

|  |
| --- |
|  |
|  |

| Related Artifacts | |
| --- | --- |
| Ref. | Name |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Abbreviations and Acronyms | |
| DDT | [Data Driven Testing](https://ctc-customs.atlassian.net/wiki/pages/viewpage.action?pageId=45318221) |
| SoapUI | The web service testing application for service-oriented architectures (SOA) and representational state transfers (REST) |
| SoapUI project | In SoapUI, work is organized into projects, which are displayed under the root node in the workspace navigator. A project can contain any number of functional tests, load tests and service simulations required for testing purposes. |
| DDT Framework Library | Library of scrips created on Groovy to improve testing process of [DDT](https://ctc-customs.atlassian.net/wiki/pages/viewpage.action?pageId=45318221) in SoapUI |
| DDT Framework Plugin | Plugin for SoapUI developed with DDT Framework Library which adds the test data descriptions to the Test Suite Log during test execution and add user cookies to request if it required |

Contents

[1 description 4](#_Toc453865255)

[2 SoapUI Tool 4](#_Toc453865256)

[2.1 SoapUI Instalation 4](#_Toc453865257)

[2.2 different Java Vertion for SoapUi 4](#_Toc453865258)

[3 DDT Framework library package 5](#_Toc453865259)

[3.1 Dowload DDT Framework library package 5](#_Toc453865260)

[3.1.1 Clone DDT Framework from GIT Repository 5](#_Toc453865261)

[3.1.2 Update DDT Framework from GIT Repository 6](#_Toc453865262)

[3.1.3 Alternatively way to download DDT Framework 7](#_Toc453865263)

[3.2 Package Structure 8](#_Toc453865264)

[3.3 Build DDT Framework library package 8](#_Toc453865265)

[3.4 Add DDT Framework library to SoapUI 10](#_Toc453865266)

[3.5 Add DDT Framework Plugin to SoapUI 11](#_Toc453865267)

[4 Soap UI Project 12](#_Toc453865268)

[4.1 Dowload SoapUI Project 12](#_Toc453865269)

[4.2 Soapui file encoding configuration 5](#_Toc453865270)

[4.3 Configure SoapUI Project for Testing Environment 13](#_Toc453865271)

[4.4 User-defined Cookie 14](#_Toc453865272)

[4.5 Import SoapUI Project 15](#_Toc453865273)

[4.6 Structure of SoapUI Project 17](#_Toc453865274)

[5 Test Execution and Result Validation 18](#_Toc453865275)

[5.1 Start Test Suite Execution 18](#_Toc453865276)

[5.2 View Test Suite Execution Log 19](#_Toc453865277)

[5.3 View DDT Framework Log 20](#_Toc453865278)

[5.4 Test Step Result Validation 21](#_Toc453865279)

[5.5 Closing SoapUI 25](#_Toc453865280)

[Appendix A. Troubleshooting 26](#_Toc453865281)

[Appendix A.1. Build Task for DDT Framework Failed 26](#_Toc453865282)

[Appendix A.2. Properties from Configuration File is not available in SoapUI Project 26](#_Toc453865283)

[Appendix A.3. Assertion Messages for Failed Steps are too long 26](#_Toc453865284)

[Appendix A.4. Tests are not work or Data-Driver is not works 26](#_Toc453865285)

[Appendix A.5. The “soapui.log” File is Missed 27](#_Toc453865286)

[Appendix A.6. Nothing is helped 27](#_Toc453865287)

# description

This document describes the process of preparing test environment and test execution for DDT Framework.

DDT Framework consists:

* DDT Framework Library
* DDT Framework Plugin for SoapUI
* SoapUI project (with or without custom project properties, custom project properties can be missed in case existing configuration file)
* configuration file (testProps.properties, it can be missed in case existing custom project properties)
* Test Data files

DDT Framework Library and DDT Framework Plugin are created with Groovy language, Spock testing framework and Gradle build automation tool. SoapUI project with regression tests is created and executed via SoapUI Community Edition. Configuration file (testProps.properties) is created by text editor and contains some configurations that are required during tests executions. Test Data files are created by MS Office Excel application and contains test data for data driving testing.

Following steps are required to start using DDT Framework:

* Check SoapUI tool is installed and is configured appropriately
* Check JAVA is installed
* Build DDT Framework library package and add it to SoapUI tool libraries
* Prepare SoapUI project and testProps.properties file

# SoapUI Tool

The SoapUI is the main tool for adding, configuring and executing of test cases. Tests scenarios developed with SoapUI Community version.

For detailed information on SoapUI, please visit the <http://www.soapui.org/> web site.

## SoapUI Instalation

Please use following instructions to install SoapUI:

1. Download the latest version of SoapUI;
2. Open the downloaded file and install SoapUI.
3. The SoapUI Community version is available for download on the official site <http://www.soapui.org/downloads/soapui/open-source.html>.

**Recommendations**: It’s better to install SoapUI on not System driver

**Info**: SoapUI 5.2.1, SoapUI 5.2.0 have been successfully used

## different Java Vertion for SoapUi

SoapUI include ‘jre’ into the installation package by default. It is possible to use different Java versions. DDT Framework Library should work both with Java 7 and 8 versions.

**Recommendations**: It’s better to install JAVA on the workstation where you plan to use DDT Framework

**Info**: JAVA 1.8.0\_60 (64bit) has been successfully used

## Soapui file encoding configuration

To be sure, that all requests containing specific symbols displayed correctly (umlauts for example), it is required to set UTF-8 file encoding format for SoapUI tool.

