



# Hands-on session: deploying the INSPIRE Reference Validator and using the API

**Carlos Palma Zurita, Enrique Soriano, Juan Martín Pelegrina,  
Daniel Navarro Cueto**

*INSPIRE 2020 Online Conference – June 9, 2020*

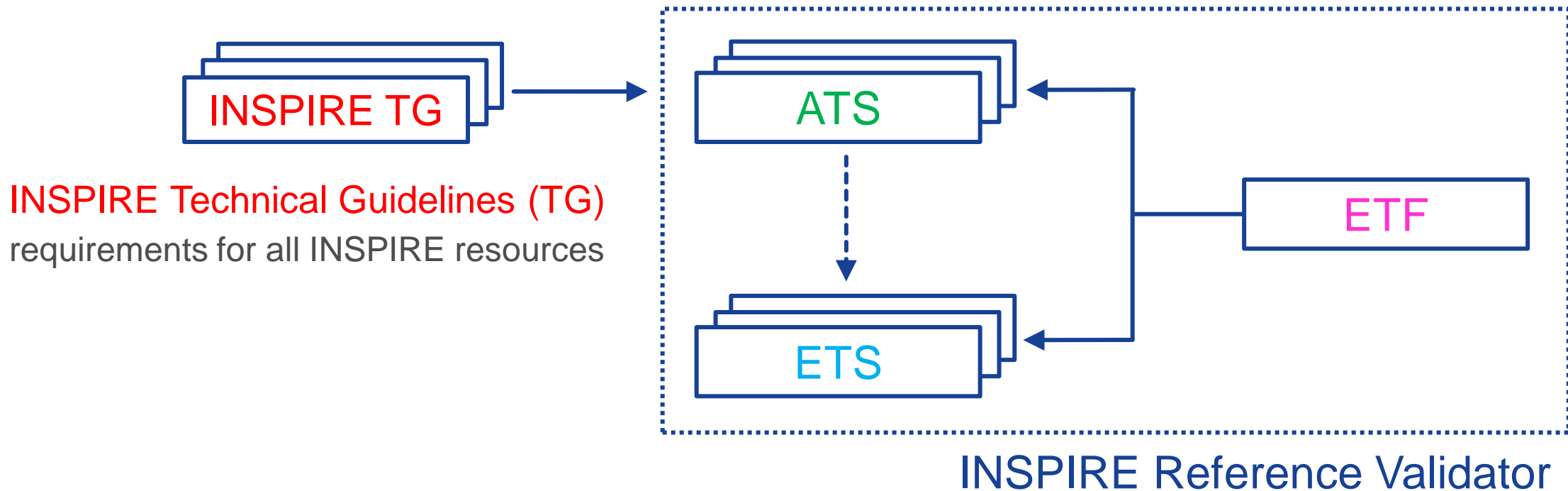
# INSPIRE Reference Validator

Abstract Test Suites (ATS)

high-level descriptions of test cases

Testing Framework (ETF)

software where ETS are run



INSPIRE Technical Guidelines (TG)

