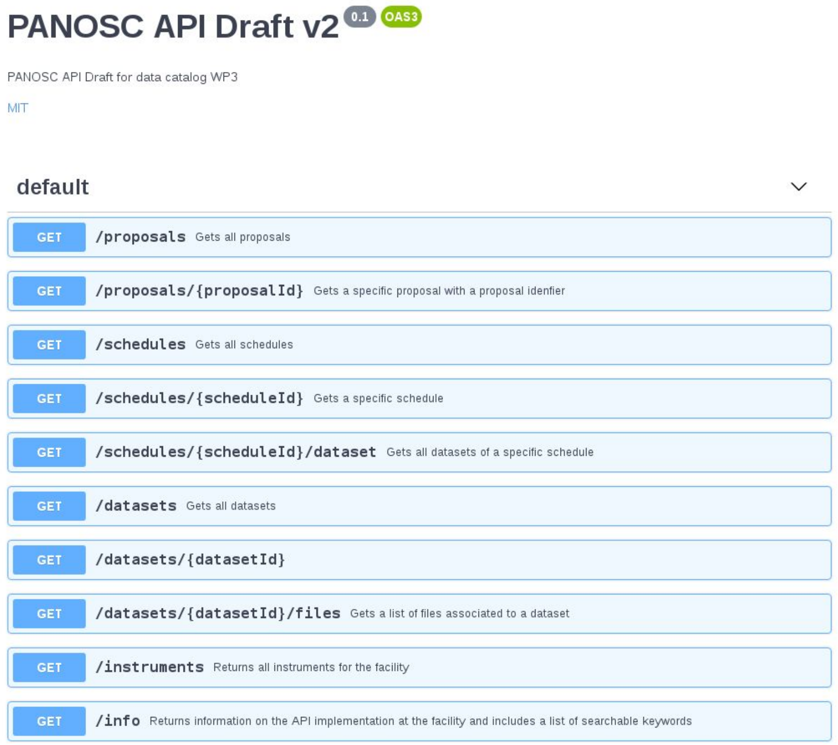
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852


How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations installed
How the Customer was billed
How the Solution was supported
What the Customer really needed

http://www.projectcartoon.com/cartoon/61940

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 823852

API: https://github.com/panosc-eu/panosc



PANOSC API Draft v2 0.1 OAS3

PANOSC API Draft for data catalog WP3

MIT

default

GET `/proposals` Gets all proposals

GET `/proposals/{proposalId}` Gets a specific proposal with a proposal idenfier

GET `/schedules` Gets all schedules

GET `/schedules/{scheduleId}` Gets a specific schedule

GET `/schedules/{scheduleId}/dataset` Gets all datasets of a specific schedule

GET `/datasets` Gets all datasets

GET `/datasets/{datasetId}`

GET `/datasets/{datasetId}/files` Gets a list of files associated to a dataset

GET `/instruments` Returns all instruments for the facility

GET `/info` Returns information on the API implementation at the facility and includes a list of searchable keywords