1. Open the ‘SOAPUI\_HOME/bin’ directory;
2. Open the ‘SoapUI-5.x.x.vmoptions’ file in text editor;
3. Add following line to the ‘SoapUI-5.x.x.vmoptions’;

-Dfile.encoding=UTF-8

1. Save changes.

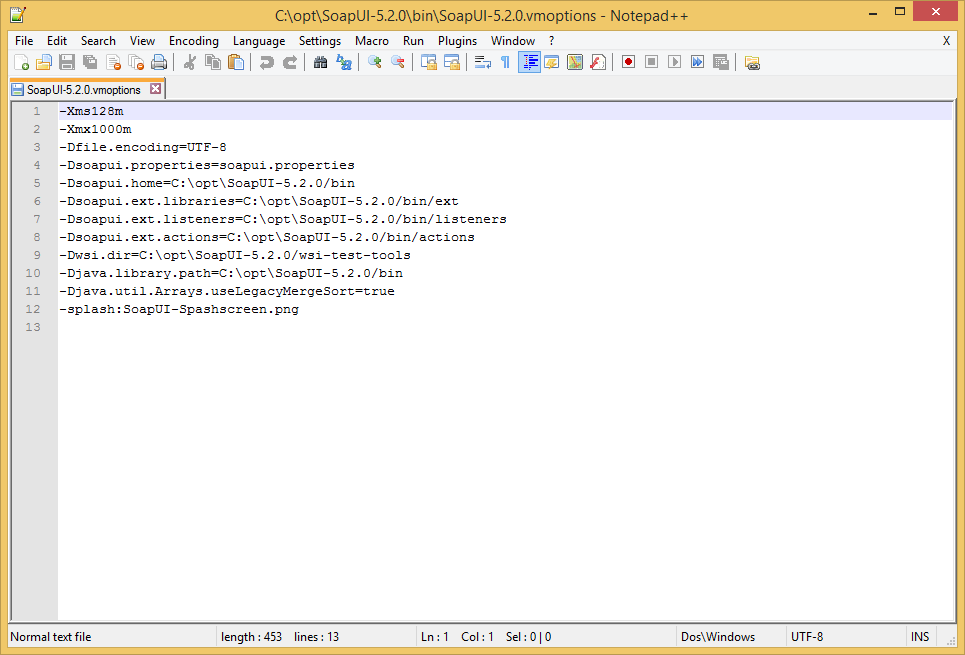


Figure Example of “SoapUI-5.x.x.vmoptions” file

# DDT Framework library package

SoapUI tests use the DDT Framework Library for automation of test data file navigation, configuration of test steps and REST response assertion. It’s developed using the Groovy language, the Gradle build automation tool and the Spock testing framework.

## Dowload DDT Framework library package

### Clone DDT Framework from GIT Repository

The latest version of DDT Framework is available on GIT repository <https://gitbud.epam.com/aleksei_galkin/ddt-framework.git>. It is possible to use the ‘git’ tool to download the DDT Framework library with following commands:

1. Locate to the ‘ddt-framework’ download folder;

cd path\to\ddt-framework\download\folder

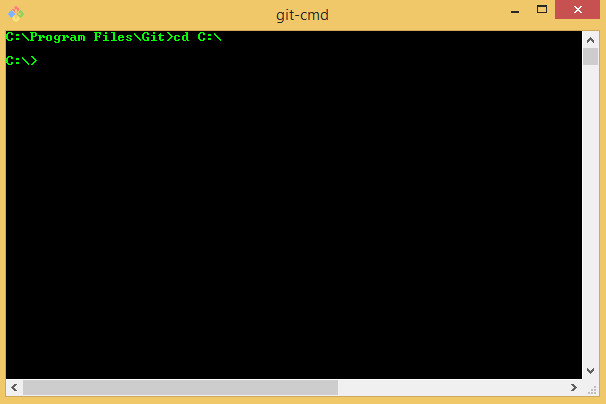


Figure 2 Locate to the ‘ddt-framework’ download folder from Command Prompt on Windows OS

1. Clone the GIT repository: ‘git clone <https://gitbud.epam.com/aleksei_galkin/ddt-framework.git>’.

git clone <https://gitbud.epam.com/aleksei_galkin/ddt-framework.git>

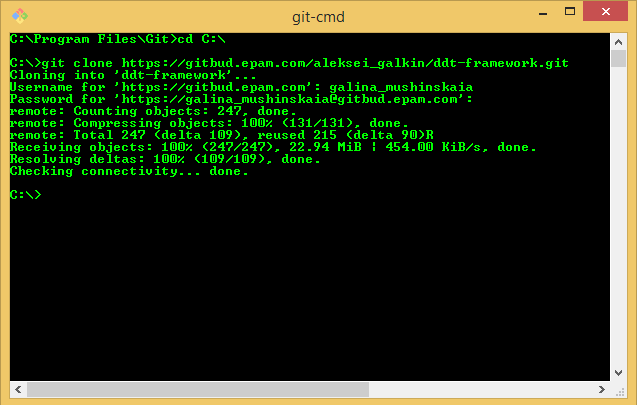


Figure 3 Clone DDT Framework from the GIT repository from Command Prompt on Windows OS

1. Please contact to the DDT Framework Team if access to the GIT repository not granted.

### Update DDT Framework from GIT Repository

Use following commands to update already downloaded DDT Framework from the GIT repository:

1. Locate to the ‘ddt-framework’ folder: ‘cd path\to\ddt-framework\folder’;

cd path\to\ddt-framework\folder

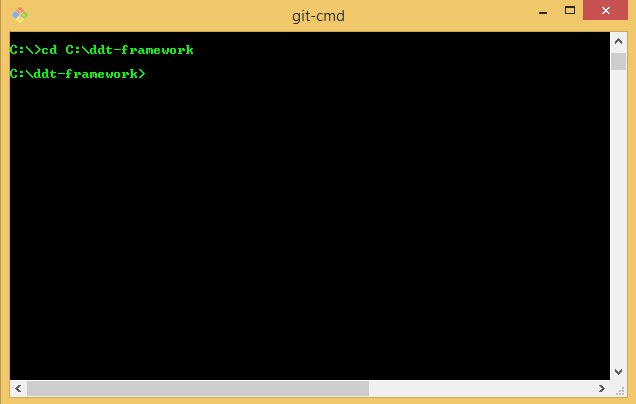


Figure 4 Locate to the ‘ddt-framework’ folder from Command Prompt on Windows OS

1. Use pull command to get the latest updates from GIT repository: ‘git pull’.

git pull

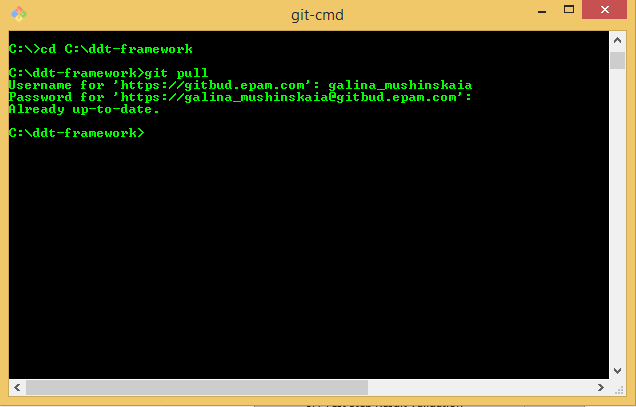


Figure 5 Pull updates for DDT Framework from the GIT repository from Command Prompt on Windows OS

1. Please contact to the DDT Framework Team if access to the GIT repository not granted.

### Alternative way to download DDT Framework

Alternatively, it is possible to download the zip archive directly from the EPAM GITLAB site <https://gitbud.epam.com/aleksei_galkin/ddt-framework/tree/master>:

1. Click on the “Download zip” button to download the latest version of DDT Framework package;

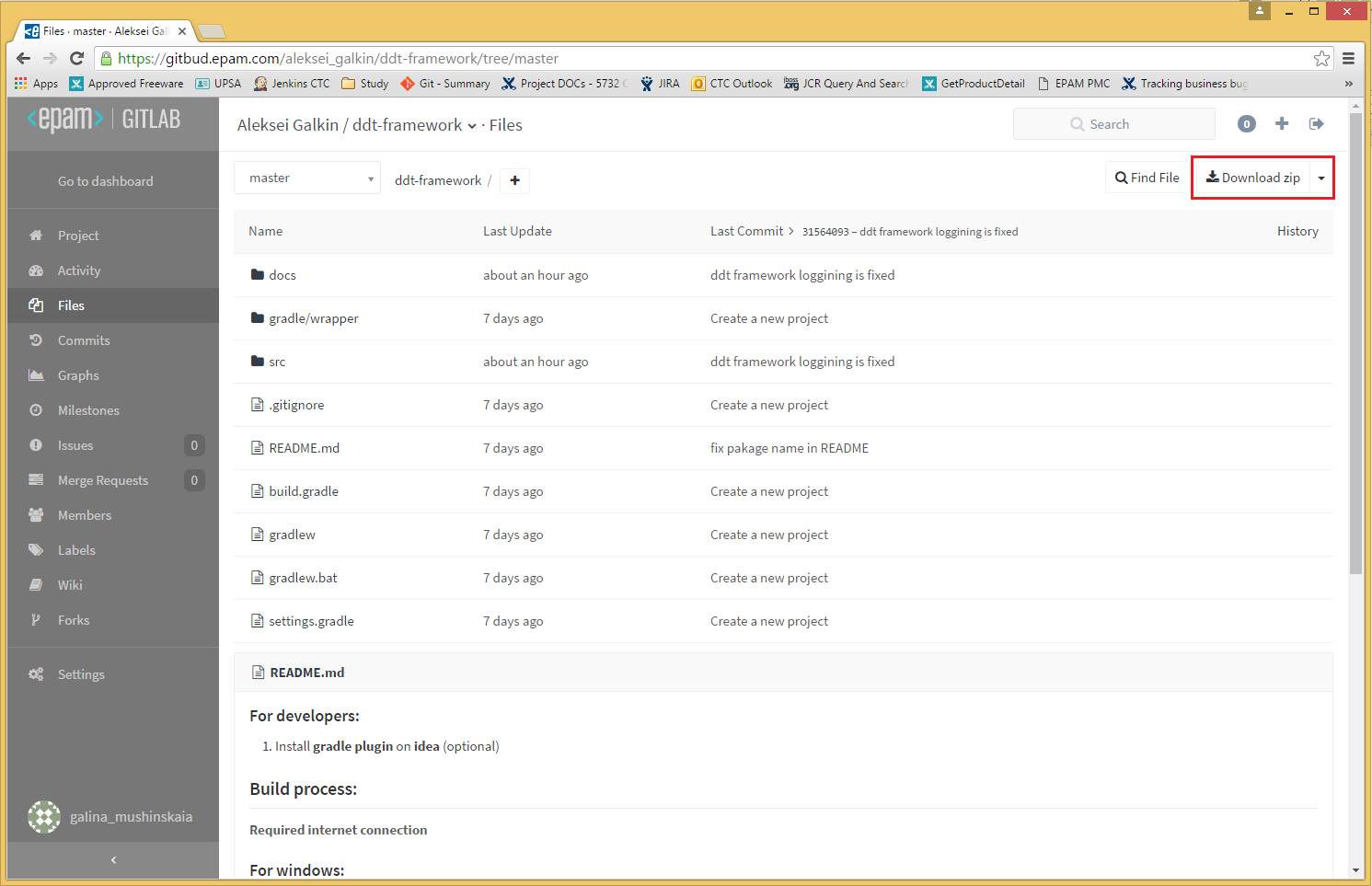


Figure 6 Download zip archive with DDT Framework from the EPAM GITLAB site

1. Extract downloaded zip archive.

## Package Structure

The DDT Framework have following structure:

1. ‘/docs’ – folder with library documentation, demo example and templates;
2. ‘/gradle’ – folder with gradle wrapper for build automation;
3. ‘/src’ – folder with sources and tests for library;
4. ‘build.gradle’ – gradle script with build configuration;
5. ‘gradlew’ and ‘gradlew.bat’ – gradle wrappers for Linux shell and Windows command line to build project without gradle installation;
6. ‘settings.gradle’ – settings for build automation tool; and
7. ‘README.md’ – basic instruction to run build automation tool.

## Build DDT Framework library package

The DDT Framework use Gradle build automation tool. To build DDT Framework package need to perform following steps:

1. Open the “Command Prompt” (use the “Win+R” key combination and run “cmd” command for Windows);

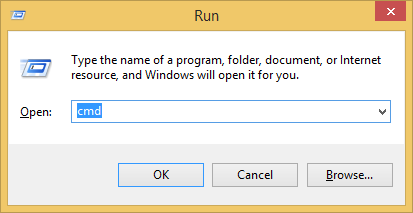


Figure 7 Start “Command Prompt” using the Run command

1. Locate to the ‘ddt-framework’ folder: ‘cd path\to\ddt-framework\folder’;

cd path\to\ddt-framework\folder



Figure 8 Locate to the ‘ddt-framework’ folder from Command Prompt on Windows OS

1. Run the “gradlew.bat build” command line to build the DDT Framework library package.

gradlew.bat build

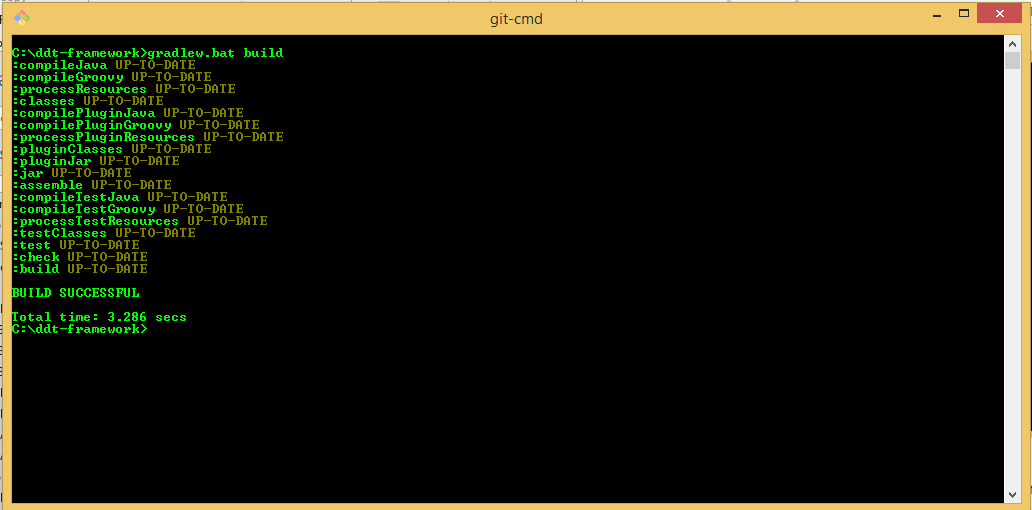


Figure 9 Build DDT Framework Library package from Command Prompt on Windows OS

1. It might take some time for Gradle to download and resolve all library dependences for the first build time.
2. The JAR file(-s) should be found in ‘build\libs’ folder when build task successfully completed.
3. If build process is failed with some error try to use the “gradlew.bat jar” command instead to eliminate the library tests execution. If it does not helps please contact to the DDT Framework Team.

gradlew.bat jar

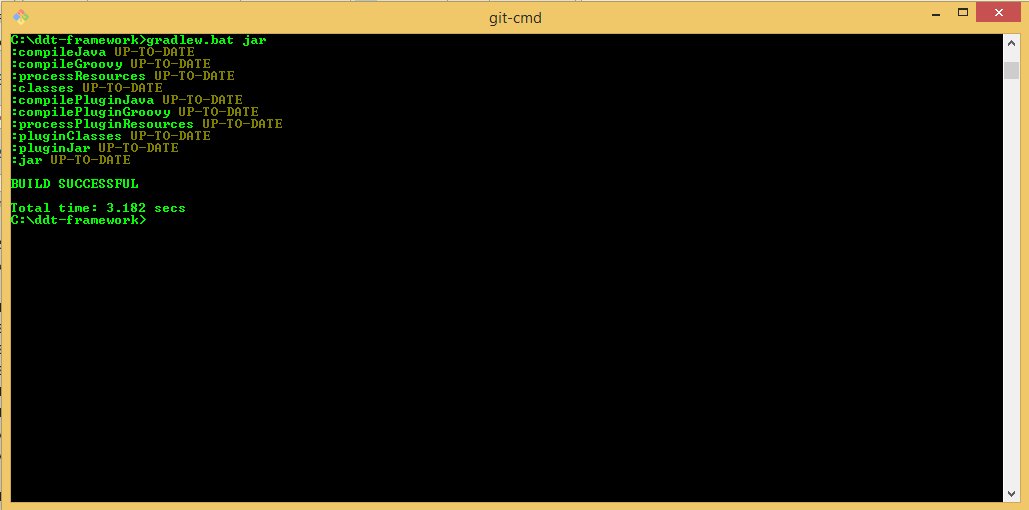


Figure 10 Build DDT Framework Library package and skip tests from Command Prompt on Windows OS

1. Please see the “EPM-SPI\_DDT\_Framework\_Overview.docx” document from ‘docs’ folder of the for more details on DDT Framework Library build process.

