Citrus

Context

Automated integration tests for message protocols and data formats.

HTTP REST, JMS, TCP/IP, SOAP, FTP, SSH, XML, JSON and more.

Description

In a typical test scenario the system under test is deployed on some application server and interacts with Citrus over various message transports. During the test run Citrus is able to act on both sides as client and/or server simulating request/response messages.

With each test step you can validate the exchanged messages with expected control data. The test is fully automated and repeatable, so you can easily add the integration tests to your continuous build.

Usage

Your enterprise applications requires detailed integration testing. Citrus provides a tooling for complex integration test with great validation capabilities for XML, JSON and plain text.

- **Interface mocks and simulators:** client and server side
- **Test automation:** repeatable integration tests within your continuous build
- **Message transport connectivity:** HTTP, JMS, TCP/IP, REST, SOAP, XML, JSON, and more
- **Validation:** message header and body assertions

Integration within the test automation framework

Citrus has been integrated to the framework using **maven dependencies** and used for the **REST** integration testing.

The Citrus **Domain Specific Language** (DSL), in combination with **JsonPath**, are used to perform the assertions on the service calls.

Code sample

```
http().client(mockAppClient).send().get("therapeuticExclusion/v2/therapeuticExclusions/${patientSsin}")
                .queryParam("hcPartySsin", "${hcPartySsin}")
                .header("Authorization", "Bearer " + authToken).accept(ContentType.APPLICATION_JSON.
getMimeType());

http().client(mockAppClient).receive().response(HttpStatus.OK).messageType(MessageType.JSON)
                .validate("$[:1].patient.identifier[:1].value", "${patientSsin}")
                .validate("$[:1].hcParty.identifier[?(@.type=='ssin')]", "${hcPartySsin}")
```

Within the project, you can find the code using Citrus in the REST project, in the Assertion module, under *src/test/java/be/cetic/M2/features*

This part of the project is made to design **new test cases** and **should not** be **modified** by **Software vendors**.

Further information

Citrus has a pretty well made and complete documentation including examples on their website : https://citrusframework.org/citrus/reference/2.8.0/html/index.html