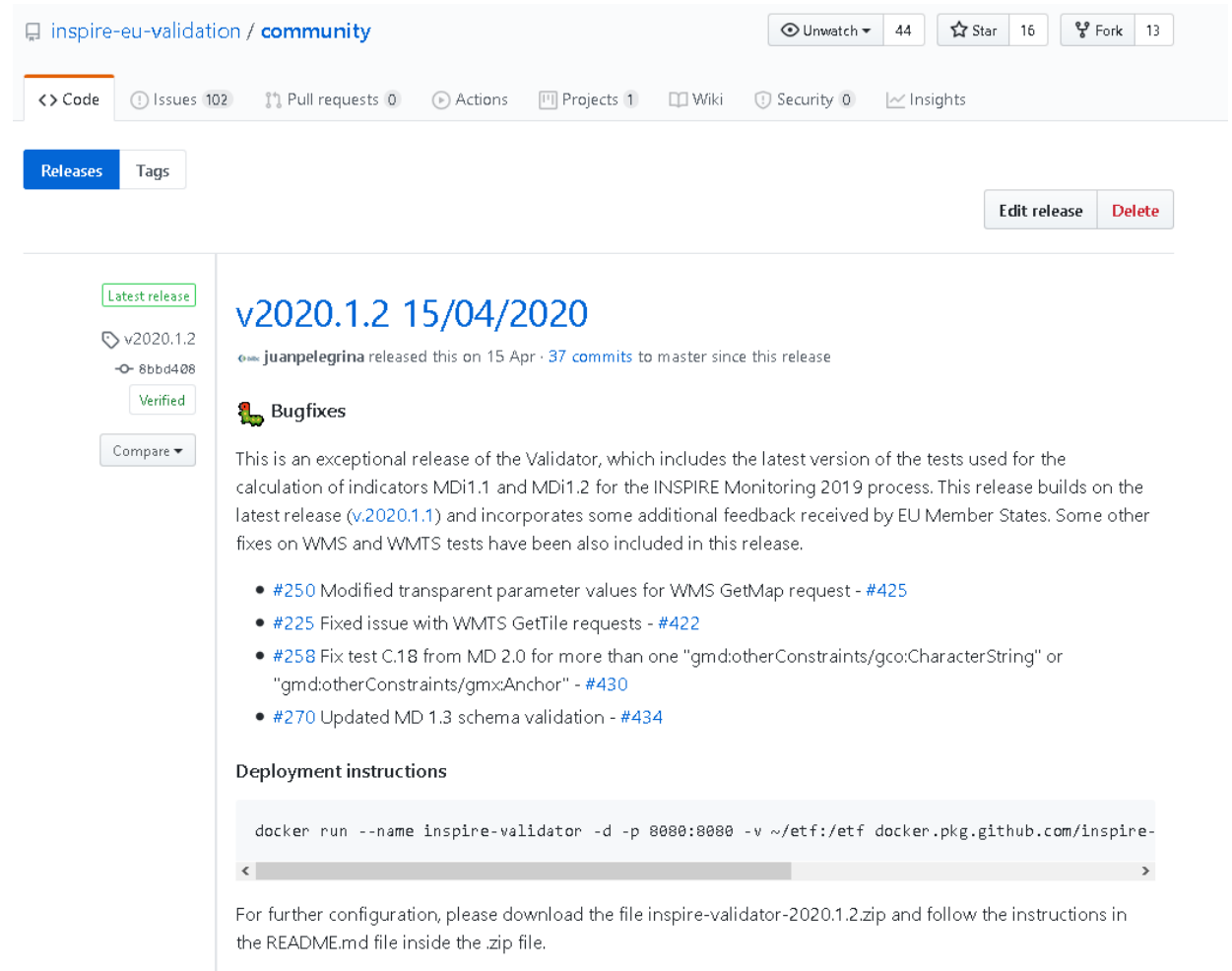
requirements for all INSPIRE resources

Executable Test Suites (ETS)

lower-level descriptions of test cases

# Release planning strategy

- The latest release of the INSPIRE Validator (v2020.1.2) is available in the [Code](#) section of the GitHub community space.
- This section includes the latest modifications introduced by all releases, and all the files and instructions to deploy your own Validator instance.



The screenshot shows the GitHub interface for the repository 'inspire-eu-validation / community'. The 'Releases' tab is selected, displaying the latest release 'v2020.1.2' dated '15/04/2020'. The release is marked as 'Latest release' and 'Verified'. It was released by 'juanpelegrina' on 15 Apr with 37 commits to master. The release includes bugfixes and is described as an exceptional release that includes the latest version of tests and incorporates feedback from EU Member States. A list of bugfixes is provided, including issues #250, #225, #258, and #270. Deployment instructions are shown as a code block, and a note at the bottom instructs users to download the zip file and follow the README instructions.

inspire-eu-validation / community

Unwatch 44 Star 16 Fork 13

<> Code Issues 102 Pull requests 0 Actions Projects 1 Wiki Security 0 Insights

Releases Tags

Edit release Delete

Latest release

v2020.1.2  
8bbd408  
Verified

Compare

v2020.1.2 15/04/2020

juanpelegrina released this on 15 Apr · 37 commits to master since this release

Bugfixes

This is an exceptional release of the Validator, which includes the latest version of the tests used for the calculation of indicators MDi1.1 and MDi1.2 for the INSPIRE Monitoring 2019 process. This release builds on the latest release (v.2020.1.1) and incorporates some additional feedback received by EU Member States. Some other fixes on WMS and WMTS tests have been also included in this release.

