

Comparison test case scenario

Summary

- Goals
- What the test case does?
- How the test case works?
 - General Algorithm
 - Ontologies for testing
 - Detailed algorithm
- To run the scenario

Goals

The comparison test consists of validating the integrity of the vivo form layout for the i18n version and the original 1.11.1 version.

The original version is implemented with the en_US language. The hypothesis is that the i18n version in the en_US linguistic context should have the same behavior as the original version. This is what this test case measures.

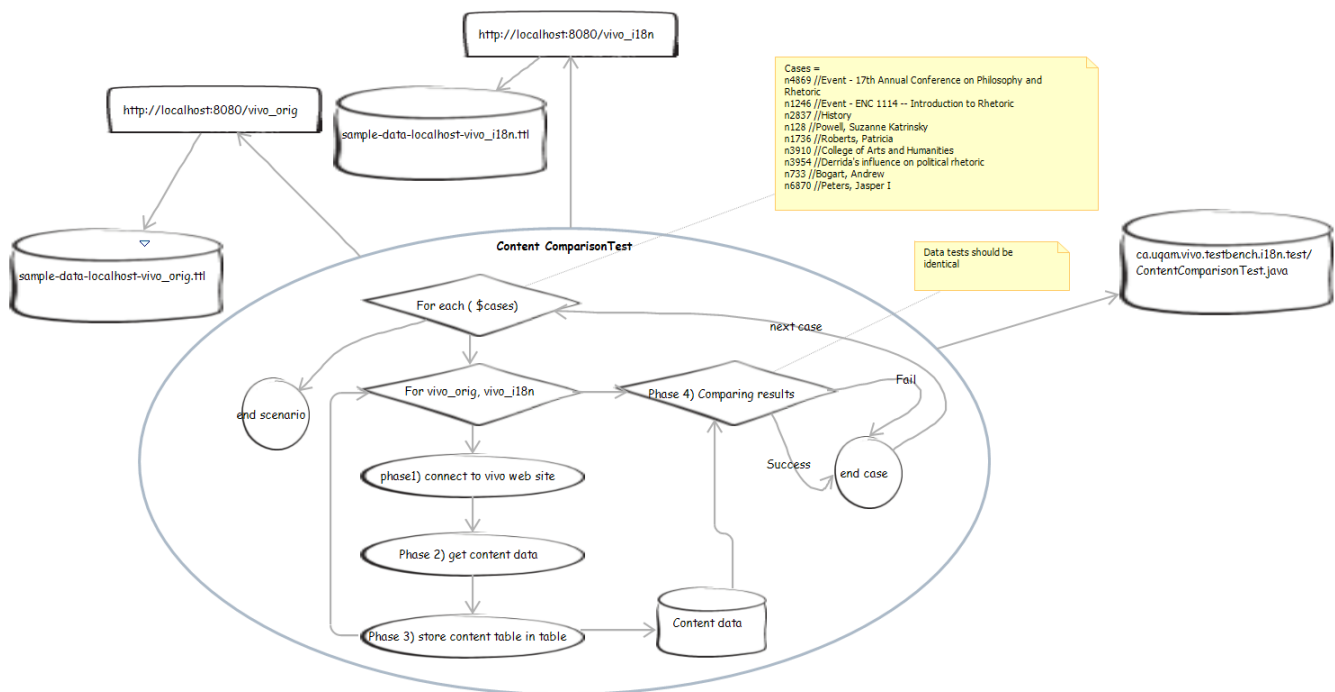
What the test case does?

the test case measures similarities in presentation mode. It does not address vivo's editing features. It compares forms for specific users, events, institutions, and researches. The Test case does not make a comparison of the Capability Map.

How the test case works?

General Algorithm

The figure below shows the general algorithm of the test case. The test case is in communication with two Vivo instances: the original instance and the i18n instance respectively containing ontologies of similar semantics whose URIs are specifically adapted to the linguistic context of the Vivo servers hosting them. In the case of original vivo no particular linguistic tag is assigned to the URI labels, whereas in the case of deployment in the i18n server linguistic tags are adjusted to represent the en_US labels.



Ontologies for testing

The ontologies used for the test are available at these locations

- https://github.com/MichelHeon/vivo-regression-tests/blob/test-1894/src/main/resources/kg/sample-data-localhost-vivo_i18n.ttl
- https://github.com/MichelHeon/vivo-regression-tests/blob/test-1894/src/main/resources/kg/sample-data-localhost-vivo_orig.ttl

Detailed algorithm

For each case in the scenario, the test case is run in two cycles. The first cycle requests the original VIVO server to extract the data contained in the HTML pages and store them in memory for future comparison. The second cycle repeats in the VIVO i18n environment the same sentences contained in the first cycle in the VIVO i18n environment. The data from this cycle is also stored in memory for future comparison.

Phase 1)

At this test phase, the ontological data are transmitted to the server and the selenium client connects to the concerned server via the root login.

Phase 2)

for each case to be tested a call to the server is sent by selenium. The server reacts to this call and produces the related page.

Phase trois)

the page content is interpreted according to pre-established search criteria and the search result is sent to memory.

Phase quatre)

both cycles are now complete. The test compares the results and ensures that they are identical for each of the cycles. It then sends a pass or fail as appropriate.

To run the scenario

Edit settings.xml

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.1.0 http://maven.apache.org/xsd/settings-1.1.0.
xsd">
  <activeProfiles>
    <activeProfile>ContentComparisonTest</activeProfile>
  </activeProfiles>
</settings>
```

Run maven

```
mvn clean install -s settings.xml
```