



This material has been developed under Specific Contract 1 within the Framework Contract 08143.2017.003-2017.579 (JRC Ref. CCR.B.C934097.X1)

# *INSPIRE Validator Workshop*

## *Day 2: Developers*



# Introduction to the workshop

- In an effort to make the adoption of the INSPIRE Directive easier, the Joint Research Center of the European Commission has developed the INSPIRE validator, which implements a set of tests over data sets, services and metadata.
- This workshop covers all the relevant aspects of the validator from two perspectives:
  - **Users** of the validators: what are the main components, how to integrate the validator in their workflow and how to provide feedback;
  - **Developers** interested in the technology behind the validator, looking for deploying their own instance and/or add new functionalities.

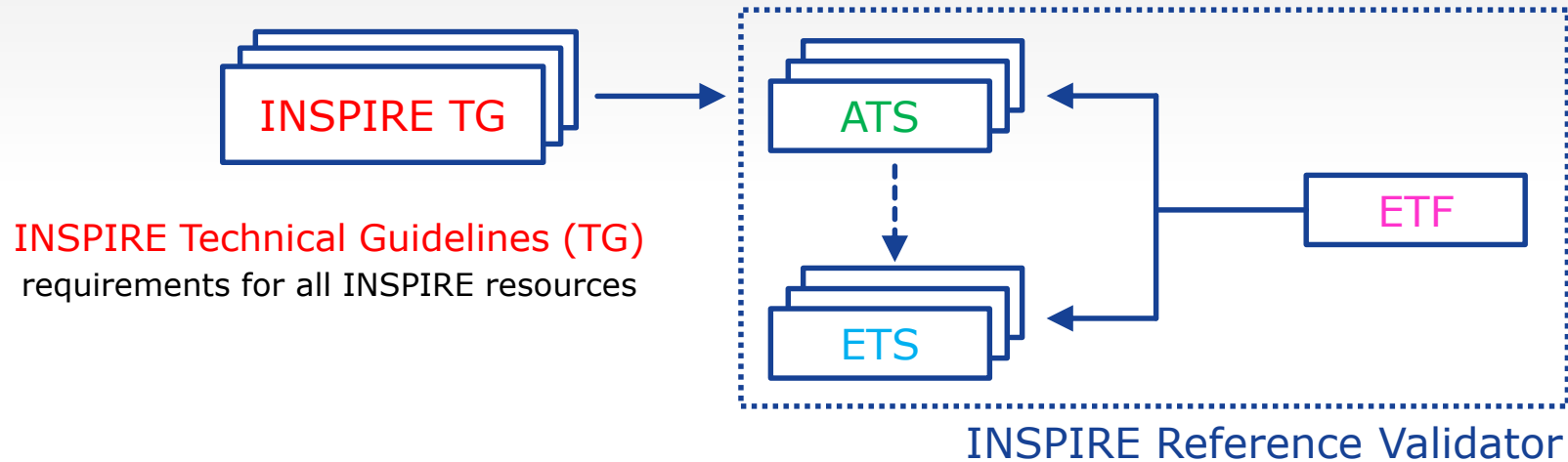
# Developers' workshop

- Main **target users**:
  - Maintainers of Spatial Data Infrastructures (SDIs) with a technical background
  - Open source software developers
- **Prerequisites**:
  - Familiarity with the INSPIRE technical framework
  - Web service and webapp deployment
  - Basic system administration
  - API usage
  - Proficiency in Java and JavaScript