- #250 Modified transparent parameter values for WMS GetMap request - #425
- #225 Fixed issue with WMTS GetFile requests - #422
- #258 Fix test C.18 from MD 2.0 for more than one "gmd:otherConstraints/gco:CharacterString" or "gmd:otherConstraints/gmx:Anchor" - #430
- #270 Updated MD 1.3 schema validation - #434

Deployment instructions

```
docker run --name inspire-validator -d -p 8080:8080 -v ~/etf:/etf docker.pkg.github.com/inspire-
```

For further configuration, please download the file inspire-validator-2020.1.2.zip and follow the instructions in the README.md file inside the .zip file.

# Content of each release





- **Docker image** with a pre-built version of the INSPIRE Validator (shown in the code snippet in the deployment instructions)
- ZIP files including:
  - the ETS for Conformance Classes (ets-repository-2020.1.2.zip)
  - the Dockerfile and built .war file of the Validator (inspire-validator-2020.1.2.zip)
    - The other two files hold the contents of the Community space itself, no code inside of them

## Deployment instructions

```
docker run --name inspire-validator -d -p 8080:8080 -v ~/etf:/etf docker.pkg.github.com/inspire-
```

For further configuration, please download the file inspire-validator-2020.1.2.zip and follow the instructions in the README.md file inside the .zip file.

## Assets 4

 <a href="#">ets-repository-2020.1.2.zip</a>	1.85 MB
 <a href="#">inspire-validator-2020.1.2.zip</a>	165 MB
 <a href="#">Source code (zip)</a>	
 <a href="#">Source code (tar.gz)</a>	

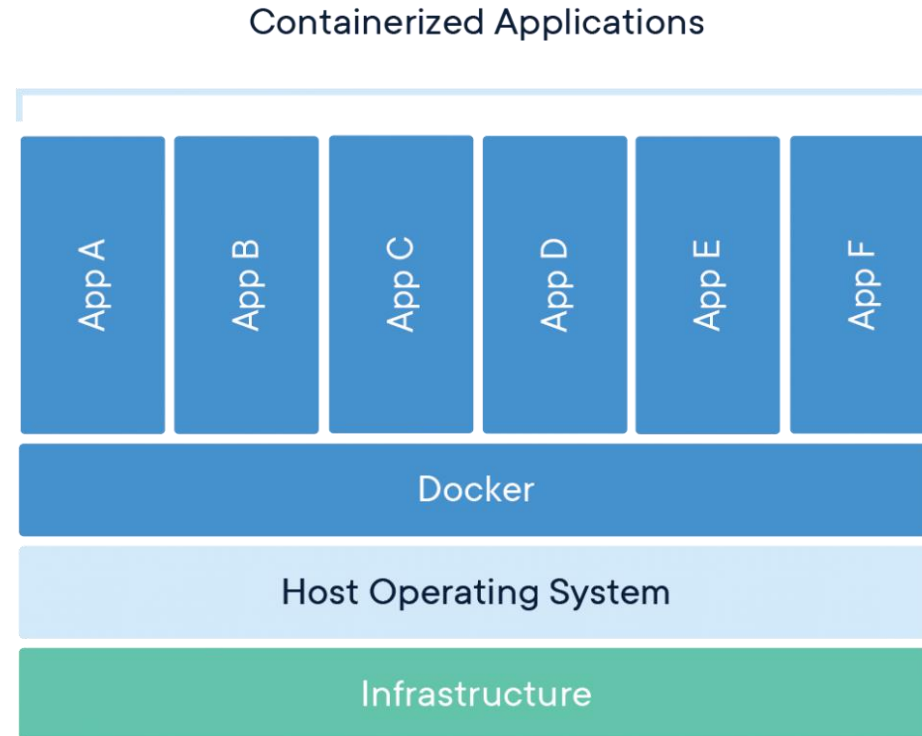
# Documentation resources

- **Instructions & references** can be found in the README file inside the Validator .zip file.
- Instructions explain how to deploy manually and customize your instance.
- Also a link to external documentation of the ETF is provided, for general issues not directly related with the INSPIRE Validator.