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

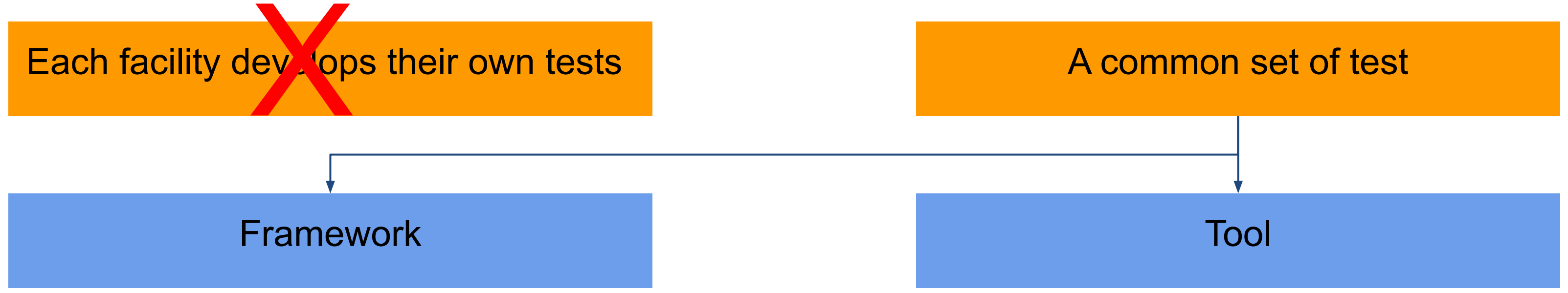
Choices to be made

Each facility develops their own tests

A common set of tests

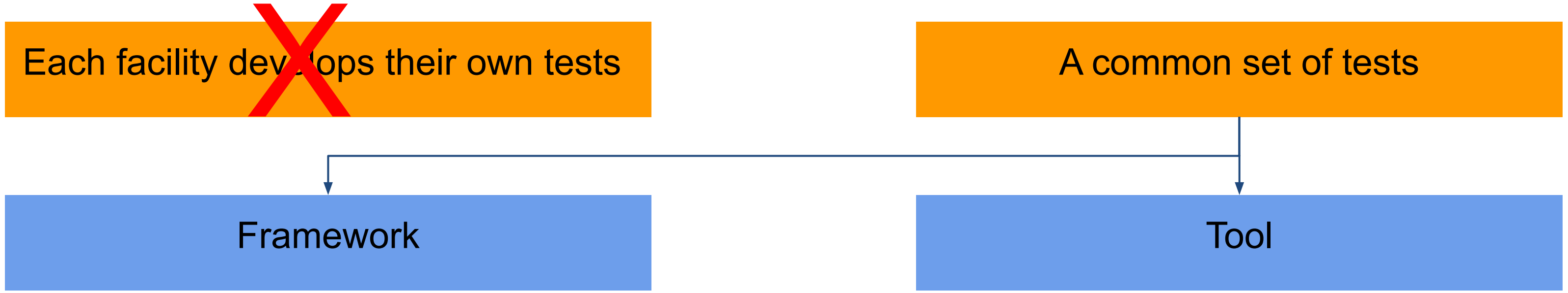
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 823852

