

Starter Kit for users, developers and admins

JRC INSPIRE Team


May 31, 2024

INSPIRE Reference Validator: Starter Kits


- Users
 - General Operation
 - Test run example
 - From a failed test run to a valid test run
 - INSPIRE Reference Validator UI and ETF UI
 - Validator UI and ETF API
- Developers
 - Understanding ETF Set-up
 - Architecture: components, endpoints, ...
 - ETF folder analysis
 - Deploy ETF using Jetty
 - Build ETF from source code
 - Creating an ETF workspace
 - API usage
 - Understanding Swagger
 - API TestRun
 - Develop an ETS
- Admins
 - Deploy ETF
 - Docker and Validator
 - Introduction to docker
 - Deploy Validator with Docker
 - Understanding Validator Docker ZIP
 - How to build your own Validator Docker Image
 - Parameters and configuration

INSPIRE Reference Validator: Users

General operation



European Commission

 English

Inspire

INSPIRE Reference Validator

[Home](#)
[Test selection](#)
[Test reports](#)
[Get support](#)
[More on the INSPIRE Reference Validator](#)


[European Commission](#) >
 [INSPIRE](#) >
[Validator](#) >
[Home](#)

Welcome to the INSPIRE Reference Validator

The purpose of the INSPIRE Reference Validator is to help data providers, solution providers and national coordinators to check whether data sets, network services and metadata meet the requirements defined in the INSPIRE Technical Guidelines. The Validator provides detailed test reports to help implementers understand how well their data, services, metadata or software solutions are doing (or where improvements may be needed).

The Validator is based on the [Abstract and Executable Test Suites](#) agreed between Member States and the Commission in the INSPIRE Maintenance and Implementation Group, and includes a [helpdesk](#) to address feedback, bug reports and feature requests from the INSPIRE community. See the [changelog of the current and past releases](#) and the [release planning strategy](#) for the plan of future releases of the INSPIRE Reference Validator.


The Validator has been developed under [ARE3NA](#) and [ELISE](#) Actions of the ISA/ISA2 Programmes.



[Test your data, services or metadata](#)

Pick your resource (data, services or metadata), select the test(s) to launch and check the results to see how well you are doing (or where you need to improve).


Start a test



API

If you are a developer, you can access and call the operations of the validator API to power your own applications.


Try the API



Want more?

Download the software, deploy it in your own infrastructure and customise it to fit your specific needs.

Get the software



Feedback

Use the Community space to provide your feedback or proposals on the ATS, ETS or the ETF test framework.

Provide your feedback

INSPIRE Reference Validator: Users

Test run example: test selection

Test selection

Configure your test

Select the INSPIRE resource you would like to test

- ☒ Metadata
- ☐ View Service
- ☐ Download Service
- ☐ Discovery Service
- ☐ Data set

Select the Technical Guidelines version

- ☐ Version 1.3 - DEPRECATED
- ☒ Version 2.0

Select the type of metadata record(s) to be tested

- ☒ Data sets and data set series
- ☐ Network Service
- ☐ Spatial Data Service

Advanced options ▾

Provide the resource to test

Select the input type and upload or link the resource

Provide a URL to a remote XML file or a ZIP file containing one or multiple XML files, either from a request to an INSPIRE Spatial Data Service endpoint or a file repository. The download will be initiated after clicking on the 'Start' button. This download has a timeout of 120s.

Remote file 

Credentials ▾


Provide a label for your test report (optional)

Your test report will appear with the label below; edit the text if you wish to change it.

Start test >


INSPIRE Reference Validator: Users

Test run example: Test run


Production Metada Test run Example

Started	5:22 PM - 28.05.2024
Status	RUNNING
Test object	https://yzqiqfakm4.execute-api.eu-west-1.amazonaws.com/validator/v2/TestRuns/EIDa84aef39-6614-4e3e-9d72-50b058c51c88.xml
Test suites	<ul style="list-style-type: none"> • Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records • Conformance Class 1: Baseline metadata for data sets and data set series • Conformance Class 2: INSPIRE data sets and data set series interoperability metadata • Conformance Class 2b: INSPIRE data sets and data set series metadata for Monitoring • Conformance Class 8: INSPIRE data sets and data set series linked service metadata

[See report](#)
[Log file](#)
[Download report](#)
[Delete report](#)
[Re-run test](#)


Production Metada Test run Example

Started	5:22 PM - 28.05.2024
Status	PASSED
Test object	https://yzqiqfakm4.execute-api.eu-west-1.amazonaws.com/validator/v2/TestRuns/EIDa84aef39-6614-4e3e-9d72-50b058c51c88.xml
Test suites	<ul style="list-style-type: none"> • Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records • Conformance Class 1: Baseline metadata for data sets and data set series • Conformance Class 2: INSPIRE data sets and data set series interoperability metadata • Conformance Class 2b: INSPIRE data sets and data set series metadata for Monitoring • Conformance Class 8: INSPIRE data sets and data set series linked service metadata

[See report](#)
[Log file](#)
[Download report](#)
[Delete report](#)
[Re-run test](#)

INSPIRE Reference Validator: Users


Test run example: Test report

European Commission > INSPIRE > Validator > Test reports		
i	Test object: Dataset_metadata_2.0_example.xml	
i	Metadata record statistics	
i	Log path: Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records.	
i	Log path: Conformance Class 1: INSPIRE data sets and data set series baseline metadata.	
i	Log path: Conformance Class 2: INSPIRE data sets and data set series interoperability metadata.	
i	Log path: Conformance Class 2b: INSPIRE data sets and data set series metadata for Monitoring	
i	Log path: Conformance Class 8: INSPIRE data sets and data set series linked service metadata	
+	Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records.	3
+	Conformance Class 1: INSPIRE data sets and data set series baseline metadata.	3
+	Conformance Class 2: INSPIRE data sets and data set series interoperability metadata.	4
+	Conformance Class 2b: INSPIRE data sets and data set series metadata for Monitoring	1
+	Conformance Class 8: INSPIRE data sets and data set series linked service metadata	1

INSPIRE Reference Validator: Users


From a failed test run to a valid test run

How to validate and correct a metadata: upload, run and obtain the test report


Test run on 17:34 - 28.05.2024 with test suite Conformance Class 8: INSPIRE data sets and data set series linked service metadata

Started	5:34 PM - 28.05.2024
Status	FAILED
Test object	https://zyyiqfakm4.execute-api.eu-west-1.amazonaws.com/validator/v2/TestRuns/EIDc8d4c47f-28e5-4455-bb34-9ff05b9ac9c3.xml
Test suites	<ul style="list-style-type: none"> • Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records • Conformance Class 1: Baseline metadata for data sets and data set series • Conformance Class 2: INSPIRE data sets and data set series interoperability metadata • Conformance Class 2b: INSPIRE data sets and data set series metadata for Monitoring • Conformance Class 8: INSPIRE data sets and data set series linked service metadata

[See report](#)
[Log file](#)
[Download report](#)
[Delete report](#)
[Re-run test](#)


Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records.
Failed: 1/3

Please report any issues or problems [in GitHub](#).

Known limitations are documented in the description of the applicable test case or test assertion.

There is a general limitation in all assertions that polymorphism and containment by reference (see the [Technical Guidance](#), sub-clauses A.3, A.4 and A.5) are not supported. However, the current Abstract Test Suite does not support polymorphism and references either (all XPath expressions do not support polymorphism or references; in addition, schema validation is only executed against the ISO/OGC schemas without extensions). It is therefore unclear if this is really a limitation or if the sections in the technical guidance are outdated.

Source: [Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records](#)

Status Failed
Duration 4 s
Version 1.0.13

+ General requirements	Failed: 1/5
+ Identification	12
+ Data quality	3
+ Conformance Class 1: INSPIRE data sets and data set series baseline metadata.	3

https://raw.githubusercontent.com/INSPIRE-MIF/helpdesk-validator/master/training%20material/2024-05-31%20JRC%20Training/Dataset_metadata_2.0_FAILED.xml

INSPIRE Reference Validator: Users

From a failed test run to a valid test run

Inspect the root cause of the issue

General requirements

Failed: 1 / 5

Execute tests that apply for all metadata records on elements that are directly child elements of the root node of the XML document: MD_Metadata node.

Status Failed
Duration 4 s

- md common req C.1: XML Schema
- md common req C.2: Root Element
- md common req C.5: Language Code
- md common req C.6: Metadata Point of Contact
- md common req C.7: Metadata Date

md common req C.1: XML Schema

md common req C.2: Root Element

md common req C.5: Language Code

Test that a resource language is given pointing to one of the official languages of the Community expressed in conformity with ISO 639-2

More information: [Metadata Language](#)

Status Failed
Duration 0.001 s

Messages

The metadata record set has 1 record(s) with errors for this assertion.
XML document 'Dataset_metadata_2_0_example.xml', record '123e4567-e89b-12d3-a456-426655440000': A dataset (series) metadata record must have a gmd:Language code within its identification information. No gmd:LanguageCode is provided in this metadata record.

INSPIRE Reference Validator: Users

From a failed test run to a valid test run

Apply changes to your metadata according to the Test Report message

```
<?xml version="1.0" encoding="UTF-8"?>
<gmd:MD_Metadata xmlns:gmd="http://www.isotc211.org/2005/gmd"
  xmlns:gco="http://www.isotc211.org/2005/gco" xmlns:gmx="http://www.isotc211.org/2005/gmx"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.isotc211.org/2005/gmd http://schemas.opengis.net/csw/2.0.2/profiles/apiso/1.0.0/apiso.xsd">

  <!--TG Recommendation C.1: metadata/2.0/rec/common/fileIdentifier: The metadata record should contain a globally unique and persistent fileIdentifier-->
  <gmd:fileIdentifier>
    <gco:CharacterString>123e4567-e89b-12d3-a456-426655440000</gco:CharacterString>
  </gmd:fileIdentifier>

  <!--TG Requirement C.5: metadata/2.0/req/common/metadata-language-code: The language of the provided metadata content shall be given. It shall be encoded using gmd:MD_Metadata-->
  <gmd:language>
    <gmd:LanguageCode codeList="http://www.loc.gov/standards/iso639-2/" codeListValue="eng"/>
  </gmd:language>
  <gmd:characterSet>
    <gmd:MD_CharacterSetCode codeListValue="utf8" codeList="http://standards.iso.org/iso/19139/resources/gmxCodeLists.xml#MD_CharacterSetCode" codeListValue="dataset"/>
  </gmd:characterSet>

  <!--TG Requirement 1.1: metadata/2.0/req/datasets-and-series/resource-type: The resource type shall be declared as "dataset" or "series" using the first gmd:hierarchyLevel-->
  <gmd:hierarchyLevel>
    <gmd:MD_ScopeCode codeList="http://standards.iso.org/iso/19139/resources/gmxCodeLists.xml#MD_ScopeCode" codeListValue="dataset"/>
  </gmd:hierarchyLevel>

</gmd:MD_Metadata>
```