# Deployment options

1. Execute the Docker Image (as stated on «Deployment instructions») that contains the ETS for Conformance Classes and all the default configuration for local deployments (easiest option).
2. Build the Docker image and automatically create the Validator file system and apply configurations (recommended for production instances) using the file «inspire-validator-2020.1.2.zip».
3. Deploy the .war from «inspire-validator-2020.1.2.zip» file in an application server (i.e. Tomcat or Jetty) and follow the instructions to create the file system, apply properties and include ETS from ets-repository-2020.1.2.zip (full customization).

# Docker containers



- A Docker container is a closed **virtual environment** with all the necessary software and dependencies that is able to run inside another system.
- It is used to automatically configure the environment, keep control of all the components and be able to quickly launch and stop applications

# Contents of the Validator Docker image

- Definition can be found in the [Dockerfile](#), that sets up the tools and downloads external dependencies. It also copies all the necessary files inside the image. This file comes inside the Validator .zip file from the release.
- For the INSPIRE Validator, contents are:
  - environment variables defined for the ETF
  - source of ETS
  - built version of the ETF webapp
  - configuration proxy cache for external schemas
  - configuration of the HTTPS redirection for INSPIRE
  - scheduling of cleanup operations

```
# Possible values: "latest", <version as MAJOR.MINOR.BUGFIX> or
# <version as MAJOR.MINOR>
# Packed with the Webapp
ENV ETF_TESTDRIVER_SUI_VERSION latest

# Possible values: "latest", <version as MAJOR.MINOR.BUGFIX> or
# <version as MAJOR.MINOR>
# Packed with the Webapp
ENV ETF_TESTDRIVER_TE_VERSION latest

# Default repository configuration
ENV REPO_URL https://services.interactive-instruments.de/etfdev-af/etf-public-dev
ENV REPO_USER etf-public-dev
ENV REPO_PWD etf-public-dev

# Possible values: "none" or URL to ZIP file
ENV ETF_DL_TESTPROJECTS_ZIP https://github.com/inspire-eu-validation/ets-repository/archive/staging.zip

# Subfolder in the projects directory
ENV ETF_DL_TESTPROJECTS_DIR_NAME inspire-ets-repository
# Possible values: true for overwriting the directory on every container start,
# false for keeping an existing directory
ENV ETF_DL_TESTPROJECTS_OVERWRITE_EXISTING true

# Maximum JAVA heap size (XmX parameter) in MB or "max" (max available memory-768MB if at least 3GB ava
ENV MAX_MEM max
```



# Quick setup

- From the [Packages](#) section of the GitHub community space, a hosted Docker image can be downloaded and executed with a single command.
- This image is configured to use the corresponding test files for the release.
- The webapp runs automatically on the port 8080, which can be mapped to any port on the host machine.
- This command also creates a volume on the file system, to persist the file system in case of shutdown.

```
docker run --name inspire-validator -d -p 8080:8080 -v ~/etf:/etf docker.pkg.github.com/inspire-eu-validation/community/inspire-validator:2020.1.2
```

# Customization of deployment

- Source ETS can be modified on the Dockerfile to point to modified tests.
- Proxy server and redirection can be configured on the [res/entrypoint.sh](#) file.
- The base URL and the available resources such as CPU usage and memory allocated must be modified inside the [validator.war](#) file, in the file [WEB-INF/classes/config.properties](#).
- After this, the Docker build should be executed in the Dockerfile folder, to generate a new Docker image.