# INSPIRE Reference Validator Components

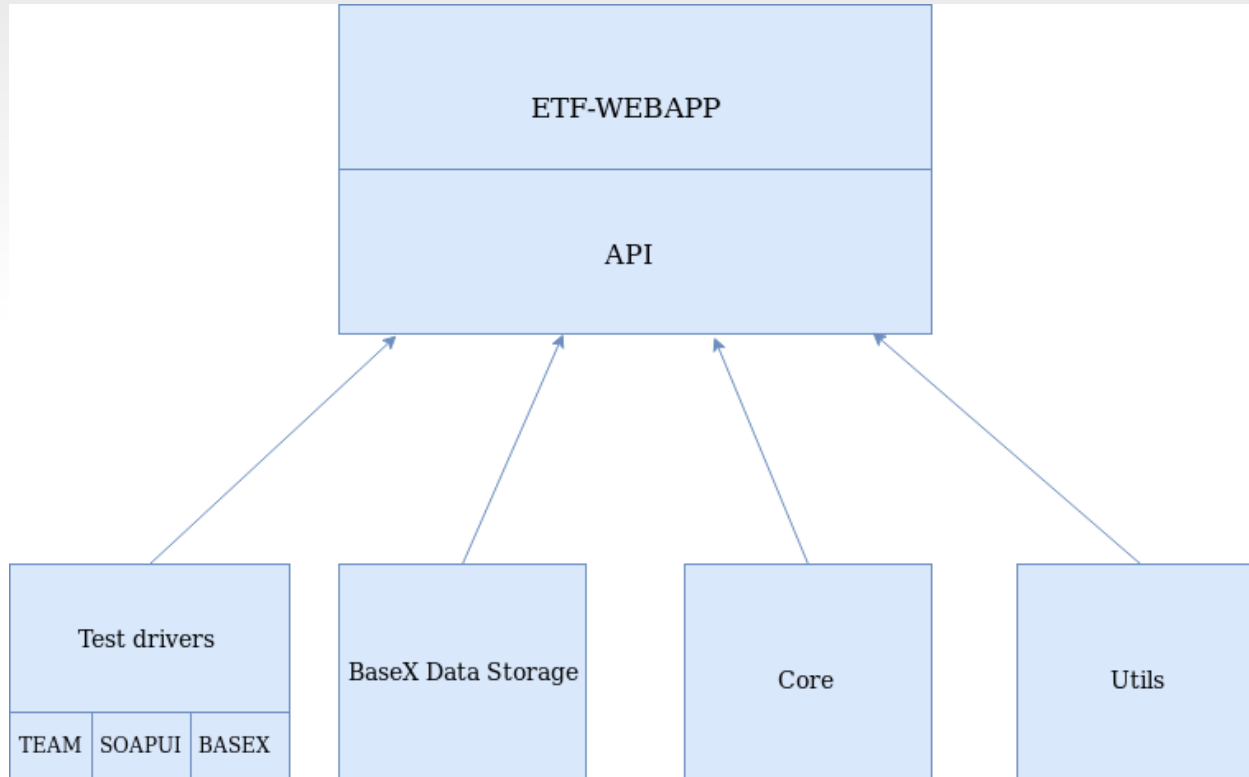
**Abstract Test Suites (ATS)** high-level descriptions of test cases

**Testing Framework (ETF)** software where ETS are run



**Executable Test Suites (ETS)** lower-level descriptions of test cases

# Architecture of the ETF system



# Architecture of the ETF system

## WebApp

- Developed using jQuery-mobile for the UI, plus Thymeleaf as template engine
- This component also deploys the API interface, and instance all the dependant component and services:
  - TestDrivers
  - DataStorageService
  - Core
  - Utils libraries

# Architecture of the ETF system

## TestDrivers

- **SoapUI**
  - SoapUI is a testing application for service-oriented architecture. Its functionality covers web service inspection, invoking, development, simulation and mocking, functional testing, load and compliance testing
  - It is used to analyze service responses
  - Can be extended with Groovy scripts to make assertions over the XML responses
- **BaseX**
  - XML processor and database
  - Query XML documents using XQuery to build the test cases
- **TEAM Engine**
  - Test driver to execute OGC TEAM Engine tests in a remote instance
  - Used in WFS 2.0 OGC 09-025r2/ISO 19142 implementation Test Suite

# Architecture of the ETF system

## **DataStorage**

- The ETF uses BaseX also as a persistence engine
- All the objects (TestRuns, TestObjects, TestResults) have an XML representation on disk
- The documents are retrieved using the DataStorageService instantiated on the webapp, using a facade to create the XQuery expressions



# Architecture of the ETF system

## Using the ETF through the API

We send a POST HTTP request to the endpoint  
<http://inspire.ec.europa.eu/validator/v2/TestRuns>

Using as payload this data

```
{"label":"Workshop  
Test","executableTestSuiteIds":["ElDeec9d674-d94b-4d8d-b744-  
l309c6cae1d2"],"arguments":{},"testObject":{"resources":{"servi  
ceEndpoint":"http://www.ign.es/wms-inspire/mapa-raster"}}}
```

# Architecture of the ETF system

## Using the ETF through the API

Checking the response of the last API call, we can get the TestRun EID to monitor the results, making a GET request.

In this example:

**`http://inspire.ec.europa.eu/validator/v2/TestRuns/EID47407575-105c-4bf3-8d8c-0643e2201bfe/progress`**

# Architecture of the ETF system

## Using the ETF through the API

After the test run has finished, we can always obtain the test report in HTML, JSON or XML formats.

Using a GET request on the API:

**`http://inspire.ec.europa.eu/validator/v2/EID47407575-105c-4bf3-8d8c-0643e2201bfe[.xml | html]`**

# Architecture of the ETF system

## Using the ETF through the API

- The API can be also used to:
  - Check the status of the service
  - List all the ExecutableTestSuites available
  - Check the TestObjectTypes (services or data) available to test

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

# Architecture of the ETF system

## ETS integration and test drivers (BaseX-based tests)

Test Object Types in the BaseX test driver:

- Set of XML documents
- Metadata records
- GML feature collections
- WFS 2.0 feature collections
- GML 3.2 feature collections
- GML 2.1/GML 3.1 feature collections
- INSPIRE SpatialDataSet documents
- CityGML 2.0 CityModel documents
- CityGML 1.0 CityModel documents