## Add DDT Framework library to SoapUI

To make DDT Framework library available in SoapUI please copy JAR file with DDT Framework library to the ‘SOAPUI\_HOME/lib’ folder where SOAPUI\_HOME is the installation folder of SoapUI application.

**Note**: DDT Framework library JAR file usually named like ddt-framework-<version number>.jar

1. When DDT Framework Library build process is successfully completed open the ‘ddt-framework\build\libs’ folder;

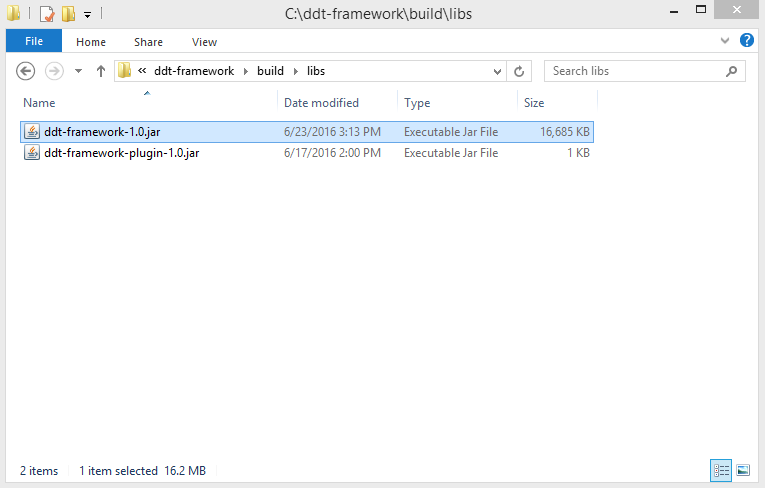


Figure 11 Build folder with DDT Framework package

1. Copy the ‘ddt-framework-1.0.jar’ file to the ‘SOAPUI\_HOME/lib’ folder where SOAPUI\_HOME is the installation folder of SoapUI application.

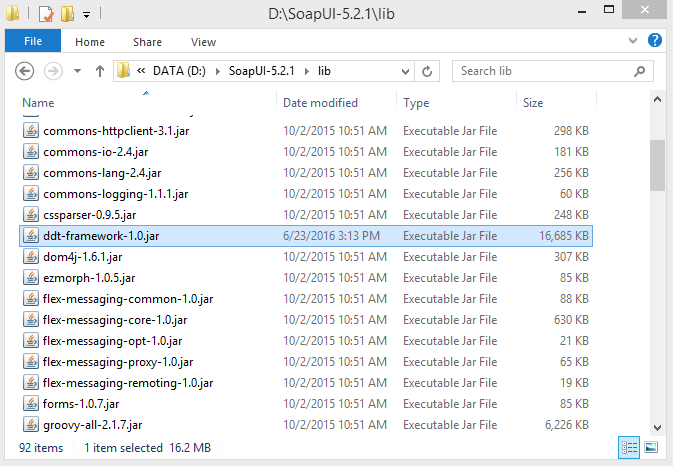


Figure 2 SoapUI ‘lib’ folder with DDT Framework Library package

1. Please see the “EPM-SPI\_DDT\_Framework\_Overview.docx” document from ‘docs’ folder of the for more details on integration of DDT Framework Library to SoapUI project.

## Add DDT Framework Plugin to SoapUI

To make DDT Framework plugin available in SoapUI please copy JAR file with DDT Framework Plugin to the ‘SOAPUI\_HOME/bin/plugins’ folder where SOAPUI\_HOME is the installation folder of SoapUI application.

Note: DDT Framework plugin JAR file usually named like ddt-framework-plugin-<version number>.jar

1. When DDT Framework build process is successfully completed open the ‘ddt-framework\build\libs’ folder;

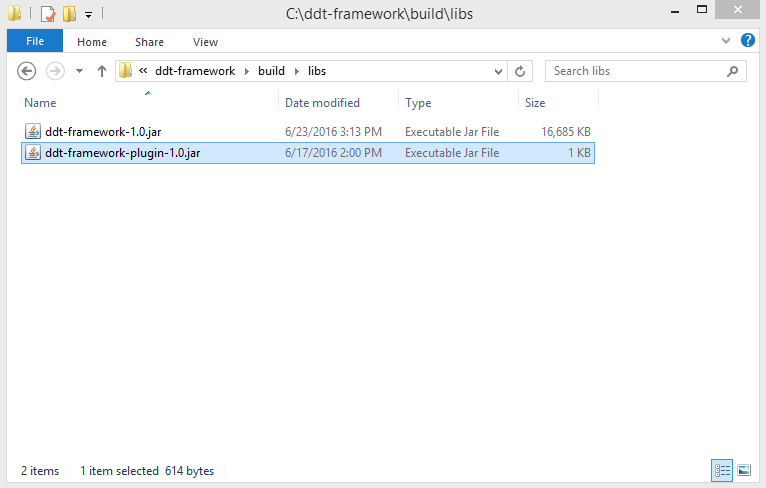


Figure 3 Build folder with DDT Framework

1. Copy the ‘ddt-framework-plugin-1.0.jar’ file to the ‘SOAPUI\_HOME/bin/plugins’ folder where SOAPUI\_HOME is the installation folder of SoapUI application. If the ‘plugins’ folder does not exist under the ‘SOAPUI\_HOME/bin’ please create it.