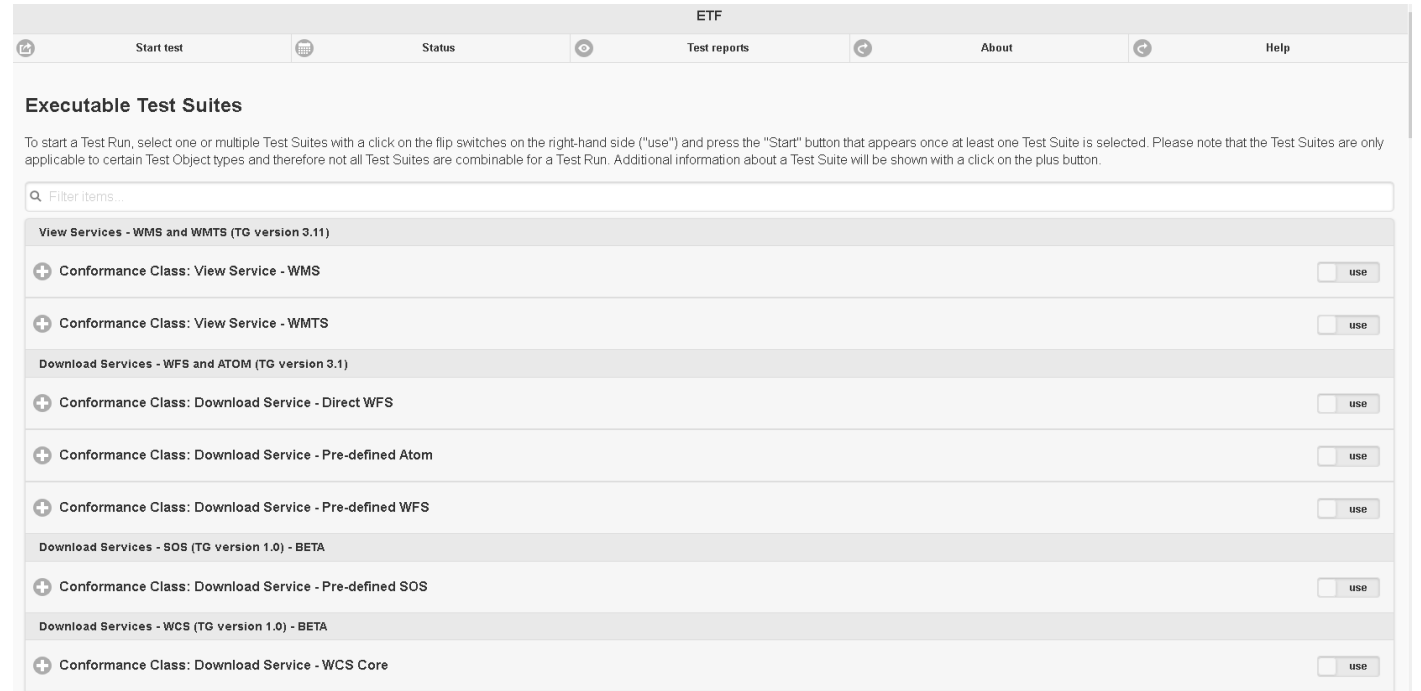
# Manual deployment

- Deploy the [validator.war](#) file using an application server such as Tomcat.
- Following the instructions in the official ETF repository, a [properties](#) file should be created and the filesystem should be created for the validator.
- You need to download the official release of the ETS and include it on the instance.
- No proxy cache or redirection is configured in this option.

<http://docs.etf-validator.net/v2.0/index.html>

# Usage of the INSPIRE Validator

- Once the Docker container is started, the Validator can be accessed at [localhost:8080/validator](http://localhost:8080/validator).
- TestRuns can be created in the same manner as in the production instance.
- If a volume is not used when starting the container, all TestReports will be lost after shutdown.



The screenshot displays the web interface of the INSPIRE Validator. At the top, there is a navigation bar with the title 'ETF' and several menu items: 'Start test', 'Status', 'Test reports', 'About', and 'Help'. Below the navigation bar, the main content area is titled 'Executable Test Suites'. A brief instruction reads: 'To start a Test Run, select one or multiple Test Suites with a click on the flip switches on the right-hand side ("use") and press the "Start" button that appears once at least one Test Suite is selected. Please note that the Test Suites are only applicable to certain Test Object types and therefore not all Test Suites are combinable for a Test Run. Additional information about a Test Suite will be shown with a click on the plus button.' Below this instruction is a search bar labeled 'Filter items...'. The main content area lists several test suites, each with a plus icon on the left and a 'use' button on the right. The test suites are grouped into categories: 'View Services - WMS and WMTS (TG version 3.11)', 'Download Services - WFS and ATOM (TG version 3.1)', 'Download Services - SOS (TG version 1.0) - BETA', and 'Download Services - WCS (TG version 1.0) - BETA'. Each category contains one or more 'Conformance Class' entries.