# Architecture of the ETF system

## ETS integration and test drivers (BaseX-based tests)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ExecutableTestSuite
3   xmlns="http://www.interactive-instruments.de/etf/2.0"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   id="EID59692c11-df86-49ad-be7f-94alelddd8da" xsi:schemaLocation="http://www.i
6   <itemHash>bQ==</itemHash>
7   <remoteResource>http://github.com/inspire-eu-validation/ets-repository/metada
8   <localPath>/auto</localPath>
9   <label>Common Requirements for ISO/TC 19139:2007 based INSPIRE metadata recor
10  <description>
11    <![CDATA[<br/><br/><b>This is a draft version. It has limitations and is
12      target=" _blank">in GitHub</a>.<br/><br/>
13    Known limitations are documented in the description of the applicable test case o
14    There is a general limitation in all assertions that polymorphism and containment
15    3, A.4 and A.5) are not supported. However, the current Abstract Test Suite does
16    against the ISO/OGC schemas without extensions). It is therefore unclear if this
17    <br/><br/>
18    Source: <a href="http://inspire.ec.europa.eu/id/ats/metadata/2.0/common" target="
19    </description>
20    <reference>../../inspire-md-bsxets.xq</reference>
21    <version>0.1.1</version>
22    <author>Consortium Bilbomatica, Guadaltel y Geograma</author>
23    <creationDate>2018-06-30T00:00:00Z</creationDate>
24    <lastEditor>Consortium Bilbomatica, Guadaltel y Geograma</lastEditor>
25    <lastUpdateDate>2019-05-16T11:40:00Z</lastUpdateDate>
26  </tags>
27    <tag ref="EIDc6567beb-fc33-4f2e-865d-0c3ee5b3d1ae"/>
28  </tags>
29    <testDriver ref="EID4dddc9e2-1b21-40b7-af70-6a2d156ad130"/>
30    <translationTemplateBundle ref="EID70a263c0-0ad7-42f2-9d4d-0d8a4ca71b52"/>
31    <ParameterList name="ETF Standard Parameters for metadata XML test objects">
32      <parameter name="files_to_test" required="true">
33        <defaultValue>.</defaultValue>
34        <description ref="TR.filesToTest"/>
35        <allowedValues>.</allowedValues>
36        <type>string</type>
37      </parameter>
38      <parameter name="tests_to_execute" required="false">
39        <defaultValue>.</defaultValue>
40        <description ref="TR.testsToExecute"/>
41        <allowedValues>.</allowedValues>
42        <type>string</type>
43      </parameter>
44    </ParameterList>
45    <supportedTestObjectTypes>
46      <testObjectType ref="EID5a60dded-0cb0-4977-9b06-16c6c2321d2e"/>
47    </supportedTestObjectTypes>
```

tag for ETF organization

document with exception translations

testDriver. Fixed value

Test object type

# Architecture of the ETF system

## ETS integration and test drivers (SoapUI-based tests)

Test Object Types in the SoapUI test driver:

- Web service