Choices to be made



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Choices to be made

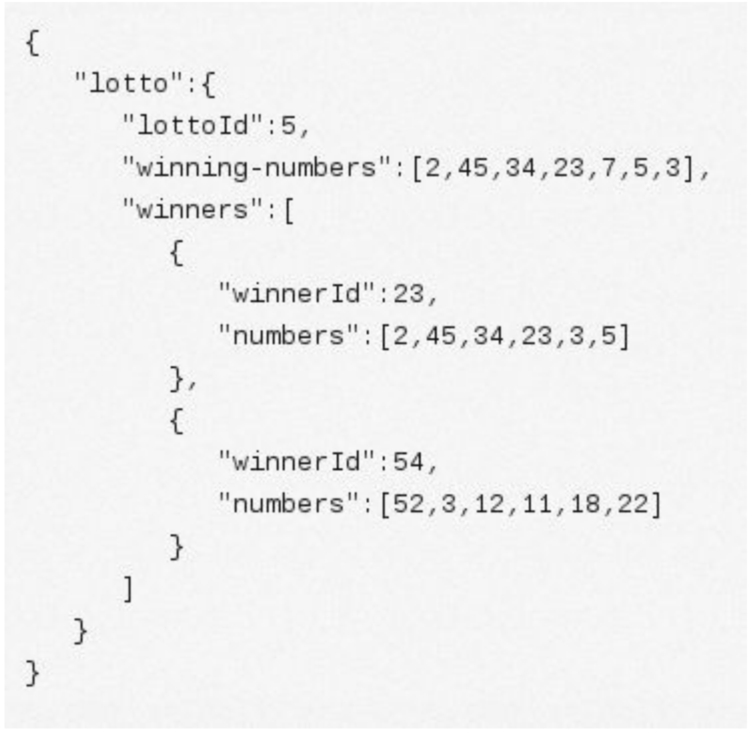


REST-assured Chakram Tavern

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

panosc

"http://localhost:8080/lotto/{id}":



```
{
   "lotto":{
      "lottoId":5,
      "winning-numbers":[2,45,34,23,7,5,3],
      "winners":[
         {
            "winnerId":23,
            "numbers":[2,45,34,23,3,5]
         },
         {
            "winnerId":54,
            "numbers":[52,3,12,11,18,22]
         }
      ]
   }
}
```

REST-assured

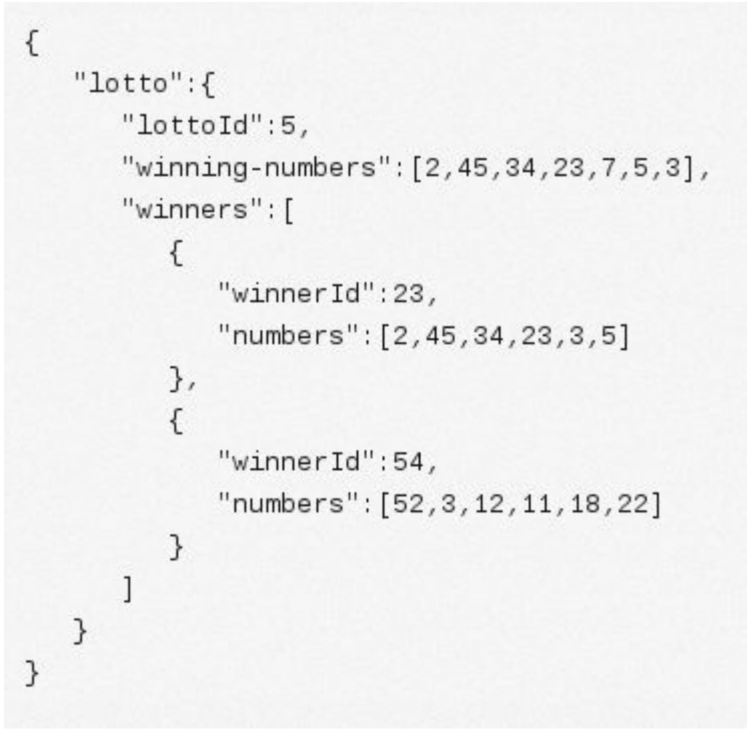
build passing maven central 4.1.1 javadoc 4.1.1

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 823852

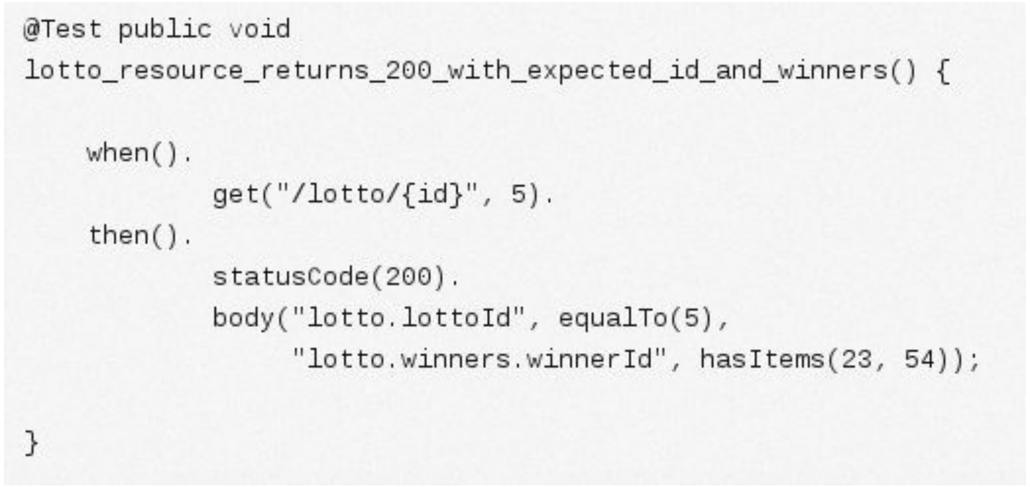
"http://localhost:8080/lotto/{id}":