Category	Conformance Class	Use
View Services - WMS and WMTS (TG version 3.11)	+ Conformance Class: View Service - WMS	use
	+ Conformance Class: View Service - WMTS	use
Download Services - WFS and ATOM (TG version 3.1)	+ Conformance Class: Download Service - Direct WFS	use
	+ Conformance Class: Download Service - Pre-defined Atom	use
	+ Conformance Class: Download Service - Pre-defined WFS	use
Download Services - SOS (TG version 1.0) - BETA	+ Conformance Class: Download Service - Pre-defined SOS	use
Download Services - WCS (TG version 1.0) - BETA	+ Conformance Class: Download Service - WCS Core	use

# API Documentation



default (/v2/api-docs) ▾

Explore

## ETF Web API

This is an interactive documentation and a web user interface for interacting with the Web API version 2 of the test framework [ETF](#). This semi-automatic generated documentation covers basic functionality, but consulting the [API Documentation](#) may be required to get a deeper understanding of the ETF model and further procedures. Issues can be reported in [GitHub](#).

Content negotiation is not supported and therefore JSON is always returned for endpoints without file extension. For most operations, a link to the XML response schema is provided in the implementation nodes. JSON responses are derived from XML the response schema, based on this [stylesheet](#).

[Back to user interface](#)

Created by ETF Team  
See more at <http://www.etf-validator.net>  
[Contact the developer](#)  
[European Public License 1.2](#)

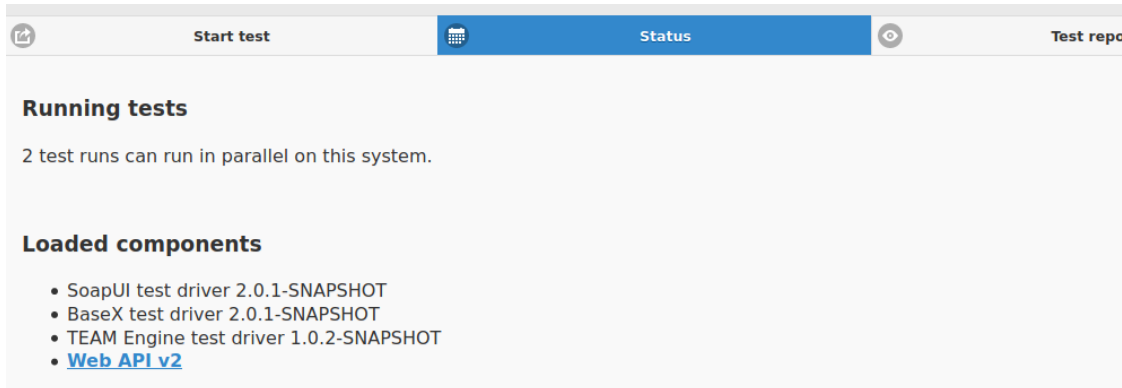
<b>1. Service Status : Monitor service workload and health</b>	Show/Hide	List Operations	Expand Operations
<b>2. Service Capabilities : Retrieve test framework metadata</b>	Show/Hide	List Operations	Expand Operations
<b>3. Manage Test Objects : Define Test Objects and upload test data</b>	Show/Hide	List Operations	Expand Operations
<b>4. Manage Test Runs : Start and control test runs</b>	Show/Hide	List Operations	Expand Operations
<b>5. Test Run Results : Retrieve test results</b>	Show/Hide	List Operations	Expand Operations
GET /v2/TestRuns.json			Get multiple Test Results as JSON
GET /v2/TestRuns.xml			Get multiple Test Results as XML
DELETE /v2/TestRuns/{id}			Cancel and delete a Test Run
GET /v2/TestRuns/{id}			Get a single Test Result as JSON
HEAD /v2/TestRuns/{id}			Check if the Test Run exists
GET /v2/TestRuns/{id}.html			Generate a HTML Test Report