- OGC Web Feature Service
- OGC Web Feature Service 2.0
- OGC Web Feature Service 1.1
- OGC Web Feature Service 1.0.0
- OGC Web Map Service
- OGC Web Map Service 1.3.0
- OGC Web Map Service 1.1.1
- OGC Web Map Tile Service
- OGC Web Map Tile Service 1.0
- OGC Web Coverage Service
- OGC Web Coverage Service 2.0
- OGC Web Coverage Service 1.1
- OGC Web Coverage Service 1.0.0
- OGC Sensor Observation Service
- OGC Sensor Observation Service 2.0
- OGC Catalogue Service
- OGC Catalogue Service 3.0
- OGC Catalogue Service 2.0.2
- OGC CSW-ebRIM Registry Service 1.0
- Atom feed

# Architecture of the ETF system

## ETS integration and test drivers (SoapUI-based tests)

Conformance Class: View Service - WMS

Overview | TestSuites | WS-Security Configurations | Security Scan Defaults

**Project Summary**

File Path: [C:\Users\daniel.navarro\workspace\\_SoapUI\ETS view service](C:\Users\daniel.navarro\workspace_SoapUI\ETS view service)

**Interface Summary**

**Test Summary**

TestSuites	4
TestCases	62

Properties:

Name	Value
serviceEndpoint	
service	WMS
version	1.3.0
extended_capabilities_scenario	1
lastServiceEndpoint	
authUser	
authPwd	
etf.ignore.properties	service, version, extended_capabilities_scenario, lastServiceEn...
etf.tag.ids	EID7c15a770-986a-4aa0-b4cd-7facbca96a1d
etf.translation.template.collection.id	EIDfa68eb83-a25a-4009-84dd-036de4539c93
etf.supported.test.object.type.ids	EID9981e87e-d642-43b3-ad5f-e77469075e74
etf.author	Consortium Bilbomatica, Guadaltel & Geograma
etf.creation.date	2018-11-07T00:00:00
etf.last.editor	Consortium Bilbomatica, Guadaltel & Geograma
etf.last.update.date	2019-06-07T13:15:00
etf.version	0.3.11
etf.reference	<a href="https://github.com/inspire-eu-validation/ets-repository/tree...">https://github.com/inspire-eu-validation/ets-repository/tree...</a>

Description | Properties | Load Script | Save Script

Test object type

tag for ETF organization

document with exception translations



# Architecture of the ETF system

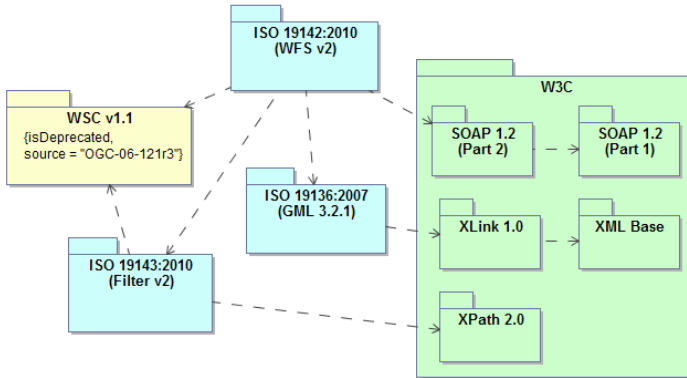
## ETS integration and test drivers (TEAM Engine)

The only integration available on the ETF right now is the WFS 2.0 Test Suite.  
Check <http://cite.opengeospatial.org/teamengine/about/wfs/2.0.0/site/> for references.

### WFS 2.0 Conformance Test Suite

#### Scope

This executable test suite (ETS) verifies that a WFS 2.0 implementation conforms to OGC 09-025r2/ISO 19142 (*Geographic information -- Web feature service*) and related standards as depicted in Figure 1. Conformance testing is a kind of "black box" testing that examines externally visible characteristics or behaviors of the SUT and is independent of any implementation details.




Test endpoint <http://cite.opengeospatial.org/teamengine/rest/suites/wfs20/>