REST-assured

build passing maven central 4.1.1 javadoc 4.1.1



```
{
   "lotto":{
      "lottoId":5,
      "winning-numbers":[2,45,34,23,7,5,3],
      "winners":[
         {
            "winnerId":23,
            "numbers":[2,45,34,23,3,5]
         },
         {
            "winnerId":54,
            "numbers":[52,3,12,11,18,22]
         }
      ]
   }
}
```



```
@Test public void
lotto_resource_returns_200_with_expected_id_and_winners() {

    when().
            get("/lotto/{id}", 5).
    then().
            statusCode(200).
            body("lotto.lottoId", equalTo(5),
                 "lotto.winners.winnerId", hasItems(23, 54));

}
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

panosc

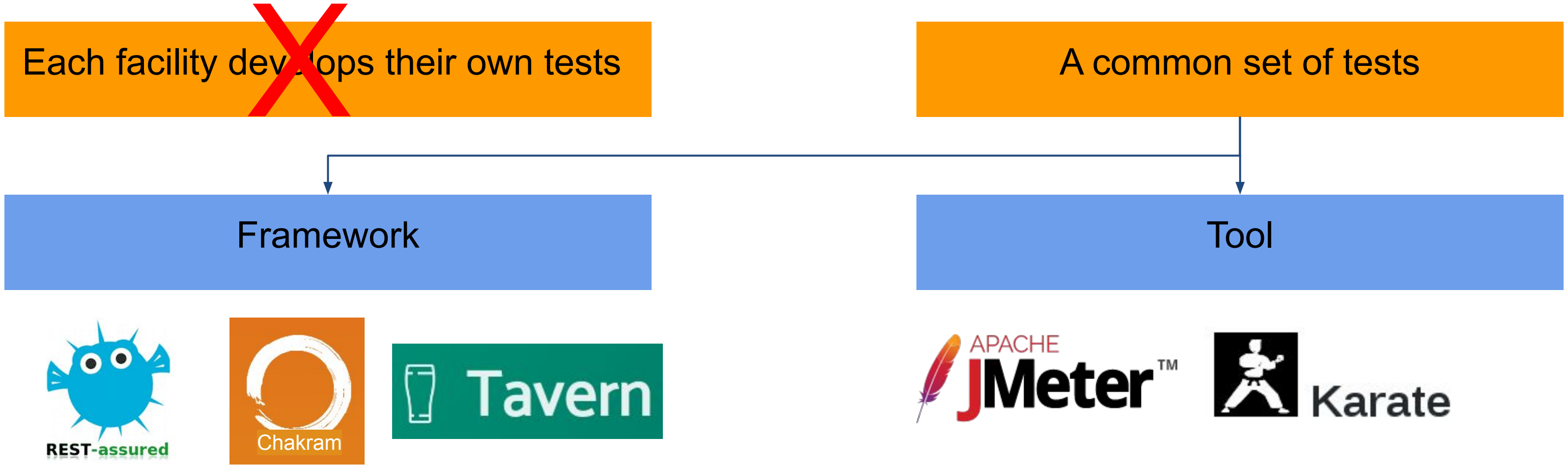
Chakram is a REST API testing framework

```
describe("HTTP assertions", function () {
  it("should make HTTP assertions easy", function () {
    var response = chakram.get("http://httpbin.org/get?test=chakram");
    expect(response).to.have.status(200);
    expect(response).to.have.header("content-type", "application/json");
    expect(response).not.to.be.encoded.with.gzip;
    expect(response).to.comprise.of.json({
      args: { test: "chakram" }
    });
    return chakram.wait();
  });
});
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