<https://inspire.ec.europa.eu/validator/swagger-ui.html>

# API Usage

- Check the status of the service, i.e. if there is thread run available

<https://inspire.ec.europa.eu/validator/#home>



The screenshot shows the 'Status' page of the Inspire Validator. The page has a navigation bar with 'Start test', 'Status', and 'Test repo' tabs. The main content area is divided into two sections: 'Running tests' and 'Loaded components'. Under 'Running tests', it states '2 test runs can run in parallel on this system.' Under 'Loaded components', it lists: SoapUI test driver 2.0.1-SNAPSHOT, BaseX test driver 2.0.1-SNAPSHOT, TEAM Engine test driver 1.0.2-SNAPSHOT, and [Web API v2](#).

```
curl -X GET \
https://inspire.ec.europa.eu/validator/v
2/status
```

```
name: "ETF"
status: "GOOD"
heartbeat: "1591094136025"
willExpireAt: "1591094296025"
version: "2.0.0-b191118T1324"
uptime: "36522"
allocatedMemory: "960746832"
presumableFreeMemory: "13485222576"
totalSpace: "9223372036853727232"
freeSpace: "9223372035071148032"
cpuLoad: "0.000250"
```

# API Usage

- Locate TestSuite IDs

```
curl -X GET http://inspire.ec.europa.eu/validator/v2/ExecutableTestSuites
```

```
▼ EtfItemCollection:
  version: 2
  returnedItems: 67
  position: 0
  ▼ ref: "https://inspire.ec.europa.eu/validator/v2/ExecutableTestSuites.json?offset=0&Limit=1500"
  ▼ executableTestSuites:
    ▼ ExecutableTestSuite:
      ▼ 0:
        id: "EID59692c11-df86-49ad-be7f-94a1e1ddd8da"
        ▶ remoteResource: "https://github.com/inspire-tree/master/metadata/2.0"
        ▶ label: "Common Requirements for ...SPIRE metadata records."
        ▶ description: "<br/><br/>Please report ...ta records</a><br/><br/>"
        reference: ".../inspire-md-bsxets.xq"
        version: "1.0.6"
        author: "Consortium Bilbomatica, Guadaltel y Geograma"
        createDate: "2018-06-30T00:00:00Z"
        lastEditor: "Consortium Bilbomatica, Guadaltel y Geograma"
        lastUpdateDate: "2020-04-01T17:30:00Z"
        disabled: false
        ▶ tags: {...}
        ▶ testDriver: {...}
        ▶ translationTemplateBundle: {...}
        ▶ parameterList: {...}
        ▶ supportedTestObjectTypes: {...}
        ▶ testModules: {...}
      ▶ 1: {...}
      ▼ 2:
        id: "EID2be1480a-fe42-40b2-9420-eb0e69385c80"
        ▶ remoteResource: "https://github.com/inspire-tree/master/metadata/2.0"
        ▶ label: "Conformance Class 2: INS_eroperability metadata."
        ▶ description: "This test suite examines...ses: Conformance Class 1"
        reference: ".../inspire-md-bsxets.xq"
        version: "1.0.0"
        author: "Consortium Bilbomatica, Guadaltel y Geograma"
        createDate: "2018-06-30T00:00:00Z"
        lastEditor: "Consortium Bilbomatica, Guadaltel y Geograma"
        lastUpdateDate: "2019-06-06T18:12:00Z"
        disabled: false
```

# API Usage

- Create TestObject

```
curl -X POST 'http://inspire.ec.europa.eu/validator/v2/TestObjects?action=upload' -F fileupload=@[PATH/TO/LOCAL/FILE]
```

```
{
  "testObject": {
    "id": "EID844c4c8c-ab7a-4104-b94b-4607921119c5"
  },
  "files": [
    {
      "name": "xml.metadata.get.xml",
      "size": "24625",
      "type": "application/xml"
    }
  ]
}
```



