# **WEB BROWSER**

Test suites can be generated outside of the jUnit runner. It is easy to demonstrate it using a web browser. Note, that this tutorial is not aimed at beginners and does not cover ecFeed basics. You should already know how to generate a keystore and create a sample model. Furthermore, familiarity with the jUnit runner would be a great advantage.

## 1) Installation

First, a personal ecFeed keystore should be generated (you can use the default one, i.e. 'security.p12', which is located in the '~/.ecfeed/' directory). In this case, its name and location can be arbitrary. Next, it must be installed in the browser. If you use Firefox, access the section 'Preferences -> Privacy & security -> view certificates -> your certificates' and press the 'import' button. Note, that the default keystore password is 'changeit'. However, as it suggests, you might want to change it.

# 2) Test stream

To send the same request as the jUnit runner does, copy and paste the following URL. Don't forget to change the model UUID:

https://develop-gen.ecfeed.com/testCaseService?requestType=requestData&request=%7B%22method%22:%22void %20com.example.test.Demo.typeString,Str

The request is filled with escaped characters, and therefore, it might be difficult to read. The more human-friendly version is below. It is exactly the same, however, in this case, simple copying and pasting might not work as expected:

https://develop-gen.ecfeed.com/testCaseService?requestType=requestData&request={"method":"void com.example.test.Demo.typeString,

Note, that each response line is a separate JSON file, and there are plenty of intermediate information. The request can be customized with the various parameters (they are the same for all runners). Their definitions can be found on the ecFeed website, which is <a href="https://ecfeed.com/tutorials/junit-integration">https://ecfeed.com/tutorials/junit-integration</a>.

### 3) Default templates

The previous format, in many test scenarios, might not be sufficient. Therefore, there are other possibilities. To use them, add the 'template' parameter to the request, and change the value of 'requestType' to 'requestExport'. Currently, there are four default templates, namely, 'JSON', 'CSV', 'XML', and 'Gherkin'. The updated request looks as follows:

https://develop-gen.ecfeed.com/testCaseService?requestType=requestExport&request=%7B%22method%22:%22void %20com.example.test.Demo.typeString,S

%22,%22model%22:%227482-5194-2849-1943-2448%22,%22template%22:%22JSON%22,%22userData%22:%22%7B %27dataSource%27:%27genNWise%27,%27constraints%27:%27NONE%27%7D%22%7D

#### The human-friendly version:

https://develop-gen.ecfeed.com/testCaseService?requestType=requestExport&request={"method":"void com.example.test.Demo.typeString,Strin

### 4) Custom templates

If any of the default templates does not satisfy you needs, you can design a custom one. To do it, you must define the 'template' string. It might consist of various element and the manual can be found on the ecFeed website, which is https://ecfeed.com/tutorials/export-templates-documentation.

For example, it might take the following form:

"template": "[Header]\n\$1.name,\$2.name\n[TestCase]\n\$1.value,\$2.value\n[Footer]\n"

#### What results in:

https://develop-gen.ecfeed.com/testCaseService?requestType=requestExport&request=%7B%22method%22:%22void%20com.example.test.Demo.typeString,

#### The human-friendly version:

 $https://develop-gen.ecfeed.com/testCaseService?requestType=requestExport\&request=\{"method":"void com.example.test.Demo.typeString,Str$