# Architecture of the ETF system

## ETS integration and test drivers (TEAM Engine)

The Test Suite only needs the service endpoint to be tested and any necessary credential.

**OGC Test Suites (remote execution)**

 **WFS 2.0 (OGC 09-025r2/ISO 19142) Conformance Test Suite**

☐ use

**Description:**

This executable test suite (ETS) checks WFS 2.0 implementations for conformance with respect to OGC 09-025r2/ISO 19142 (Geographic information -- Web feature service) and related standards. Conformance testing is a kind of "black box" testing that examines externally visible characteristics or behaviors of the SUT and is independent of any implementation details.

This Executable Test Suite is executed using a remote TEAM Engine instance hosted by OGC for their Compliance Program (CITE). The results are transformed into the ETF internal report format. Some information that is typically included in ETF test results is not included in the TEAM Engine reports and cannot be included in this test report.

Please report any issues or problems with the OGC CITE tests in the [OGC Compliance Forum](#).

**Tags:**

- OGC Test Suites (remote execution)

**Applicable to Test Object Types:**

- OGC Web Feature Service 2.0

**General:**

- Version: 1.32.0
- Author: Open Geospatial Consortium
- Date created: 24.09.2019
- Last editor: Open Geospatial Consortium
- Last updated: 24.09.2019

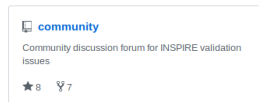
# Code repository

## ATS/ETS

- The ATS repositories are separated for each service or dataset, under the organization <https://github.com/inspire-eu-validation>
- The ETS files are hosted in a separated repository (each in a sub-directory):
  - <https://github.com/inspire-eu-validation/ets-repository>



### Pinned repositories



Find a repository... Type: All Language: All

### download-wfs

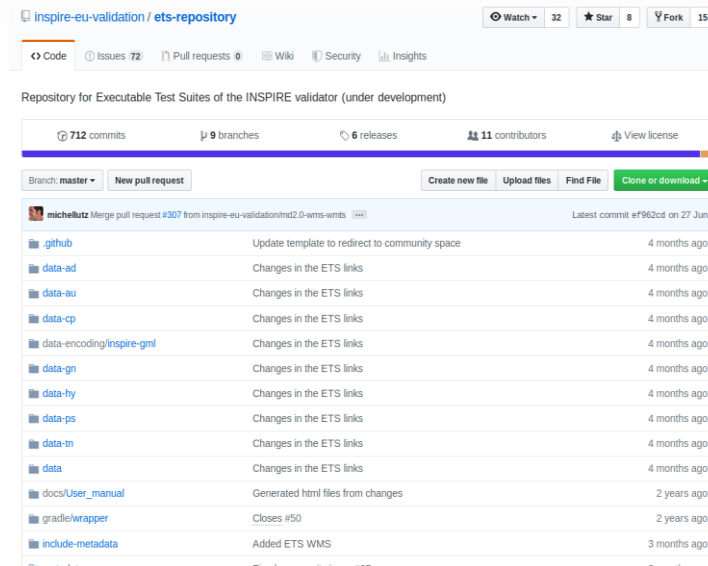
CC0-1.0 0 stars 0 forks Updated 2 days ago

### download-wcs

CC0-1.0 0 stars 0 forks Updated 2 days ago

### download-atom

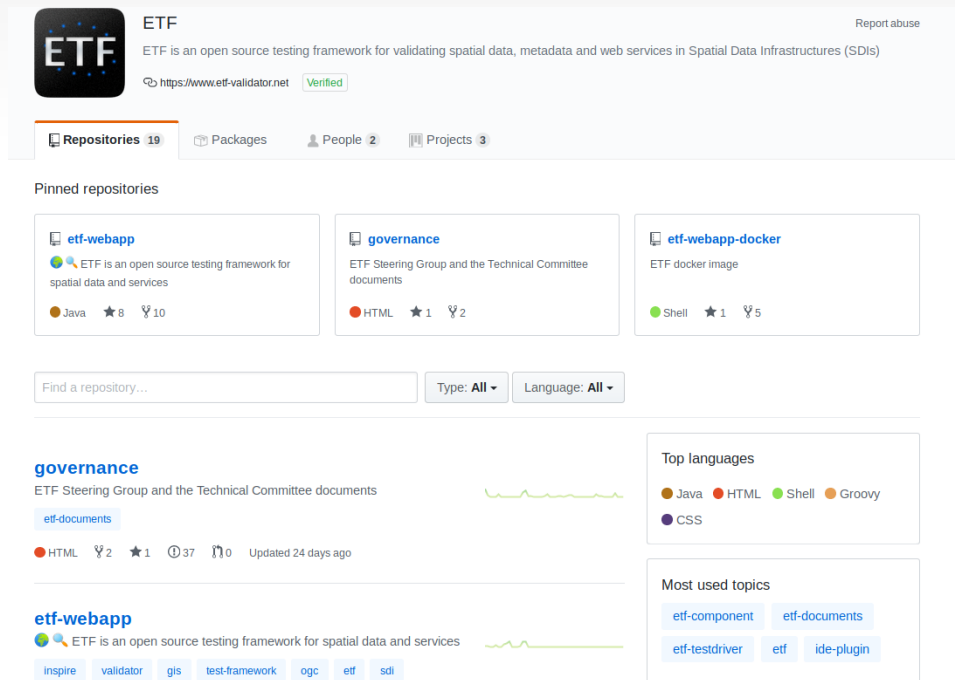
CC0-1.0 0 stars 0 forks Updated 2 days ago



# Code repository

## ETF

- The GitHub organization <https://github.com/etf-validator/> hosts all the ETF related repositories:
  - etf-webapp
  - etf-stdot
  - etf-sui-ae
  - etf-bsxds
  - etf-core
  - etf-gmlgeox
  - etf-spi
  - etf-suitd
  - etf-bsxtd



The screenshot shows the GitHub organization page for ETF. At the top, the organization name 'ETF' is displayed with a verified badge and a link to the website <https://www.etf-validator.net>. Below this, statistics show 19 repositories, 2 people, and 3 projects. The 'Pinned repositories' section features three items: 'etf-webapp' (Java, 8 stars, 10 forks), 'governance' (HTML, 1 star, 2 forks), and 'etf-webapp-docker' (Shell, 1 star, 5 forks). A search bar and