INSPIRE Reference Validator: Users

From a failed test run to a valid test run

Obtain a fully compliant metadata

Valid metadata test example

Status Passed

Started 28/05/2024 15:43:41 GMT

Duration 4 s

	Total Count	Skipped	Failed	Warnings	Manual
Test suites	5	0	0	0	0
Test cases	12	0	0	0	0
Assertions	43	0	0	0	0

Show

All

Only failed

Only manual

Level of detail

All details

Less information

Simplified

+ Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata records. 3

+ Conformance Class 1: INSPIRE data sets and data set series baseline metadata. 3

+ Conformance Class 2: INSPIRE data sets and data set series interoperability metadata. 4

+ Conformance Class 2b: INSPIRE data sets and data set series metadata for Monitoring 1

+ Conformance Class 8: INSPIRE data sets and data set series linked service metadata 1

Report generated by ETF

https://raw.githubusercontent.com/INSPIRE-MIF/helpdesk-validator/master/training%20material/2024-05-31%20JRC%20Training/Dataset_metadata_2.0_PASSED.xml

INSPIRE Reference Validator: Users

INSPIRE Reference Validator UI and ETF UI

Since the INSPIRE Reference Validator reuses the ETF, the results are the same: ETF integration

Test run on 16:59 - 28.05.2024 with test suite Annex II - Land Cover (LC)

Status	Failed	Total Count	Skipped	Failed	Warnings	Manual
Started	28/05/2024 15:00:00 GMT					
Duration	46 s					
Test suites	12	0	2	0	3	
Test cases	22	0	2	0	4	
Assertions	46	0	2	0	6	

Show: ☒ All ☐ Only failed ☐ Only manual

Level of detail: ☐ All details ☐ Less information ☒ Simplified

- + Conformance class: INSPIRE GML encoding 1
- + Conformance class: Reference systems, General requirements 2
- + Conformance class: Reference systems, Land Cover 1
- + Conformance class: Information accessibility, General requirements 1
- + Conformance class: Information accessibility, Land Cover 2
- + Conformance class: Data consistency, General requirements 2
- + Conformance class: Data consistency, Land Cover 1
- + Conformance class: INSPIRE GML application schemas, General requirements Failed: 1 / 6
- + Conformance class: GML application schemas, Land Cover Failed: 1 / 2
- + Conformance class: Application schema, Land Cover Nomenclature 2
- + Conformance class: Application schema, Land Cover Raster 1
- + Conformance class: Application schema, Land Cover Vector 1

Report generated by ETF

European Commission > INSPIRE > Validator > Test reports

INSPIRE Reference Validator - Test reports

Home Test selection Test reports Get support More on the INSPIRE Reference Validator

Test run on 16:59 - 28.05.2024 with test suite Annex II - Land Cover (LC)

Status	Failed	Total Count	Skipped	Failed	Warnings	Manual
Started	28/05/2024 15:00:00 GMT					
Duration	46 s					
Test suites	12	0	2	0	3	
Test cases	22	0	2	0	4	
Assertions	46	0	2	0	6	

Show: ☒ All ☐ Only failed ☐ Only manual

Level of detail: ☐ All details ☐ Less information ☒ Simplified

- + Conformance class: INSPIRE GML encoding 1
- + Conformance class: Reference systems, General requirements 2
- + Conformance class: Reference systems, Land Cover 1
- + Conformance class: Information accessibility, General requirements 1
- + Conformance class: Information accessibility, Land Cover 2
- + Conformance class: Data consistency, General requirements 2
- + Conformance class: Data consistency, Land Cover 1
- + Conformance class: INSPIRE GML application schemas, General requirements Failed: 1 / 6
- + Conformance class: GML application schemas, Land Cover Failed: 1 / 2
- + Conformance class: Application schema, Land Cover Nomenclature 2
- + Conformance class: Application schema, Land Cover Raster 1
- + Conformance class: Application schema, Land Cover Vector 1

Report generated by ETF

<http://staging-inspire-validator.eu-west-1.elasticbeanstalk.com/validator>
<http://staging-inspire-validator.eu-west-1.elasticbeanstalk.com/etf-webapp>

INSPIRE Reference Validator: Users

Validator UI and ETF API

How is it possible?

Most actions on the INSPIRE Reference Validator actually make API calls to the ETF API and format the JSON result obtained

Test reports

<https://inspire.ec.europa.eu/validator/v2/TestRuns.json?limit=3000>

GET

See report ↕

https://inspire.ec.europa.eu/validator/v2/TestRuns/{testrun_id}.json

GET

Start test >

<https://inspire.ec.europa.eu/validator/v2/TestRuns>

POST

Delete report ✕

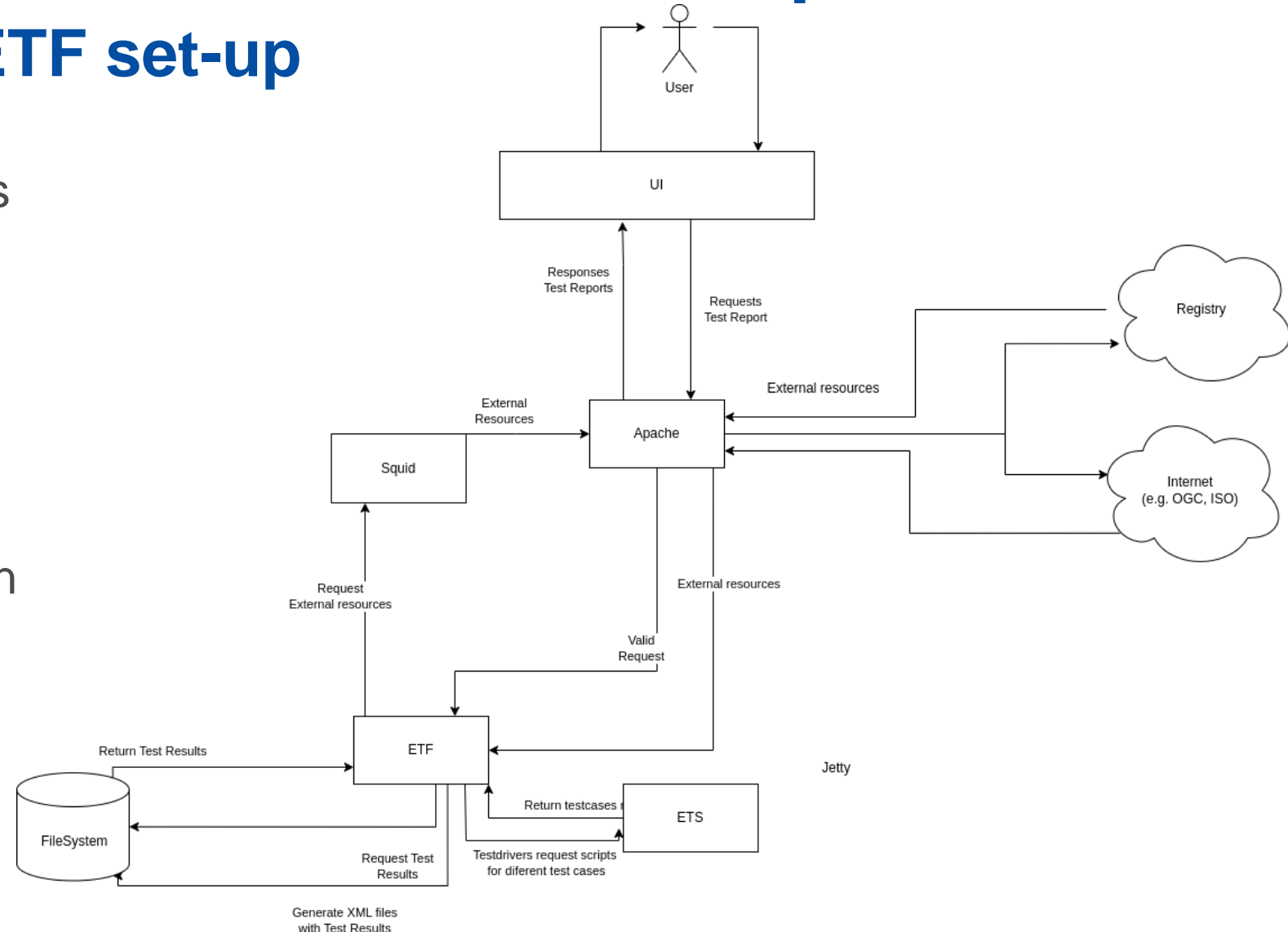
https://inspire.ec.europa.eu/validator/v2/TestRuns/{testrun_id}

DELETE

INSPIRE Reference Validator: Developers

Understanding ETF set-up

- User Interface for validations
- ETF API
- ETS repository
- TEAM Engine Integration
- INSPIRE Registry integration
- Apache as a proxy pass
- Squid as a reverse proxy

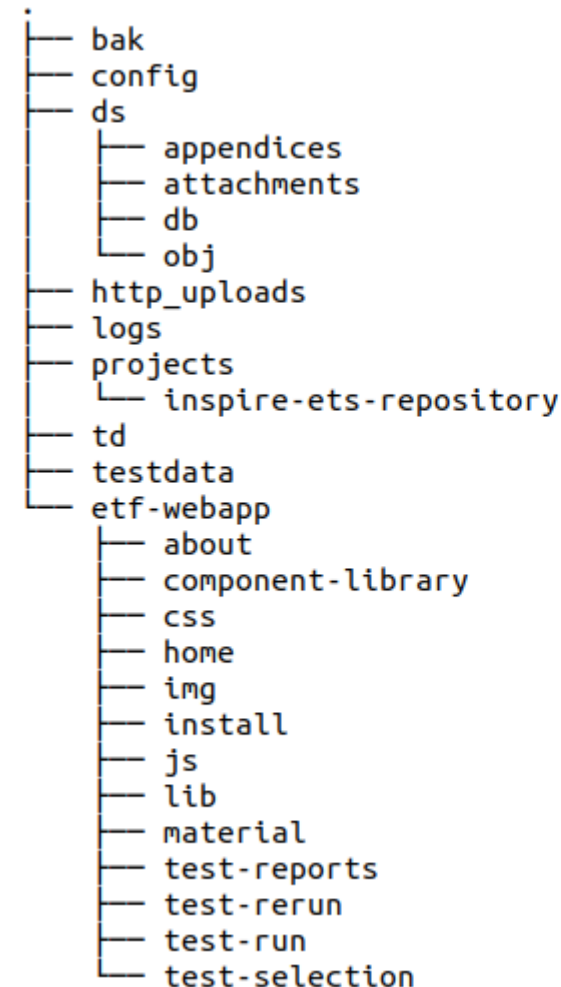


INSPIRE Reference Validator: Developers

Understanding ETF set-up

Folder structure:

- etf-webapp
 - test-reports
 - test-rerun
 - test-run
- testdata
- ...