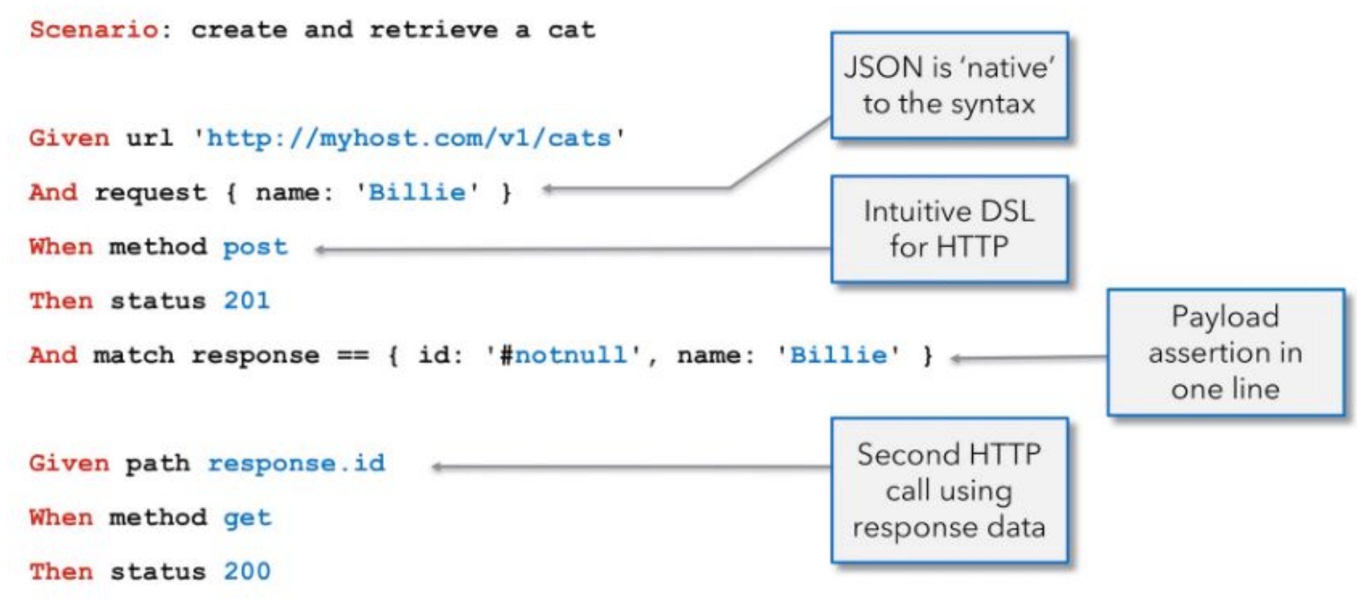
panosc

Choices to be made



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Karate



```
Scenario: create and retrieve a cat

Given url 'http://myhost.com/v1/cats'
And request { name: 'Billie' }
When method post
Then status 201
And match response == { id: '#notnull', name: 'Billie' }

Given path response.id
When method get
Then status 200
```

JSON is 'native' to the syntax
Intuitive DSL for HTTP
Payload assertion in one line
Second HTTP call using response data

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

TDD (Test Driven Development)

The TDD process consists of the following steps:

1. Start by writing a test
2. Run the test and any other tests. At this point, your newly added test should fail. If it doesn't fail here, it might not be testing the right thing and thus has a bug in it.
3. Write the minimum amount of code required to make the test pass
4. Run the tests to check the new test passes
5. Optionally refactor your code
6. Repeat from 1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852

Conclusion

- Many testing tools and/or frameworks are simple and ready to be used
- If number of end points of the API is small then:
  - Less testing to be done
  - Less documentation to write down
  - Less code to maintain
- API should not be a moving target

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 823852