filters for repository type and language are present. The main content area displays the 'governance' repository details, including its description, repository type (HTML), star/fork counts, and update status. Below this, the 'etf-webapp' repository is partially visible. On the right, sidebars list 'Top languages' (Java, HTML, Shell, Groovy, CSS) and 'Most used topics' (etf-component, etf-documents, etf-testdriver, etf, ide-plugin).

# Building and deployment of ETF

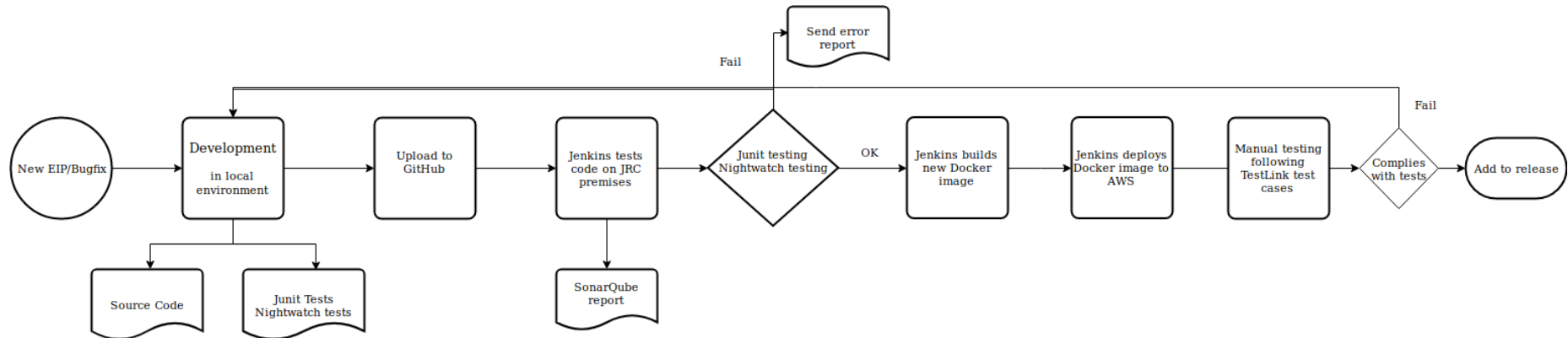
Deployment on a **servlet container** (Tomcat):

- Clone the source code from GitHub <https://github.com/etf-validator/etf-webapp> and select the specific branch
- The configuration for this specific deployment can be set up on the file `etf-config.properties`
- Inside the ETF folder, execute `./gradlew build war`
  - this will download all the dependencies from the other modules
  - these dependencies are hosted on an artifact repository
- Move the file `etf-webapp.war` from the folder `/build/libs` to the `webapps` folder on Tomcat
- Start the server and access `localhost:8080`

# Building and deployment of ETF

## Testing and continuous integration:

- Alongside with the cloud deployment, an initiative for automatic testing and continuous integration is being developed:
  - <https://github.com/etf-validator/etf-system-tests>
- Unit tests are run on compilation time on the ETF webapp
- For UI changes, a repository for testing has been created, using Nightwatch.js
- It is planned to have a complete cycle of testing at each pull request, create a build using Jenkins and deploy automatically with Docker containers



# Building and deployment of ETF

Deployment on **servlet container** (Tomcat):

- If the environment variable `ETF_WEBAPP_PROPERTIES_FILE` or the Java property `etf.webapp.properties.file` is set, the file will be used to configure the etf directory.
- If a `/etc/etf/etf-config.properties` file exists, the file will be used to configure the etf data directory.
- If an `etf-config.properties` file exists in the 'etf' subfolder of the root filesystem (`/etf`), the file will be used to configure the etf directory.
- If a `etf-config.properties` file exists in the home subfolder `'~/etf'` (note that the directory is hidden) on Linux, the file will be used to configure the etf directory.
- If the `etf-config.properties` file is not found, a template configuration and the default ETF data directory structure will be created in the hidden home subfolder `'~/etf'`.

# Building and deployment of ETF

## Adding ETS files

- Create a new subfolder in the project subfolder on the ETF data directory and copy the Executable Test Suites in there.
- The easiest way to do this is to download a .zip file from the ETS repository on GitHub.