INSPIRE Reference Validator: Developers

Deploying ETF using Jetty

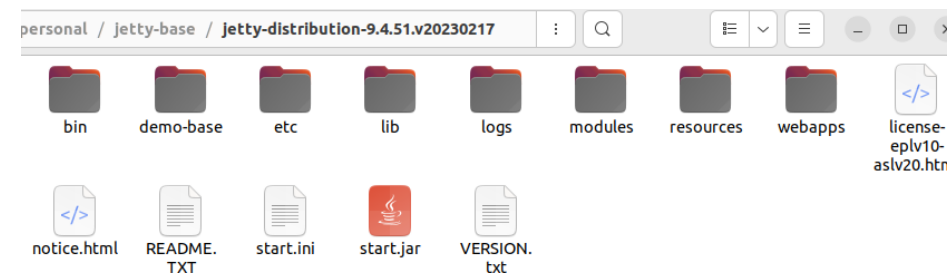
- We first will download Jetty server version 9 ZIP from the jetty url

Release			Metadata	Resources
12.0.9	.zip	.tgz	(info)	Release Notes / API Docs
11.0.20	.zip	.tgz	(info)	Release Notes / API Docs
10.0.20	.zip	.tgz	(info)	Release Notes / API Docs
9.4.54.v20240208	.zip	.tgz	(info)	Release Notes / API Docs

- We will need Java 8 for this Jetty version, which we can get from oracle url

Linux x64	170.15 MB	 jdk-8u202-linux-x64.rpm
Linux x64	185.05 MB	 jdk-8u202-linux-x64.tar.gz

- Once we have downloaded the Jetty ZIP we extract it wherever we want



- Now we download the ETF WAR from the version we want directly from the GitHub repository



<https://eclipse.dev/jetty/download.php>

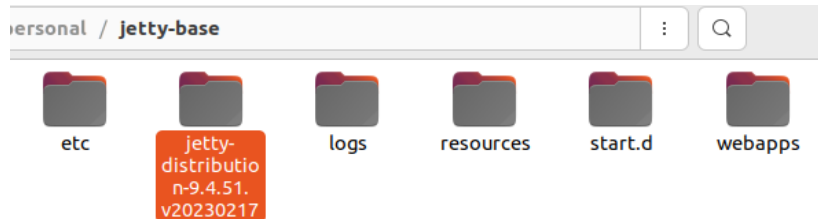
<https://www.oracle.com/es/java/technologies/javase/javase8-archive-downloads.html>

<https://github.com/etf-validator/etf-webapp/releases/tag/2.1.0>

INSPIRE Reference Validator: Developers

Deploying ETF using Jetty

- We have to copy the demo-base folder of the Jetty on our home directory and rename it to jetty-base, we also have to move the jetty folder we have inside it:



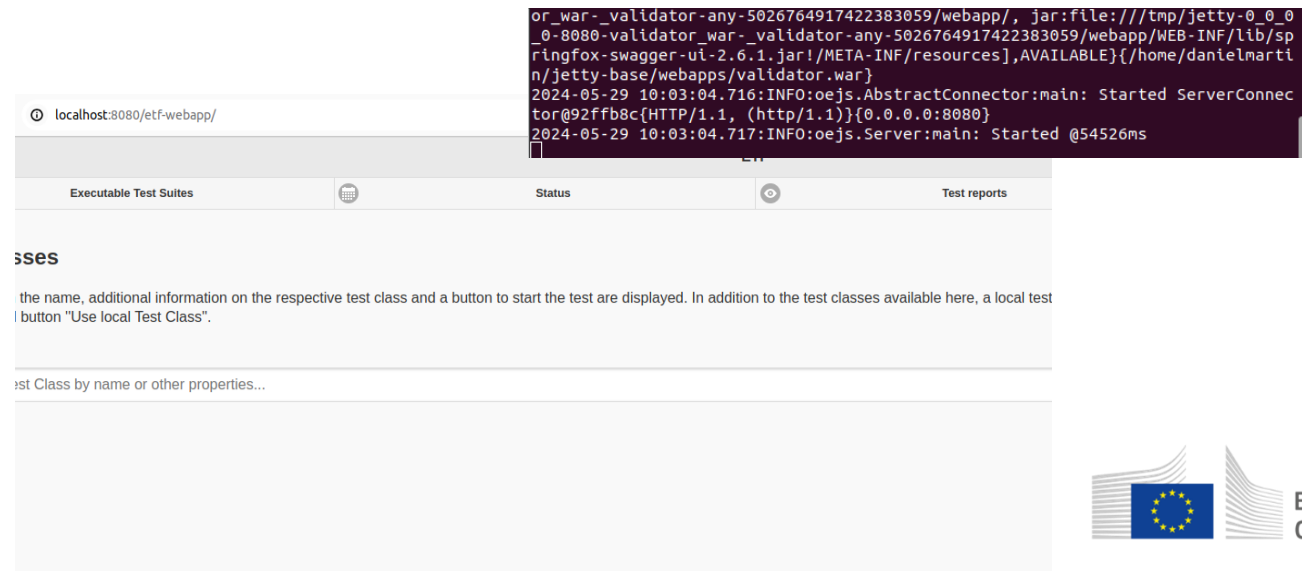
- Once we have downloaded the WAR we will add it to the folder webapps inside the Jetty-base:



- Then we go to the Jetty-base and we use the following command:

```
java -jar -
agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=:5005
./jetty-distribution-9.4.54.v20240208/start.jar --run
```

- This will start the jetty server with the ETF, we will find it in <http://localhost:8080/etf-webapp> :



INSPIRE Reference Validator: Developers

Building ETF from source code

- We first will clone the repository of the etf

```
git clone https://github.com/etf-validator/etf-webapp.git
```

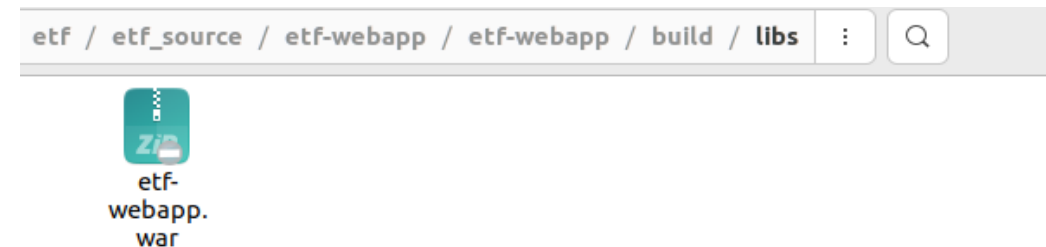
- We then select the branch or tag we want to build

```
cd etf-webapp  
git checkout <branch-tag>
```

- We build the project with these commands

```
./gradlew clean  
./gradlew spotlessApply  
./gradlew build install -x test
```

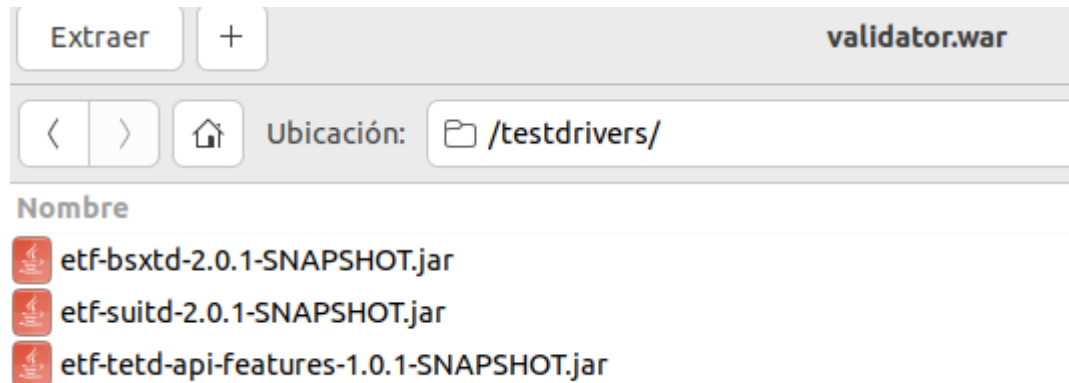
- Once the build is complete, you will find the ETF web-app .war file inside `etf-webapp/build/libs/etf-webapp.war`, which can be deployed on an application server.



INSPIRE Reference Validator: Developers

Building ETF from source code

- To install any of the testdrivers, we will build them using the same method and then we will move the generated .jar inside the war creating a testdrivers folder inside it.

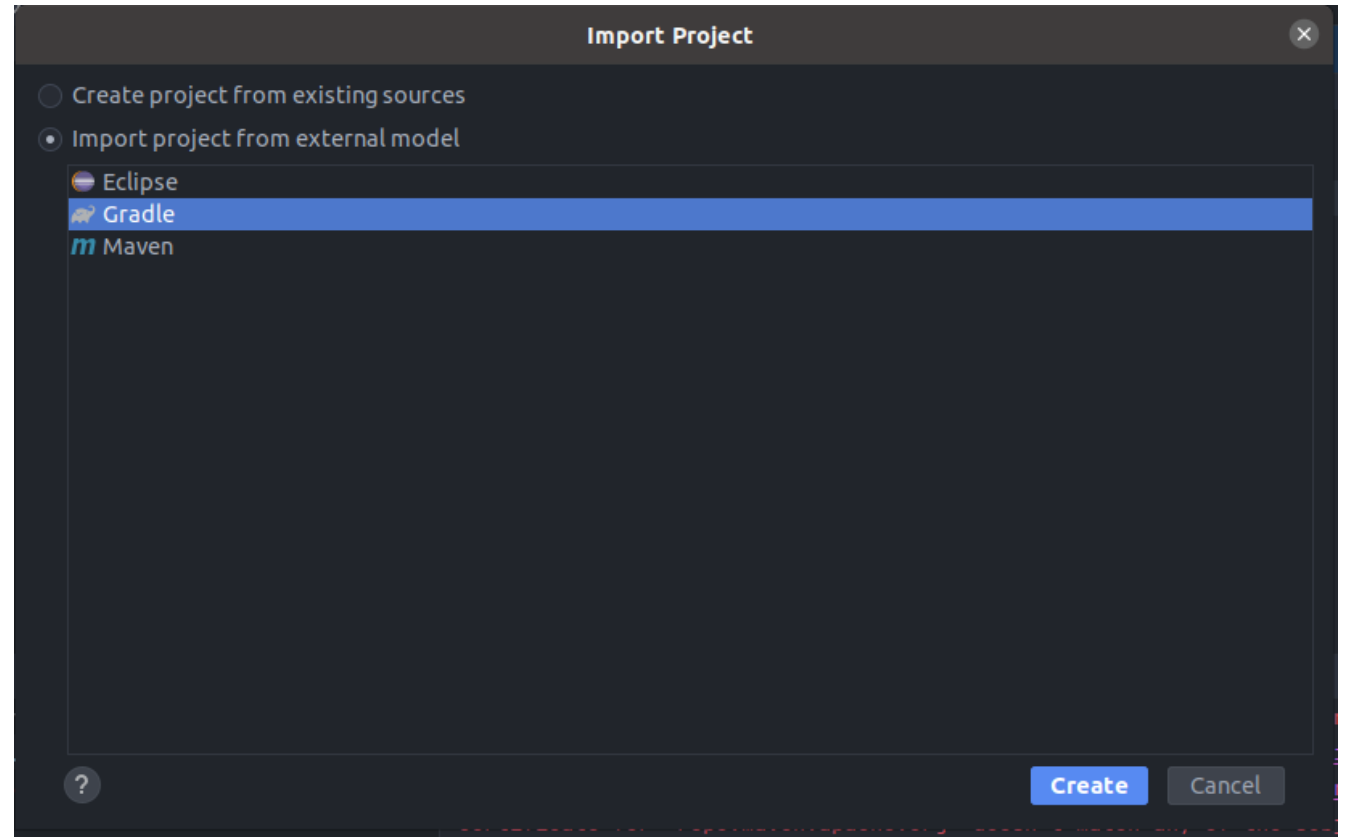


- We used the following requirements :
 - Java (We are using openjdk 11.0.22)
 - git (2.34.1)
 - Gradle 4.4.1