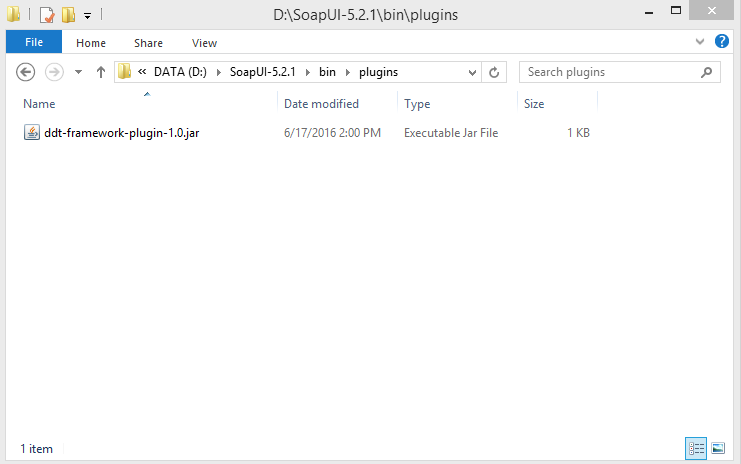


Figure 4 SoapUI ‘plugins’ folder with DDT Framework Plugin

1. Please see the “EPM-SPI\_DDT\_Framework\_Overview.docx” document from ‘doc’ folder of the for more details on integration of DDT Framework Plugin to SoapUI project.

# SoapUI Project

SoapUI is the main tool for adding, configuring and executing of test cases.

Soap UI project contains set of test cases for regression testing. There is being configured path to test data files, endpoints, REST Requests, Groovy Scripts and Assertion Scripts with DDT Framework Library methods calls, and Manual Steps if it is required.

The DEMO of SoapUI project is available on GIT repository <https://gitbud.epam.com/aleksei_galkin/ddt-framework/tree/master/> in following folder ‘\docs\examples\demo’

The template of SoapUI project is available on GIT repository <https://gitbud.epam.com/aleksei_galkin/ddt-framework/tree/master/> in following folder ‘\docs\examples\soap-ui-tests’.

## DowNload SoapUI Project

1. Please make sure you use the latest version of (template) SoapUI project from appropriate git branch. Please contact the DDT Framework team to get the correct git branch name.

The Demo-DDTF-soapui-project.xml of SoapUI project is available on GIT repository <https://gitbud.epam.com/aleksei_galkin/ddt-framework.git> in following folder: ‘\docs\examples\demo’

The template DDT-Framework-soapui-project.xml of SoapUI project is available on GIT repository <https://gitbud.epam.com/aleksei_galkin/ddt-framework.git> in following folder: ‘\docs\examples\soap-ui-tests’.

It is possible to use the ‘git’ tool to download SoapUI project, or download the zip archive directly from the EPAM GITLAB site <https://gitbud.epam.com/aleksei_galkin/ddt-framework/tree/master/docs/examples/soap-ui-tests> (please use <https://gitbud.epam.com/aleksei_galkin/ddt-framework/tree/master/docs/examples/demo> to get DEMO):

1. Click on the “Download zip” button to download the latest version of (template or demo) SoapUI project and test data:

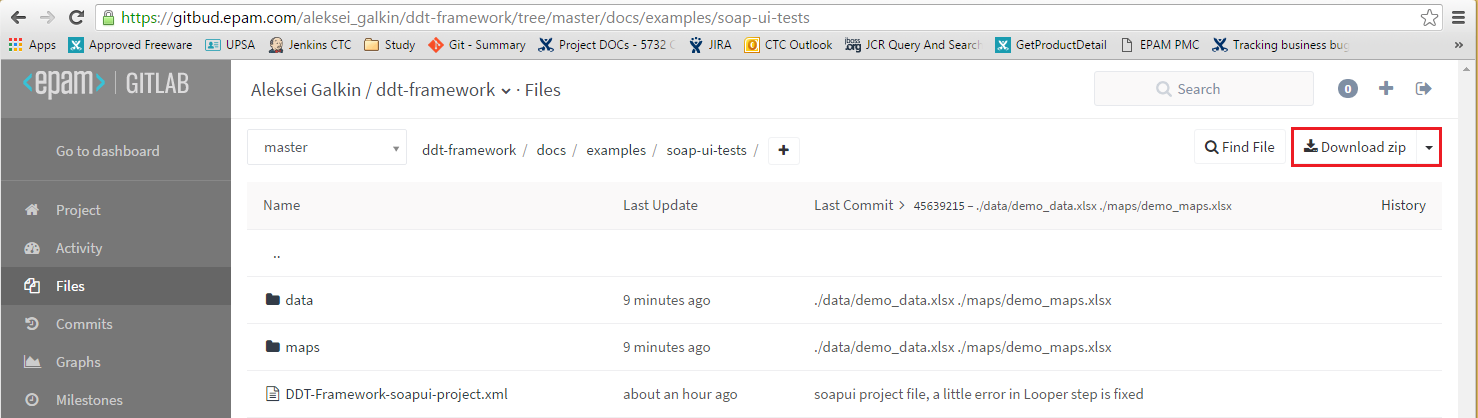


Figure 5 Download zip archive with SoapUI project from the EPAM GITLAB site

1. Extract downloaded zip archive.

## Configure SoapUI Project for Testing Environment

Configuring SoapUI project for testing environment is usually done by configuration file. Its name is kept in DDT Framework code and at the moment it should be ‘testProps.properties’. The ‘\docs\examples\soap-ui-tests\testProps.properties’ file contains list of properties for easy configuration SoapUI project:

1. Open the ‘\docs\examples\soap-ui-tests’ folder (or ‘\docs\examples\demo’ for DEMO) with SoapUI tests from the extracted archive;

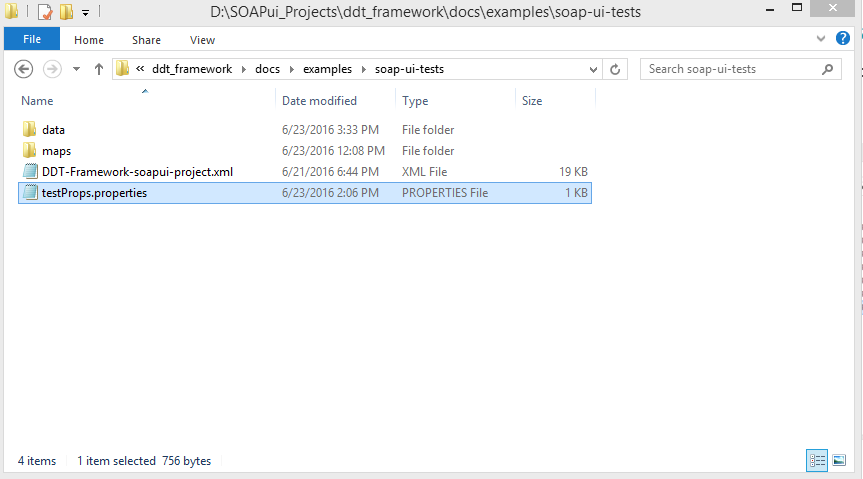


Figure Folder with SoapUI project

1. Open the ‘testProps.properties’ file in a text editor;

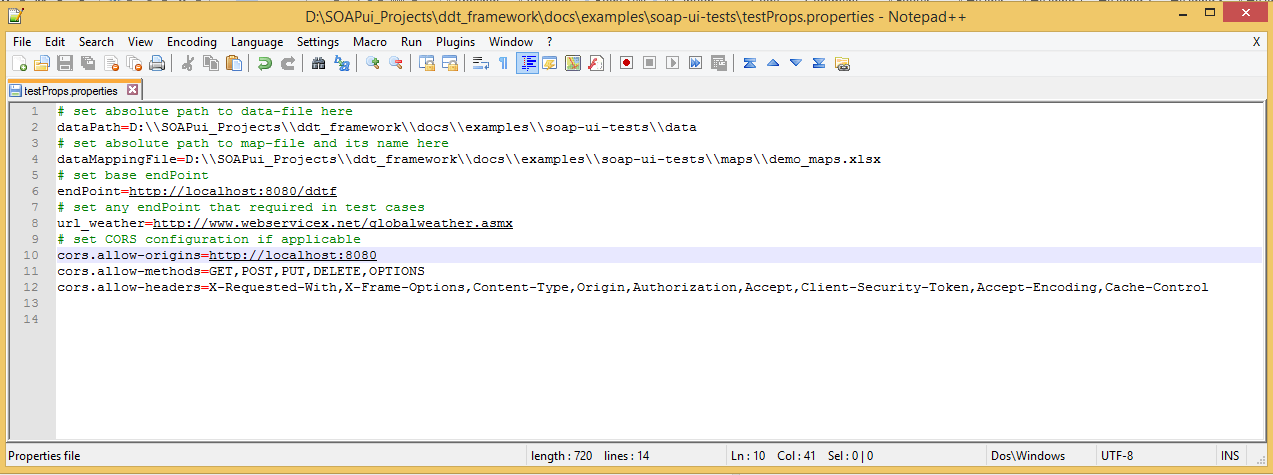


Figure Configuration file for SoapUI project

1. Update property values according the following table;

Table List of configuration properties for SoapUI project

| Configuration properties for SoapUI project | | |
| --- | --- | --- |
| Property name | Description | Example |
| dataPath | Path to directory with data files where “PATH\_TO\_EXTRACTED\_FOLDER” is path to folder where ‘soap-ui-tests’ was extracted (please use double backslashes as path separator ‘\\’). | PATH\_TO\_EXTRACTED\_FOLDER\\docs\\examples\\soap-ui-tests\\data |
| dataMappingFile | Path to file with the variables values used in data files where ‘PATH\_TO\_EXTRACTED\_FOLDER’ is path to folder where ‘soap-ui-tests’ was extracted (please use double backslashes as path separator ‘\\’). | PATH\_TO\_EXTRACTED\_FOLDER\\docs\\examples\\soap-ui-tests\\maps\\demo\_maps.xls |
| endPoint | Services layer server. Please use the ‘http://server1.epam.com:170’ for application deployed on ‘server1.epam.com’ server. | http://server1.epam.com:170 |
| url\_weather | Any service layer server. For example public web weather server | http://www.webservicex.net/globalweather.asmx |

1. Save all changes in the “testProps.properties” file.
2. The configuration file for SoapUI project should be located at the same directory as project itself.
3. Please see the “EPM-SPI\_DDT\_Framework\_Overview.docx” document from ‘docs’ folder of the for more details on configuration of SoapUI project.

## User-defined Cookie

1. This feature works only with DDT Framework Plugin.
2. The ‘testProps.properties’ configuration file for SoapUI project should be located at the same directory as project itself.

It is possible to add user-defined cookies to HTTP request with DDT Framework Plugin. If the ‘cookie’ property is defined in the ‘testProps.properties’ configuration file then it value will be added to HTTP request cookies.