- Restart the server after adding all the ETS files.



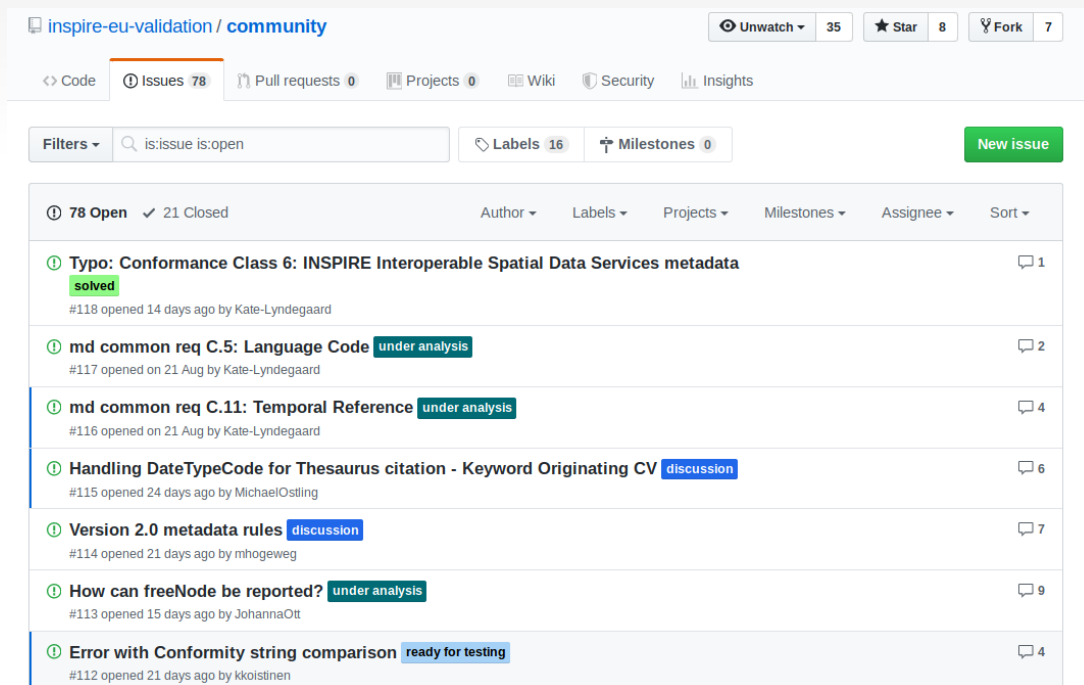
# Building and deployment of ETF

## Deploying with Docker

- Use the content of the repository <https://github.com/etf-validator/etf-webapp-docker> as a reference.
- Move the built .war file inside the folder with the Dockerfile.
- On the Dockerfile, configure the relative URL to be compatible with the name of the .war file.
- On the file `res/docker-entrypoint.sh`, configure the ETS repository URL
- Execute `docker build -t [tag]`
- Execute `docker run -p [host_port]:8080 [tag]`
- Access on `localhost:[host_port]/[relative_url]`

# How to collaborate

- Issues can be reported at <https://github.com/inspire-eu-validation/community>
- Any improvement relevant to the ETF will be moved to the ETF Steering Group discussion board at <https://github.com/orgs/etf-validator/projects/2>



inspire-eu-validation / community

Unwatch 35 Star 8 Fork 7

Code Issues 78 Pull requests 0 Projects 0 Wiki Security Insights

Filters is:issue is:open Labels 16 Milestones 0 New issue

78 Open ✓ 21 Closed	Author	Labels	Projects	Milestones	Assignee	Sort
<b>Typo: Conformance Class 6: INSPIRE Interoperable Spatial Data Services metadata</b> solved #118 opened 14 days ago by Kate-Lyndegeard						1
<b>md common req C.5: Language Code</b> under analysis #117 opened on 21 Aug by Kate-Lyndegeard						2
<b>md common req C.11: Temporal Reference</b> under analysis #116 opened on 21 Aug by Kate-Lyndegeard						4
<b>Handling DateTypeCode for Thesaurus citation - Keyword Originating CV</b> discussion #115 opened 24 days ago by MichaelOstling						6
<b>Version 2.0 metadata rules</b> discussion #114 opened 21 days ago by mhogeweg						7
<b>How can freeNode be reported?</b> under analysis #113 opened 15 days ago by JohannaOtt						9
<b>Error with Conformity string comparison</b> ready for testing #112 opened 21 days ago by kkoistinen						4

# How to collaborate

## How to collaborate - ETF

- All the bugs related to the technical parts of the validator, aside from the INSPIRE validator issues on specific tests or the UI, shall be reported on the <https://github.com/etf-validator/etf-webapp> repository.
- New features or improvements can be proposed as **EIPs (ETF Improvement Proposals)** at <https://github.com/etf-validator/governance>
- The **ETF Steering Group (SG)** evaluates the significance of the proposal, while the **ETF Technical Committee (TG)** evaluates its technical feasibility; the TC can invite the reporter to provide a pull request to add the new functionality on the relevant ETF repositories.

# How to collaborate

## How to collaborate - ETS

### Bugfix:

- Clone the repository
- Create a bugfix branch
- Fix the bug locally
- Make a pull request