INSPIRE Reference Validator: Developers

Creating a workspace in IntelliJ

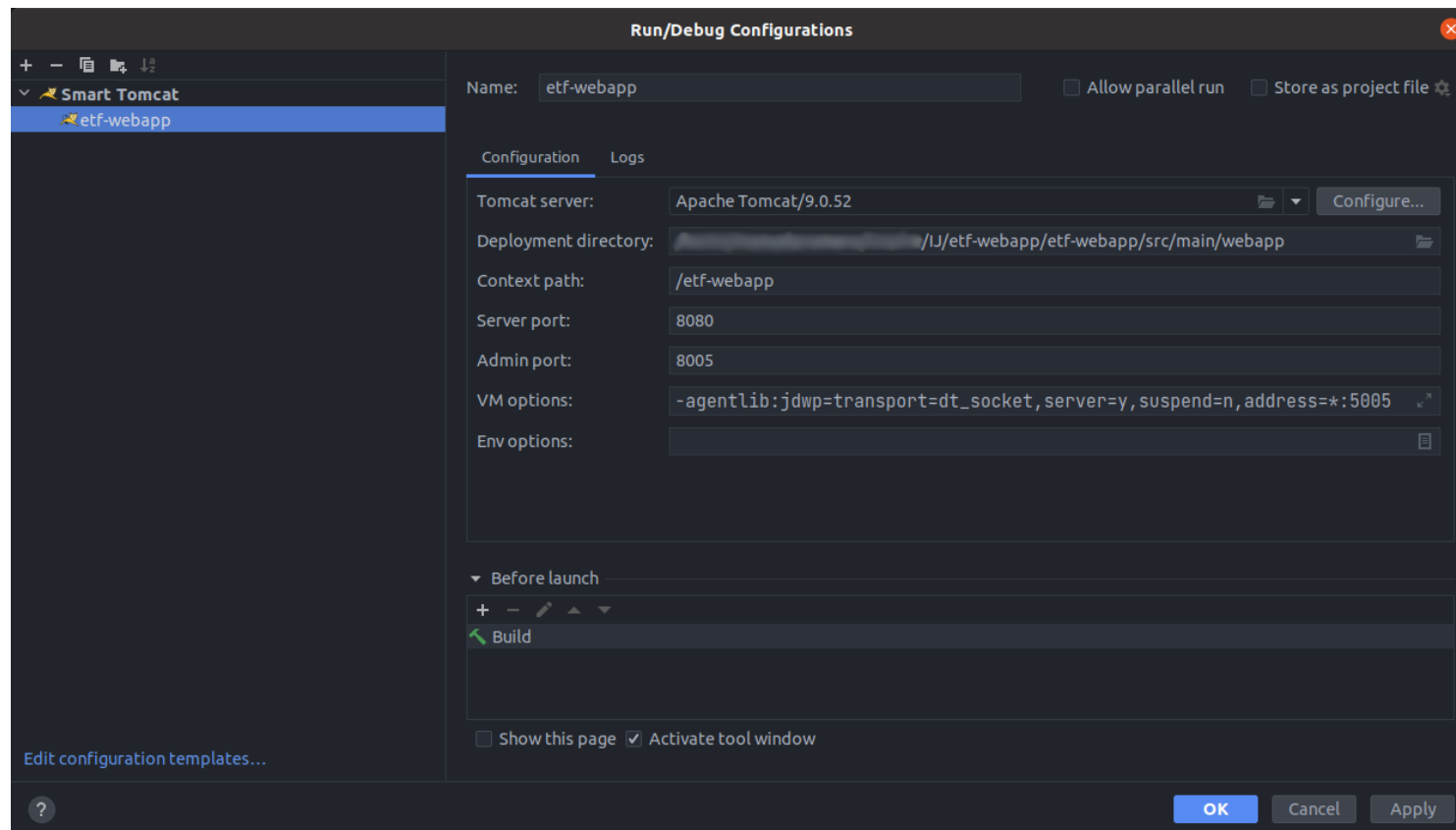
- Install Java and Tomcat plugins
- Create a gradle project



INSPIRE Reference Validator: Developers

Creating a workspace in IntelliJ

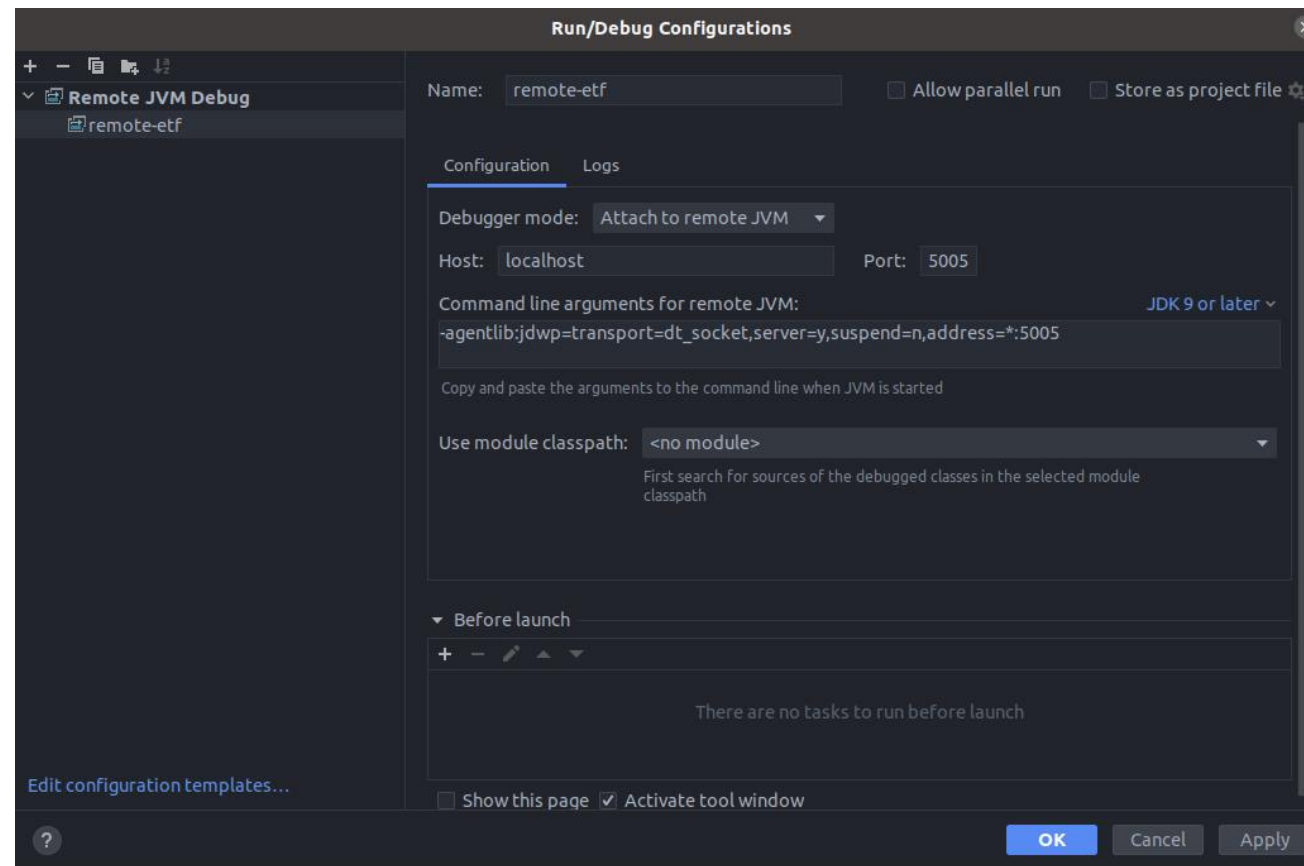
- Create a run configuration file for the different modules



INSPIRE Reference Validator: Developers

Creating a workspace in IntelliJ


- Debug by creating external calls to the specific component



INSPIRE Reference Validator: Developers

Using Validator-ETF Swagger UI

- Services Status
- Service Capabilities
- Manage Test Objects
- Manage Test Runs
- Test Run results

 **swagger**
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](#)

1. Service Status : Monitor service workload and health	Show/Hide	List Operations	Expand Operations
2. Service Capabilities : Retrieve test framework metadata	Show/Hide	List Operations	Expand Operations
3. Manage Test Objects : Define Test Objects and upload test data	Show/Hide	List Operations	Expand Operations
4. Manage Test Runs : Start and control test runs	Show/Hide	List Operations	Expand Operations
5. Test Run Results : Retrieve test results	Show/Hide	List Operations	Expand Operations

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

INSPIRE Reference Validator: Developers

Using Validator-ETF Swagger UI

- Upload a resource

POST /v2/TestObjects Upload a file for the Test Object using a MULTIPART upload request

Implementation Notes

On success the service will internally create a TEMPORARY new Test Object and return it's ID which afterwards can be used to start a new Test Run. If the Test Object ID is not used within 5 minutes, the Test Object and all uploaded data will be deleted automatically. PLEASE NOTE: This interface will create a TEMPORARY Test Object that will not be persisted as long as it is not used in a Test Run. A TEMPORARY Test Object can not be retrieved or deleted but can only be referenced from a 'Test Run Request' to start a new Test Run. The property 'data.downloadable' of a TEMPORARY Test Object is always set to true. Also note that the Swagger UI does only allow single file uploads in contrast to the API which allows multi file uploads.

Response Class (Status 200)

File uploaded and temporary Test Object created

Model

Example Value

```
{
  "files": [
    {
      "name": "file.xml",
      "size": "2048",
      "type": "text/xml"
    }
  ],
  "testObject": {
    "id": "EID12bb90ca-ee02-4f79-9dd9-63dff6d8e150",
  }
}
```

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

INSPIRE Reference Validator: Developers

Using Validator-ETF Swagger UI

- Start a test run

POST /v2/TestRuns Start a new Test Run

Implementation Notes

Start a new Test Run by specifying one or multiple Executable Test Suites that shall be used to test one Test Object with specified test parameters. If data for a Test Object need to be uploaded, the Test Object POST interface needs to be used to create a new temporary Test Object. The temporary Test Object or any other existing Test Object can be referenced by setting exclusively the 'id' in the StartTestRunRequest's 'testObject' property. If data do not need to be uploaded or a web service is tested, a temporary Test Object can be created directly with this interface, by defining at least the 'resources' property of the 'testObject' but omit except the 'id' property.