# API Usage

- Create TestRun

```
curl -X POST http://inspire.ec.europa.eu/validator/v2/TestRuns -d
'{"label":"Metadata test with uploaded TestObject",
"executableTestSuiteIds":["EID59692c11-df86-49ad-be7f-94a1e1ddd8da"],
"arguments":{"files_to_test":".*", "tests_to_execute":".*"},
"testObject":{"id":"EID844c4c8c-ab7a-4104-b94b-4607921119c5"}}'
```

```
{
  "EtfItemCollection": {
    "version": 2.0,
    "returnedItems": 1,
    "ref": "http://inspire.ec.europa.eu/validator/v2/TestRuns/e7653a13-4235-4254-bc50-eaddbb587053.json",
    "testRuns": {
      "TestRun": {
        "id": "EIDe7653a13-4235-4254-bc50-eaddbb587053",
        "status": "UNDEFINED",
        "label": "Metadata test with uploaded TestObject",
        "defaultLang": "en",
        "startTimestamp": "2019-10-07T13:31:25.821Z",
```

# API Usage

- Check the status and wait

<http://inspire.ec.europa.eu/validator/v2/TestRuns/EIDe7653a13-4235-4254-bc50-eaddbb587053>

```
{
  "EtfItemCollection": {
    "version": 2.0,
    "returnedItems": 1,
    "ref": "http://inspire.ec.europa.eu/validator/v2/TestRuns/e7653a13-4235-4254-bc50-eaddbb587053.json",
    "testRuns": {
      "TestRun": {
        "id": "EIDe7653a13-4235-4254-bc50-eaddbb587053",
        "status": "UNDEFINED",
        "label": "Metadata test with uploaded TestObject",
        "defaultLang": "en",
        "startTimestamp": "2019-10-07T13:31:25.821Z",
        "testTasks": {
          "TestTask": {
            "id": "EIDb6da9149-aff5-4d0b-b940-ea4f325934c1",
            "parent": {
              .
              .
              .
            }
          }
        }
      }
    }
  }
}
```

# API Usage

- Once the status is either PASSED or FAILED, we can get the full report in HTML using the same request as before adding .html at the end.
- You can also use the JSON representation of the TestRun to access each TestTask, that is, each Conformance Class, under the key

`testRuns.TestRun.testTasks.TestTask`

- Execute this request  
[http://inspire.ec.europa.eu/validator/v2/TestTaskResults/\[TEST\\_TASK\\_ID\].\[json|xml|html\]](http://inspire.ec.europa.eu/validator/v2/TestTaskResults/[TEST_TASK_ID].[json|xml|html])  
to access the result of individual assertions in the Conformance classes.

# Recommendations

- Bulk validation:
  - for a **small metadata collection**, the best way to proceed is to generate a .zip file with all the records;
  - for a **large metadata collection**, executions of parallel TestRuns may offer the best performance:
    - using a scripting language or an ETL tool, a **pipeline** can be set up to create test runs in bulk and collect the results
      - the JRC will release soon (as a new GitHub repository) a pipeline for massive metadata validation based on Pentaho Data Integration (PDI)
    - you need to take into account the limitation on available threads on the Validator

# Recommendations

- Known issues:
  - **geometry validation** has some issues, arising from the integration of deegree into the Validator:
    - fixes will be included in next release of ETF (expected for July 2020)

<https://github.com/inspire-eu-validation/community/issues/267>

<https://github.com/inspire-eu-validation/community/issues/301>

- **redirection** from HTTP to HTTPS in the INSPIRE Validator to access codelists from the INSPIRE Registry:
  - in the Docker configuration, it is **already setup**
  - in manual deployments, this has to be **setup manually**

<https://github.com/inspire-eu-validation/community/issues/321>

<https://github.com/inspire-eu-validation/community/issues/309>

# Thank you!



marco.minghini@ec.europa.eu



© European Union 2020

Unless otherwise noted the reuse of this presentation is authorised under the [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license. For any use or reproduction of elements that are not owned by the EU, permission may need to be sought directly from the respective right holders.





# Slido session 3

*INSPIRE 2020 Online Conference – June 9, 2020*

Joint  
Research  
Centre



# Keep in touch



EU Science Hub: [ec.europa.eu/jrc](https://ec.europa.eu/jrc)



@EU\_ScienceHub



EU Science Hub – Joint Research Centre



EU Science, Research and Innovation



Eu Science Hub