```
MINGW64: c/projects
Daniel.Navarro@H-0124 MINGW64 /c/projects
$ git clone https://github.com/inspire-eu-validation/ets-repository.git
```

```
MINGW64: c/projects/ets-repository
Daniel.Navarro@H-0124 MINGW64 /c/projects
$ cd ets-repository/

Daniel.Navarro@H-0124 MINGW64 /c/projects/ets-repository (csw-tg-3.1)
$ git checkout git checkout -b [name_bugfix_branch]
```

```
353 > <TestAssertion id="EID87ee2219-2ba5-4a27-91ac-2b3bf5730012">
354 >   <label>md common req C.10: Responsible Organization</label>
355 >   <description><![CDATA[<p>Test that the responsible organization metadata is provided</p>
356 >   <p>More information: <a href="http://inspire.ec.europa.eu/id/ats/metadata/2.0/common/responsible-organisation" target="_blank">Responsible Organizat
357 >   <parent ref="EID61e80628-c181-11e8-a355-529269fb1459"/>
358 >   <expectedResult>NOT_APPLICABLE</expectedResult>
359 >   <expression>
360 >   let $iso19115_CIRoleCode := ('resourceProvider','custodian','owner','user','distributor','originator','pointOfContact','principalInvestigator','processor','publisher','author')
361 >   let $messages :=
362 >     (for $record in $records
363 >     let $rid := $record/gmd:fileIdentifier/*text()
364 >     let $pocs :=
365 >       if ($record/gmd:hierarchyLevel/gmd:MD_ScopeCode/@codeListValue = 'service') then
366 >         $record/gmd:identificationInfo/srv:SV_ServiceIdentification/gmd:pointOfContact/gmd:CI_ResponsibleParty
367 >       else
368 >         $record/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:pointOfContact/gmd:CI_ResponsibleParty
369 >     return
370 >     if(not($pocs)) then
371 >       local:addMessage('TR.missingResponsibleParty', map { 'filename': local:filename($record), 'id': $rid })
372 >     else
373 >       for $poc in $pocs
374 >       let $orgName := $poc/gmd:organisationName
375 >       let $email := $poc/gmd:contactInfo/gmd:CI_Contact/gmd:address/gmd:CI_Address/gmd:electronicMailAddress
376 >       let $validCodeList :=
377 >         for $x in $poc/gmd:role/gmd:CI_RoleCode
378 >         where ($x/@codeList = 'http://standards.iso.org/iso/19139/resources/gmxCodeLists.xml#CI_RoleCode' or
379 >         $x/@codeList = 'http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas/resources/codeList/gmxCodeLists.xml#CI_RoleCode' or
380 >         $x/@codeList = 'http://schemas.isotc211.org/19139/resources/gmxCodeLists.xml#CI_RoleCode' or
381 >         $x/@codeList = 'https://standards.iso.org/iso/19139/resources/gmxCodeLists.xml#CI_RoleCode' or
382 >         $x/@codeList = 'https://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas/resources/codeList/gmxCodeLists.xml#CI_RoleCode' or
383 >         $x/@codeList = 'https://schemas.isotc211.org/19139/resources/gmxCodeLists.xml#CI_RoleCode')
384 >       return $x
385 >     return
386 >     if (not($orgName) or not($orgName/gco:CharacterString/node() or $orgName/gmx:Anchor/node())) then
387 >       local:addMessage('TR.noMetadataContactOrganisationName', map { 'filename': local:filename($record), 'id': $rid })
388 >     else if (not($email) or not($email/gco:CharacterString/node() or $email/gmx:Anchor/node())) then
389 >       local:addMessage('TR.noMetadataContactEmailAddress', map { 'filename': local:filename($record), 'id': $rid })
390 >     else if (not(count($poc/gmd:role/gmd:CI_RoleCode) = count($validCodeList))) then
391 >       local:addMessage('TR.wrongCodeList', map { 'filename': local:filename($record), 'id': $rid })
392 >     else if (not($poc/gmd:role) or not($poc/gmd:role/gmd:CI_RoleCode/@codeListValue = $iso19115_CIRoleCode)) then
393 >       local:addMessage('TR.noPointOfContactRole', map { 'filename': local:filename($record), 'id': $rid })
394 >     else ()
395 >   ) (position() le $limitErrors)
396 > return
397 > (if ($messages) then 'FAILED' else 'PASSED',
398 > local:error-statistics('TR.recordsWithErrors', count(fn:distinct-values($messages//etf:argument[@token='id']/text()))),
399 > $messages)
```

# How to collaborate

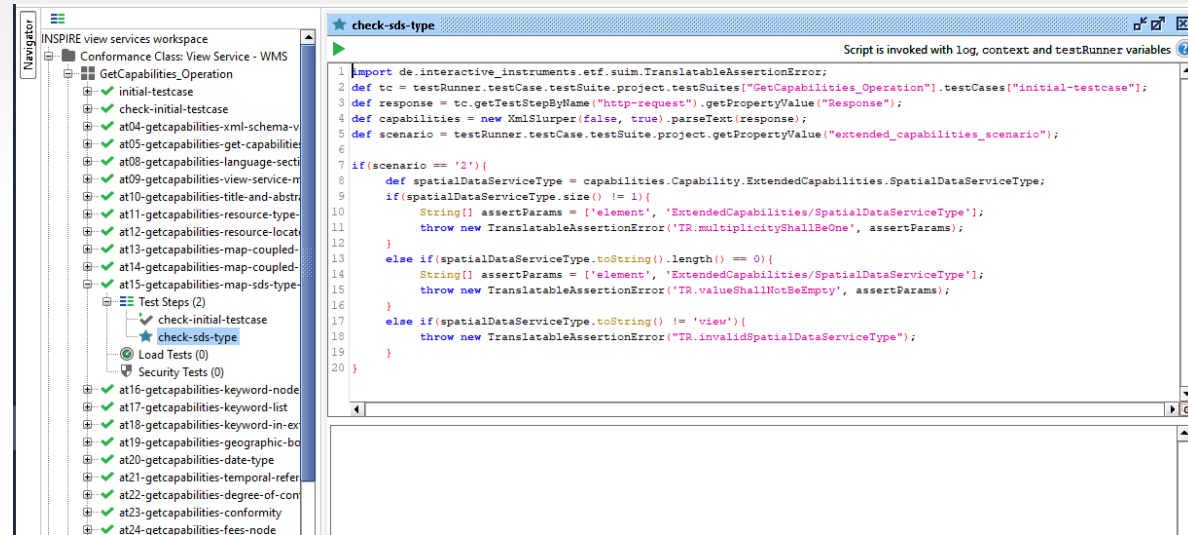
## How to collaborate - ETS

### New developments:

- Clone the repository
- Create a new dev branch
- Develop tests locally
- Make a pull request

```
MINGW64:/c/projects/ets-repository
Daniel.Navarro@H-0124 MINGW64 /c/projects/ets-repository (csw-tg-3.1)
$ git commit -m 'Message commit'
```

```
MINGW64:/c/projects/ets-repository
Daniel.Navarro@H-0124 MINGW64 /c/projects/ets-repository (csw-tg-3.1)
$ git push
```



# THANK YOU

Daniel Navarro – [daniel.navarro@geograma.com](mailto:daniel.navarro@geograma.com)

Carlos Palma – [carlospalma@guadaltel.com](mailto:carlospalma@guadaltel.com)



Bilbomática