Example for starting a Test Run for a service Test:

```
{
  "label": "Test run on 15:00 - 01.01.2017 with Conformance class Conformance Class: Download Service - Pre-defined WFS",
  "executableTestSuiteIds": ["EID174edf55-699b-446c-968c-1892a4d8d5bd"],
  "arguments": {},
  "testObject": {
    "resources": {
      "serviceEndpoint": "http://example.com/service?request=GetCapabilities&service=WFS"
    }
  }
}
```

Example for starting a Test Run for a file-based Test, using a temporary Test Object:

```
{
  "label": "Test run on 15:00 - 01.01.2017 with Conformance class INSPIRE Profile based on EN ISO 19115 and EN ISO 19119",
  "executableTestSuiteIds": ["EIDec7323d5-d8f0-4cfe-b23a-b826df86d58c"],
  "arguments": {
    "files_to_test": ".*",
    "tests_to_execute": ".*"
  },
  "testObject": {
    "id": "b502260f-1054-432e-8cd5-4a61302dfdba"
  }
}
```

Where "EIDb502260f-1054-432e-8cd5-4a61302dfdba" is the ID of the previous created temporary Test Object.

INSPIRE Reference Validator: Developers

Using Validator-ETF Swagger UI

- Check test run status

GET `/v2/TestRuns/{id}/progress` Get the Test Run progress as JSON

Implementation Notes
Retrieve one Test Run status including log messages, the estimated total number of Test Steps and the number of already executed Test Steps

Response Class (Status 200)
Task progress returned

Model	Example Value
	<pre>{ "log": "[\"Test Run started\", \"Assertion X failed\"]", "max": "103", "val": "39" }</pre>

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

INSPIRE Reference Validator: Developers

Using Validator-ETF Swagger UI

- Get test results

GET
/v2/TestTaskResults/{id}.json
Get the result from a single Test Task within a Test Run as JSON

Implementation Notes

Transforms the result from a single Test Task to JSON. The Test Task model is described in the [XML schema documentation](#). Note: a Test Run consists of one or multiple Test Task Results. A Test Task Result represents the result of the execution of one single Test Suite. Use the Test Run interface to get all results of a Test Run and the Test Task Result interfaces to get only one single result. Items are returned in an [ETF item collection](#).

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	(required)	Test Task ID. The ETF ID is an 36 characters long hexadecimal Universally Unique Identifier prefixed with 'EID', e.g. EID12bb90ca-ee02-4f79-9dd9-63dff6d8e150	path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
202	Test Task exists		
404	Test Task does not exist		

Try it out!

INSPIRE Reference Validator: Developers

API Test Run example

POST https://inspire.ec.europa.eu/validator/v2/TestRuns

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

```

1 {
2   "label": "API Test Run Example",
3   "executableTestSuiteIds": ["EIDe4a95862-9cc9-436b-9fdd-a0115d342350", "EID2be1480a-fe42-40b2-9420-eb0e69385c80", "EID0b86f7a3-2947-4841-823d-6a00d8e06d70",
4   "EIDa593a7ad-42d9-46d0-985d-9dff3e684428", "EID7ccea68-e575-4429-9959-1b6b3d201b6d"],
5   "arguments": {},
6   "testObject": {},
7   "resources": {
8     "data": "https://raw.githubusercontent.com/INSPIRE-MIF/helpdesk-validator/master/examples/Dataset_metadata_2.0_example.xml"
9   }
10 }
11 }

```

Body Cookies Headers (12) Test Results

Status: 201 Created Time: 6.04 s Size: 404.63 KB Save Response

Pretty Raw Preview Visualize JSON

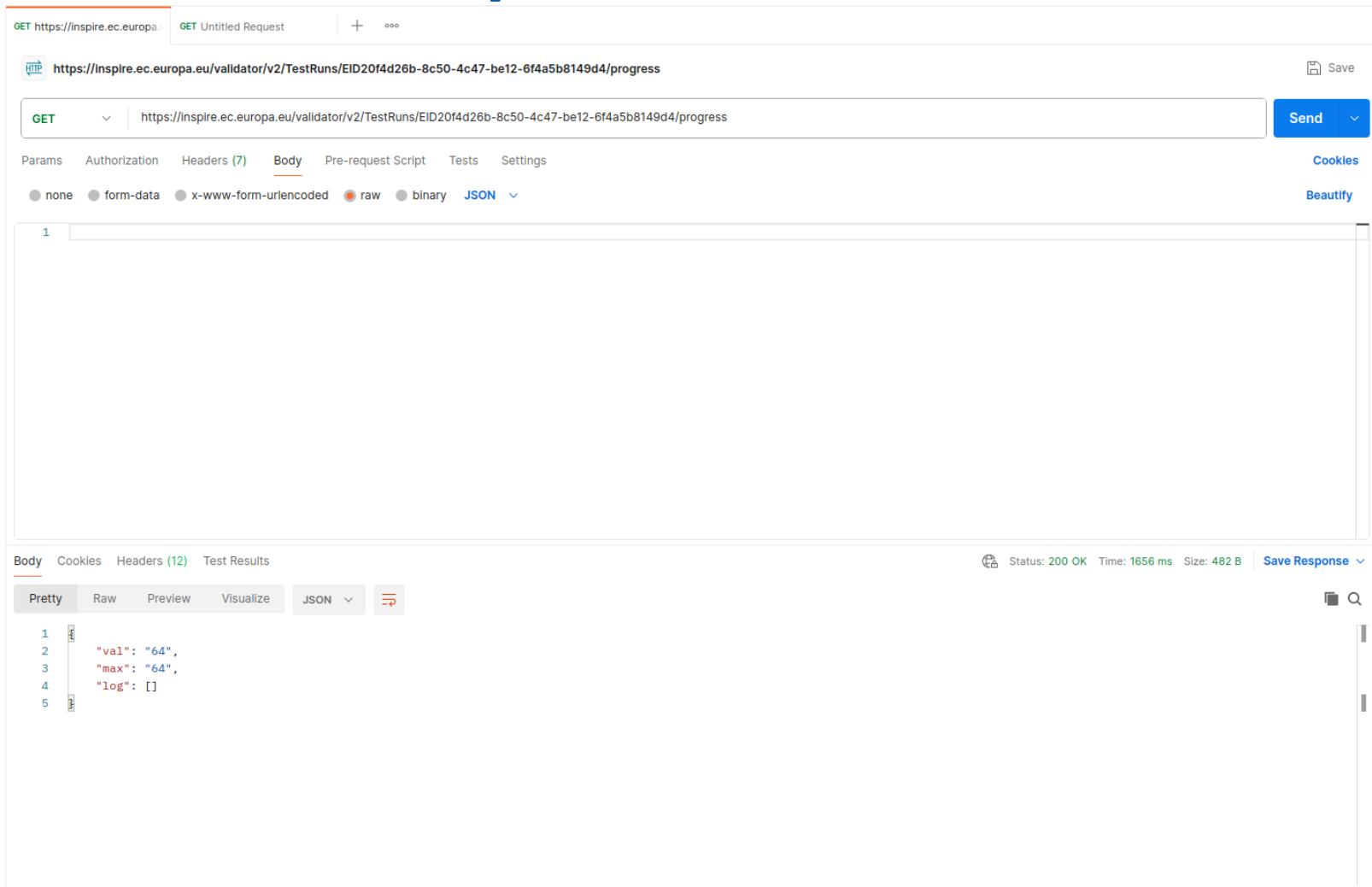
```

1 {
2   "EtfItemCollection": {
3     "version": 2.0,
4     "returnedItems": 1,
5     "ref": "https://yzyiqiakm4.execute-api.eu-west-1.amazonaws.com/validator/v2/TestRuns/fe48d8e1-7b0e-4107-9e7e-c4f256a1d742.json",
6     "testRuns": {
7       "TestRun": {
8         "id": "EIDfe48d8e1-7b0e-4107-9e7e-c4f256a1d742",
9         "status": "UNDEFINED",
10        "label": "API Test Run Example",
11        "defaultLang": "en",
12        "startTimestamp": "2024-05-29T07:03:15.913Z",
13        "testTasks": {
14          "TestTask": [
15            {

```

INSPIRE Reference Validator: Developers

API Test Run example



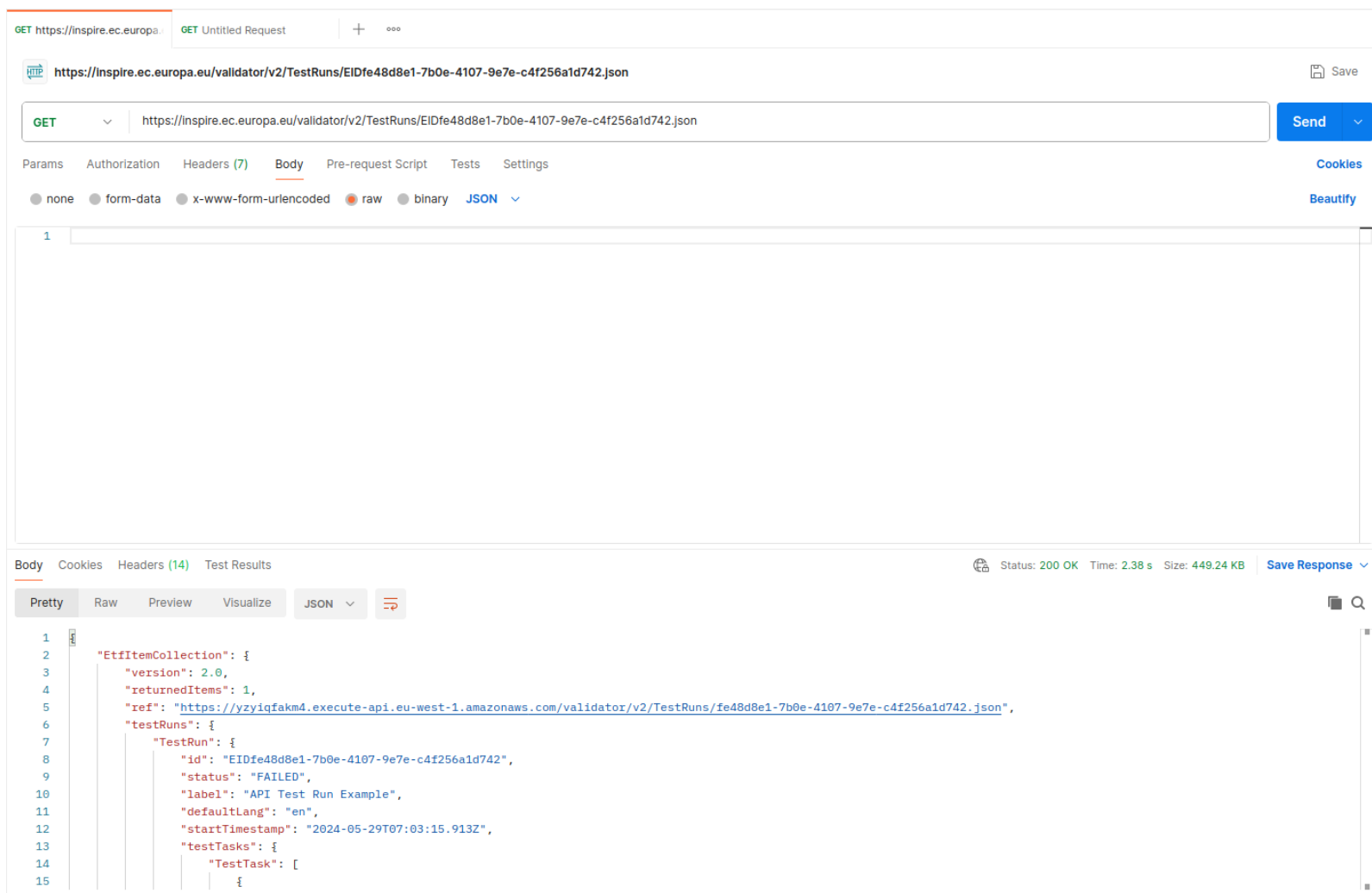
The screenshot displays a REST client interface with the following details:

- Request Method:** GET
- URL:** `https://inspire.ec.europa.eu/validator/v2/TestRuns/EID20f4d26b-8c50-4c47-be12-6f4a5b8149d4/progress`
- Response Status:** 200 OK
- Response Time:** 1656 ms
- Response Size:** 482 B
- Response Body (JSON):**

```
{  "val": "64",  "max": "64",  "log": []}
```

INSPIRE Reference Validator: Developers

API Test Run example



GET https://inspire.ec.europa.eu/validator/v2/TestRuns/EIDfe48d8e1-7b0e-4107-9e7e-c4f256a1d742.json

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary JSON

1

Body Cookies Headers (14) Test Results

Status: 200 OK Time: 2.38 s Size: 449.24 KB Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "EtfItemCollection": {
3      "version": 2.0,
4      "returnedItems": 1,
5      "ref": "https://yzyiqfakm4.execute-api.eu-west-1.amazonaws.com/validator/v2/TestRuns/fe48d8e1-7b0e-4107-9e7e-c4f256a1d742.json",
6      "testRuns": {
7        "TestRun": {
8          "id": "EIDfe48d8e1-7b0e-4107-9e7e-c4f256a1d742",
9          "status": "FAILED",
10         "label": "API Test Run Example",
11         "defaultLang": "en",
12         "startTimestamp": "2024-05-29T07:03:15.913Z",
13         "testTasks": {
14           "TestTask": [
15             {

```

INSPIRE Reference Validator: Developers

API Gateway for Production instance

- Local deployment is recommended
- API-Key with a quota can be requested in case it is needed
- 200 requests/week (refilled every Sunday)
- More information in the INSPIRE Reference Validator helpdesk

Upcoming change: API Gateway to access the Validator #594



MarcoMinghini opened this issue on Aug 26, 2021 · 16 comments



MarcoMinghini commented on Aug 26, 2021 · edited by dperezBM ▾

Member ...

Dear all,

this message is to inform you about a change we will introduce in the INSPIRE Reference Validator in the next release (v.2021.3) planned for mid-September (see the [release plan](#)).

Starting with this release, access to the Validator will be managed and controlled by an API Gateway. While this will have no impact on users using the Validator from the INSPIRE central User Interface, users using the Validator API from external clients will need a token to access the API.

More information on the access conditions as well as how organisations can request a token will follow here.

<https://github.com/INSPIRE-MIF/helpdesk-validator/issues/594>

INSPIRE Reference Validator: Developers

API Gateway for Production instance

- Last 10 days usage
- Remaining quota

statusCode:	200
body:	
items:	
0:	
date:	"2023-06-27"
uses:	34
remaining_uses:	0
1:	
date:	"2023-06-28"
uses:	0
remaining_uses:	0
2:	
date:	"2023-06-29"
uses:	0
remaining_uses:	0
3:	
date:	"2023-06-30"
uses:	0
remaining_uses:	0
4:	
date:	"2023-07-01"
uses:	0
remaining_uses:	0

token refill on Sunday

5:	
date:	"2023-07-02"
uses:	0
remaining_uses:	200
6:	
date:	"2023-07-03"
uses:	0
remaining_uses:	200
7:	
date:	"2023-07-04"
uses:	176
remaining_uses:	24
8:	
date:	"2023-07-05"
uses:	0
remaining_uses:	24
9:	
date:	"2023-07-06"
uses:	12
remaining_uses:	12

INSPIRE Reference Validator: Developers

API Gateway for Production instance

- The production instance of the INSPIRE Reference Validator is not meant to be used for massive/large validations, which may result into software unavailability for the whole community:
 - ‘heavy’ services serving several and/or large datasets
 - large GML datasets – current limit for file upload: 50 MB
 - massive validations through the API – limited through an API Gateway
- For such validation needs:
 - please setup your own instance of the INSPIRE Reference Validator (e.g. using Docker)
 - ask for we offer help with the deployment

<https://github.com/INSPIRE-MIF/helpdesk-validator/issues/594>

INSPIRE Reference Validator: Developers

Developing an ETS

- ATS: Abstract Test Suite
- ETS: Executable Test Suite
 - Xquery
 - SOAPUI + Groovy
- From ATS to ETS

The following check is performed for every feature in the dataset, for the 'narrower' codelist:

- Check that all the [currentUse](#) elements has a xlink:href attribute pointing to a [pre-defined value](#). If the check fails a manual check will be required asking to review the codelist definition in order to verify that any extensions do not overlap with the codelists that are defined in Annexes II, III and IV of the Implementing Rule. In particular, for the 'narrower' codelists the extended values shall refer to a parent value defined by the Implementing Rule. If the check fails report [reviewCodeListValue](#).

Pre-defined values for xlink:href attribute of [currentUse](#) element are available in the INSPIRE Registry

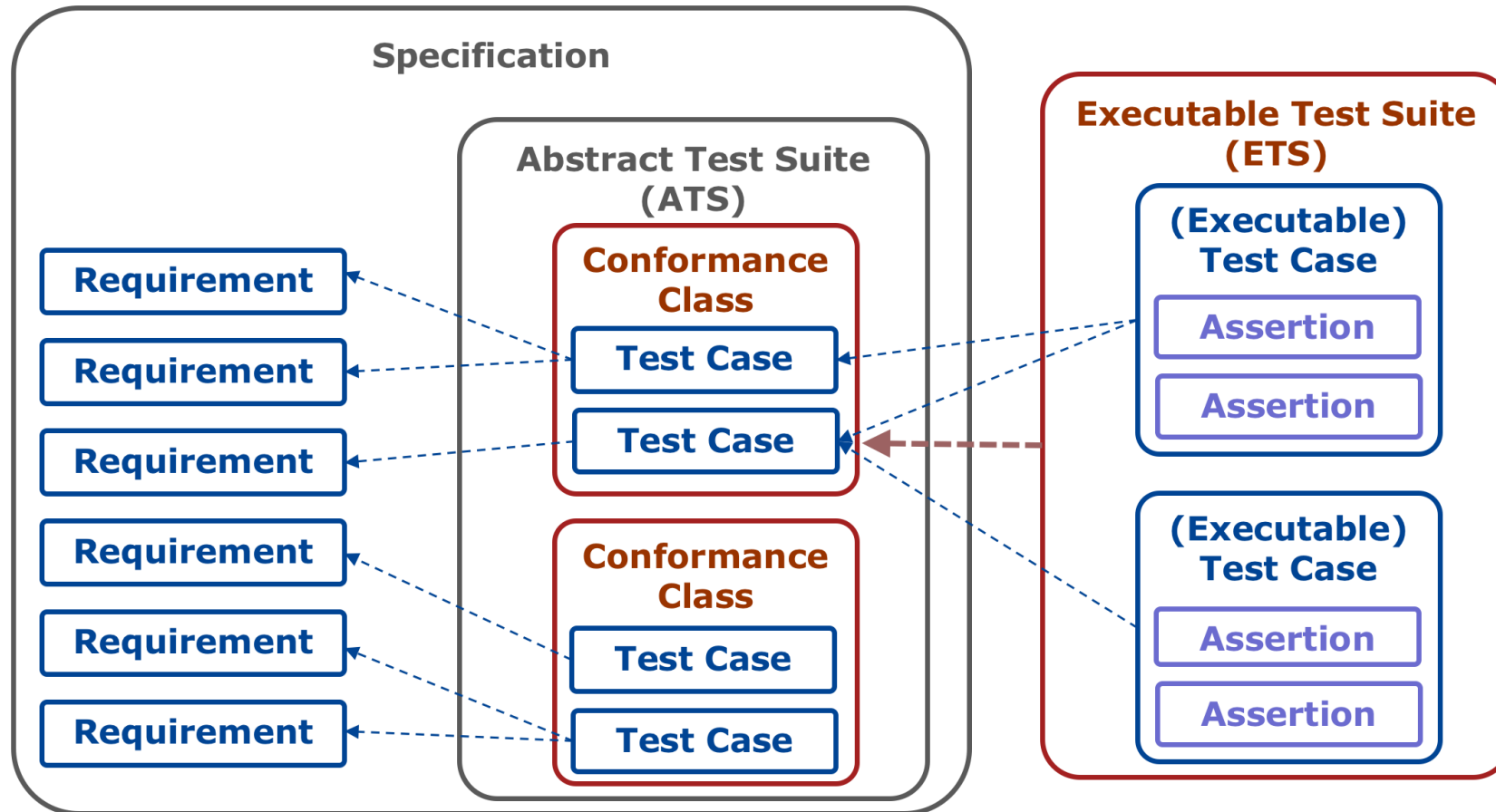
CurrentUseValue: <http://inspire.ec.europa.eu/codelist/CurrentUseValue>

- ATS: <http://inspire.ec.europa.eu/id/ats/data-bu/master/bu-as/code-list-values>
- ETS: TestCase for currentUse

<https://github.com/inspire-eu-validation/ets-repository/blob/master/data-bu/bu-3d-as/specific-schemas/ets-bu-as-bsxets.xml>

INSPIRE Reference Validator: Developers

Developing an ETS



INSPIRE Reference Validator: Developers

Developing an ETS

- ATS defines the ETS implementation

reviewCodeListValue	XML document '{filename}', {featureType} '{gmlid}': The property '{property}' has a value '{value}' that is not one of the pre-defined values in the INSPIRE data specification. The same value is used by {count} other features in the dataset, too. Extensions to the pre-defined values are allowed, but must not overlap with one of the pre-defined values and be consistent with the specification of the property. Please review the definition of the value. Please note that the URIs of all code list values from the INSPIRE Registry shall be referenced using the http protocol.
---------------------	--

Reference(s):

- [TG DS Template](#) IR requirement Article 4 (1)
- [TG DS Template](#) IR requirement Article 4 (3)
- [TG DS Template](#) IR requirement Article 6 (1)
- [TG DS Template](#) IR requirement Article 6 (2)
- [TG DS Template](#) IR requirement Article 6 (3)
- [TG DS Template](#) IR requirement Article 6 (4)

currentUse	//schema-element(bu-core2d:Building)/bu-base:currentUse/bu-base:CurrentUse/bu-base:currentUse/@xlink:href //schema-element(bu-core2d:BuildingPart)/bu-base:currentUse/bu-base:CurrentUse/bu-base:currentUse/@xlink:href //schema-element(bu-core2d:Building)/bu-base:parts/bu-core2d:BuildingPart/bu-base:currentUse/bu-base:CurrentUse/bu-base:currentUse/@xlink:href //schema-element(bu-core3d:Building)/bu-base:currentUse/bu-base:CurrentUse/bu-base:currentUse/@xlink:href //schema-element(bu-core3d:BuildingPart)/bu-base:currentUse/bu-base:CurrentUse/bu-base:currentUse/@xlink:href //schema-element(bu-core3d:Building)/bu-base:parts/bu-core3d:BuildingPart/bu-base:currentUse/bu-base:CurrentUse/bu-base:currentUse/@xlink:href	1 (0..* for the parent element)	No
------------	--	---------------------------------	----

INSPIRE Reference Validator: Developers

Developing an ETS

- Include description tag containing
 - Description of the rule

```
<TestAssertion id="EID18e34ba6-38da-4a3b-8a37-903f9fe4533a">
  <label>bu-as.a.11: currentUse</label>
  <description><![CDATA[Check that all the <a href= 'http://inspire.ec.europa.eu/id/ats/
data-bu/master/bu-as/code-list-values#currentUse'>currentUse</a> elements has a
xlink:href attribute pointing to a <a href= 'http://inspire.ec.europa.eu/id/ats/data-bu/
master/bu-as/code-list-values#predefinedValue1'>pre-defined value</a>. If the check fails
a manual check will be required asking to review the codelist definition in order to
verify that any extensions do not overlap with the codelists that are defined in Annexes
II, III and IV of the Implementing Rule. In particular, for the 'narrower' codelists the
extended values shall refer to a parent value defined by the Implementing Rule. If the
check fails report reviewCodeListValue.<br><br>]]></description>
```

- Relevant requirements

Reference(s):

- [TG DS Template](#) IR requirement Article 4 (1)
- [TG DS Template](#) IR requirement Article 4 (3)
- [TG DS Template](#) IR requirement Article 6 (1)
- [TG DS Template](#) IR requirement Article 6 (2)
- [TG DS Template](#) IR requirement Article 6 (3)
- [TG DS Template](#) IR requirement Article 6 (4)

- Source: ATS and data specification link

```
Relevant requirements:
<ul>
<li>IR Requirement Article 4 (1): For the exchange and classification of spatial objects from data sets meeting the
conditions laid down in Article 4 of Directive 2007/2/EC, Member States shall use the spatial object types and
associated data types, enumerations and code lists that are defined in Annexes II, III and IV for the themes the data sets
relate to.</li>
<li>IR Requirement Article 4 (3): The enumerations and code lists used in attributes or association roles of spatial object
types or data types shall comply with the definitions and include the values set out in Annex II. The enumeration and code
list values are uniquely identified by language-neutral mnemonic codes for computers. The values may also include a
language-specific name to be used for human interaction.</li>
<li>IR Requirement Article 6 (1): Code lists shall be of one of the following types, as specified in the Annexes:<ol
style="list-style-type:lower-alpha"><li>code lists whose allowed values comprise only the values specified in this
Regulation;</li>
<li>code lists whose allowed values comprise the values specified in this Regulation and narrower values defined
by data providers;</li>
<li>code lists whose allowed values comprise the values specified in this Regulation and additional values at any
level defined by data providers;</li>
<li>code lists, whose allowed values comprise any values defined by data providers.</li>
</ol>For the purposes of points (b), (c) and (d), in addition to the allowed values, data providers may use the values
specified in the relevant INSPIRE Technical Guidance document available on the INSPIRE web site of the Joint Research Centre.
</li>
<li>IR Requirement Article 6 (2): Code lists may be hierarchical. Values of hierarchical code lists may have a more generic
parent value. Where the valid values of a hierarchical code list are specified in a table in this Regulation,
the parent values are listed in the last column.</li>
<li>IR Requirement Article 6 (3): Where, for an attribute whose type is a code list as referred to in points (
b), (c) or (d) of paragraph 1, a data provider provides a value that is not specified in this Regulation, that value and
its definition shall be made available in a register.</li>
<li>IR Requirement Article 6 (4): Attributes or association roles of spatial object types or data types whose type is a code
list may only take values that are allowed according to the specification of the code list.</li>
</ul>
```

Source: [Abstract Test Case 'Code list values'](http://inspire.ec.europa.eu/id/ats/data-bu/master/bu-as/code-list-values), [INSPIRE Data Specification on Buildings, A.1.6](http://inspire.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_BU_v3.0.pdf)]]></description>

INSPIRE Reference Validator: Developers

Developing an ETS

- expression tag: Xquery code for the test
- testItemType tag: type of textATS: Abstract Test Suite

```
<testItemType ref="EIDf0edc596-49d2-48d6-a1a1-1ac581dcde0a"/>
```

- translationTemplates: to introduce error messages

```
<translationTemplates>  
  <translationTemplate ref="TR.missingElementBuildings"/>  
  <translationTemplate ref="TR.wrongMultiplicityBuildings"/>  
  <translationTemplate ref="TR.missingAttributeBuildings"/>  
  <translationTemplate ref="TR.manual.reviewCodeListValue"/>  
</translationTemplates>
```

INSPIRE Reference Validator: Developers

Developing an ETS

- expression tag: Xquery code for the test
- testItemType tag: type of textATS: Abstract Test Suite

```
<testItemType ref="EIDf0edc596-49d2-48d6-a1a1-1ac581dcde0a"/>
```

- translationTemplates: to introduce error messages

```
<translationTemplates>  
  <translationTemplate ref="TR.missingElementBuildings"/>  
  <translationTemplate ref="TR.wrongMultiplicityBuildings"/>  
  <translationTemplate ref="TR.missingAttributeBuildings"/>  
  <translationTemplate ref="TR.manual.reviewCodeListValue"/>  
</translationTemplates>
```

INSPIRE Reference Validator: Admins

Deploying Validator with Docker

Once we have installed Docker on our system, we can run the INSPIRE Validator with one single command:

```
docker run --name inspire-validator -d -p 8090:8090 -v ~/etf:/etf docker.pkg.github.com/inspire-mif/helpdesk-validator/inspire-validator:2024.1
```

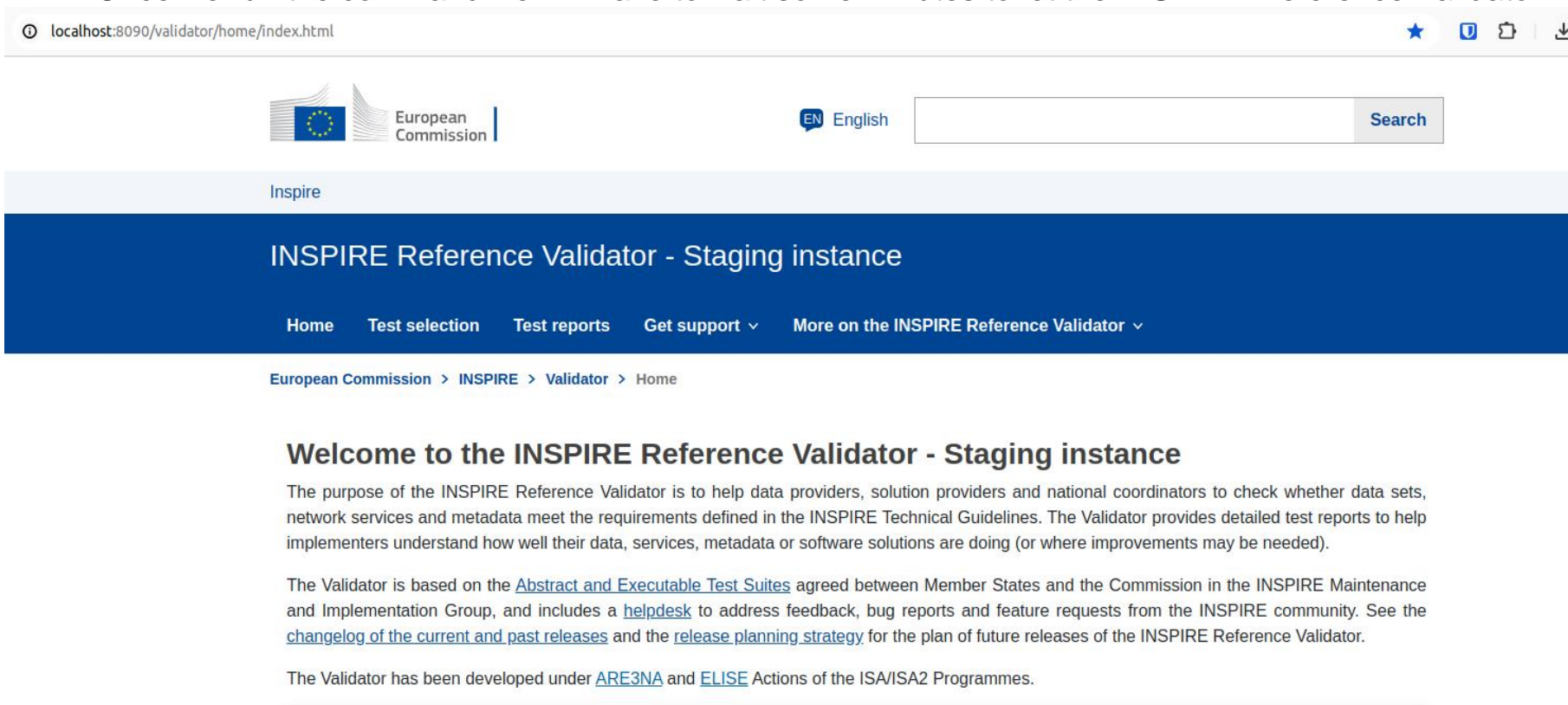
Lets understand this command:

- Docker run: We tell Docker to start a new container
- – name: We declare a name for the container, accessing further commands easily if needed
- -d: Detached mode, meaning to run the container on the background (Don't attach the terminal to it)
- -p 8090:8090: Ports redirecting, here we are redirecting the port on the right (Container port 8090) to the on the left (our system port 8090) making our port 8090 to be the same as the container. The Validator is hosted on port 8090 of the container, so our 8090 will too (We can use mostly any other port on here)
- -v: This creates a volume (A shared folder between the container and our system, so files can be accessed easier), the folder on the left (~/.etf of our system) will be the same as the one on the right (/etf of the container). All files modified on one will also be on the other (The folder we choose on the left one can be anyone we want)
- docker.pkg...: This is the url to the Docker Image (We will discuss this next) of the last Validator version. We can also use any other version available (2023.0, 2024.0.1, ...)

INSPIRE Reference Validator: Admins

Deploying Validator with Docker

Once we run the command we will have to wait some minutes to let the INSPIRE Reference Validator

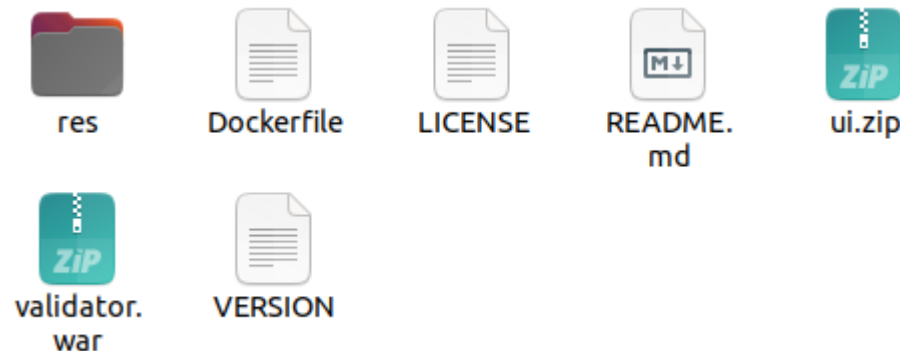


The screenshot shows a web browser window with the address bar displaying 'localhost:8090/validator/home/index.html'. The page header includes the European Commission logo, a language selector set to 'English', and a search bar. Below the header, a blue banner reads 'INSPIRE Reference Validator - Staging instance'. A navigation menu contains links for 'Home', 'Test selection', 'Test reports', 'Get support', and 'More on the INSPIRE Reference Validator'. A breadcrumb trail shows 'European Commission > INSPIRE > Validator > Home'. The main content area features a heading 'Welcome to the INSPIRE Reference Validator - Staging instance' followed by a paragraph explaining the validator's purpose: to help data providers, solution providers, and national coordinators check if data sets, network services, and metadata meet INSPIRE Technical Guidelines. It also mentions that the validator provides detailed test reports. Below this, another paragraph states that the validator is based on the 'Abstract and Executable Test Suites' agreed between Member States and the Commission, and includes a 'helpdesk' for feedback and feature requests. It also refers to a 'changelog of the current and past releases' and a 'release planning strategy'. The final paragraph notes that the validator was developed under 'ARE3NA' and 'ELISE' Actions of the ISA/ISA2 Programmes.

INSPIRE Reference Validator: Admins

Understanding Validator Docker ZIP

The Validator Docker image is the result of building the inspire-validator-xxx.zip we can find on any Validator tag on our GitHub repository (At the bottom of each tag)

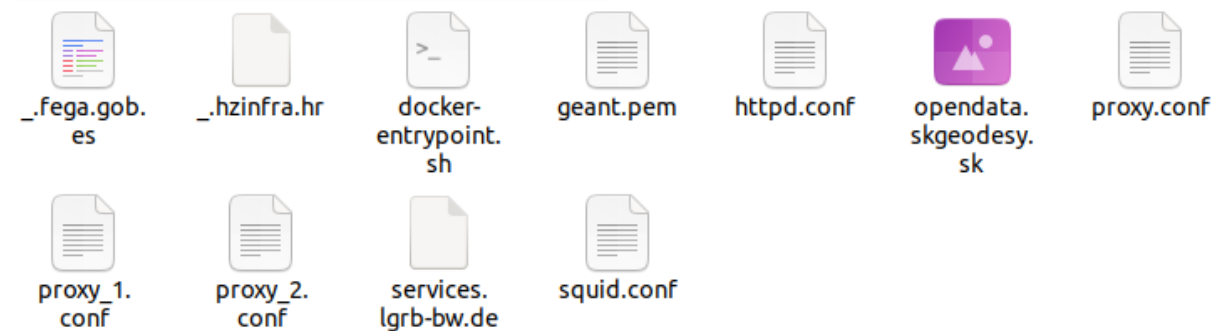


INSPIRE Reference Validator: Admins

Understanding Validator Docker ZIP

Lets understand each part of this zip:

- **VERSION:** This is a txt with the version of ETF the Validator is using.
- **validator.war:** This is the etf-webapp.war renamed.
- **ui.zip:** This is the UI for the Validator zipped.
- **README.md:** The readme of the GitHub with info about the validator and how to deploy it.
- **LICENSE:** European Union Public License for the Validator.
- **Dockerfile:** Instructions for Docker to build the Docker image (Discussed later).
- **res:**
 - **docker-entrypoint.sh:** a series of commands the Docker container will run once created in order to load the ETF, the Validator and other components (Apache, Squid, ...).
 - **httpd.conf, proxy_1.conf, proxy_2.conf:** The configuration configs for Apache. In order, the Apache enabled modules, the proxy for the inspire resources and the proxy for the validator.
 - **squid.conf:** The Squid configuration.
 - The rest of files are the certificates for various spatial agency webs, so the Validator can access their resources



INSPIRE Reference Validator: Admins

How to build your own Validator Docker image

When we have finished editing the Validator files with the configurations we want (we will discuss this configurations next) we can build our custom Validator Docker image with this command inside the extracted Validator zip:

```
docker build . -t validator
```

Once the Image has been built we can deploy it with the same command we used before, changing the default Validator image url with the name of the image we just built:

```
docker run --name inspire-validator -d -p 8090:8090 -v ~/etf:/etf validator
```

```
(base) danielmartin@danielmartin:~/Descargas/inspire-validator-2024.1 (1)$ docker build . -t validator
[+] Building 14.0s (49/49) FINISHED                                docker:default
[+] [internal] load build definition from Dockerfile
```

```
(base) danielmartin@danielmartin:~/Descargas/inspire-validator-2024.1 (1)$ docker run --name inspire-validator -p 8090:8090 validator
Service 'hwdrivers' needs non existent service 'dev'
Service 'machine-id' needs non existent service 'dev'
* Caching service dependencies ... [ ok ]
* Starting apache2 ...AH00112: Warning: DocumentRoot [/etf/] does not exist
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.16
.2. Set the 'ServerName' directive globally to suppress this message
[ ok ]
/docker-entrypoint.sh: line 12: a2enmod: command not found
/docker-entrypoint.sh: line 13: a2enmod: command not found
/docker-entrypoint.sh: line 14: a2enmod: command not found
/docker-entrypoint.sh: line 15: a2enmod: command not found
/docker-entrypoint.sh: line 16: a2dissite: command not found
/docker-entrypoint.sh: line 17: a2ensite: command not found
/docker-entrypoint.sh: line 18: a2ensite: command not found
* Gracefully restarting apache2 ...AH00112: Warning: DocumentRoot [/etf/] does not exist
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.16
.2. Set the 'ServerName' directive globally to suppress this message
[ ok ]
* Initializing cache directory /var/spool/squid ... [ ok ]
* Starting squid ...2024/05/29 11:18:53| WARNING: BCP 177 violation. Detected non-functional IPv6
opback.
[ ok ]
```

INSPIRE Reference Validator: Admins

Parameters and configurations

These are the most used configurations the Validator allows from the Dockerfile:

- ENV ETF_DL_TESTPROJECTS_ZIP: url to the ets repository
- ENV ETF_DL_TESTPROJECTS_DIR_NAME: name the ETS folder will have
- ENV ETF_DL_TESTPROJECTS_OVERWRITE_EXISTING: If true will overwrite the old ETS folder if there is one
- ENV MAX_MEM max: Memory the ETF will use, in MB or “max” (max available memory-768MB if at least 3GB available)

Less used:

- MAINTAINER: Name and email of the File maintainer.
- ETF_RELATIVE_URL: The relative url the validator will have

```
# Possible values: “none” or URL to ZIP file
ENV ETF_DL_TESTPROJECTS_ZIP https://github.com/inspire-eu-validation/ets-repository/archive/v2024.1.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 available)
ENV MAX_MEM max
```

INSPIRE Reference Validator: Admins

Parameters and configuration

Inside the `ui.zip/validator/js` we have the file `config.js`, in here we can modify parameters about the UI:

- **Captcha enabled:** If the UI will use the Captcha to start a test. (true / false)
- **Betabanner:** If we set this a true the UI will appear as the Staging one. (true / false)
- **Environment:** This is useful to differentiate environments between them (string)
- **ValidatorVersionLabel:** This will appear on the header of the main html, so users can see version used of the validator (<https://github.com/INSPIRE-MIF/helpdesk-validator/discussions/1035>) (string)
- **serverURL:** main url for the ETF API (string)
- **serverRealURL:** url for the ETF API (with alias) if we are using an alias on our domain (string)
- **ServerDirectURL:** url for the ETF API (Real domain name) if we are using an alias on our domain (string)
- **serverToken:** The token the Validator internally uses for the ETF API (string)
- **swaggerURL:** url for the Swagger-UI of the ETF (string)
- **timeUpMessage:** Message that will appear on the test reports menu (string)

```
var captchaEnabled = true;
var betaBanner = false;
var labelStaging = false;
var environment = "PROD";
var validatorVersionLabel = "2024.1 (2024-05-07)";

// PROD
var serverURL = "https://inspire.ec.europa.eu/validator/v2/";
var serverRealURL = "https://inspire.ec.europa.eu/validator/v2/";
var serverDirectURL = "https://inspire.ec.europa.eu/validator/v2/";
var serverCaptchaURL = "https://inspire.ec.europa.eu/validator/captcha/verify";
var serverToken = "";
var swaggerURL = "https://inspire.ec.europa.eu/validator/swagger-ui.html";
var timeUpMessage= "Test Reports are kept for a maximum of 48 hours.";

var captchaEnabled = true;
var betaBanner = false;
var labelStaging = true;
var environment = "STAGING";
var validatorVersionLabel = "2024.1 (2024-05-07)";

// STAGING
var serverURL = `http://localhost:8090/validator/v2/`;
var serverRealURL = "http://localhost:8090/validator/v2/";
var serverDirectURL = "http://localhost:8090/validator/v2/";
var serverCaptchaURL = "http://localhost:8090/validator/captcha/verify.php";
var serverToken = "";
var swaggerURL = "http://localhost:8090/validator/swagger-ui.html";
var timeUpMessage= "";
```

INSPIRE Reference Validator: Admins

Parameters and configurations

Additionally, if we are going to deploy the validator in a server, in addition of changing the urls in the config.js file, we will also have to change it in the file validator.war/WEB-INF/classes, in the property etf.webapp.base.url:

```
#####  
# ETF CONFIGURATION PROPERTY FILE  
#####  
  
# Version of the property path, used for backward compatibility.  
# (do not modify unless you know what you are doing)  
etf.config.properties.version = 2  
  
#####  
## General properties  
#####  
  
# Modify the base URL which points to the deployed web application.  
# The URL is used to reference this ETF instance from the Test Reports.  
# Example: http://yourserver/etf-webapp  
etf.webapp.base.url = http://localhost:8090/validator
```

Thank you!



© 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.