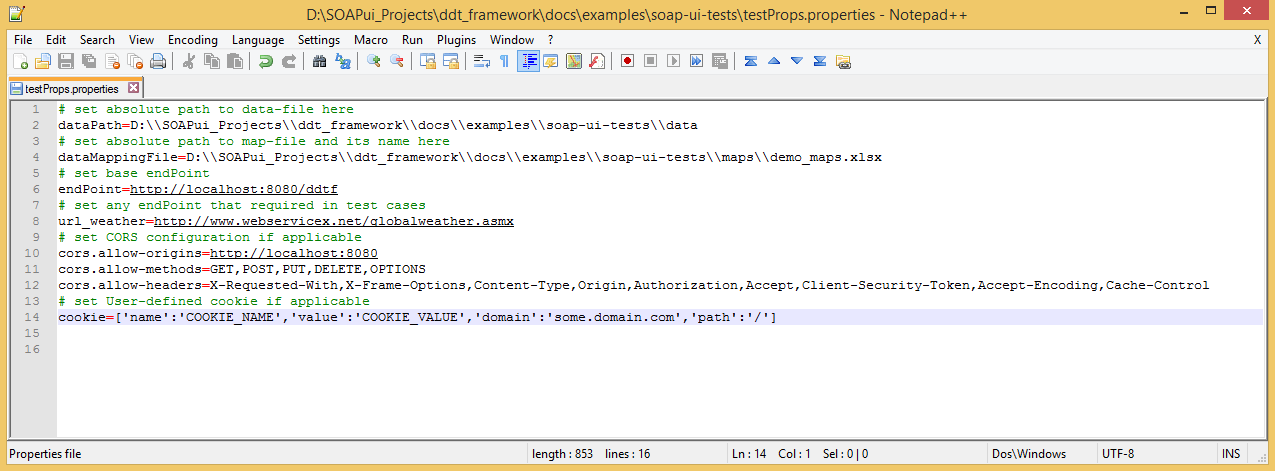


Figure Example of test configuration with user-defined cookie

## Import SoapUI Project

It is required to import the SoapUI project to the SoapUI workspace for the first time:

1. Open the installed SoapUI tool;

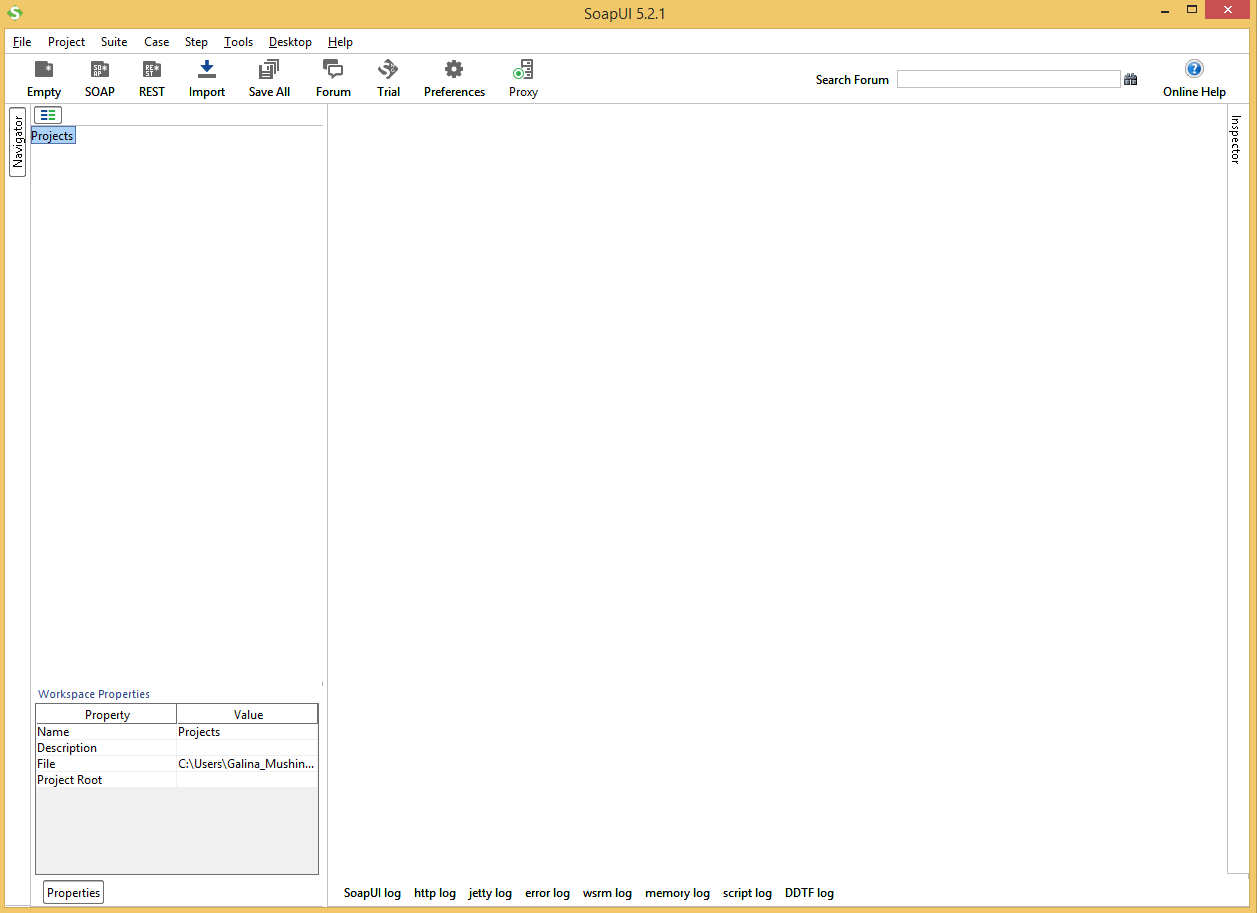


Figure SoapUI main window

1. Click on the “Import Project” item of “File” menu;

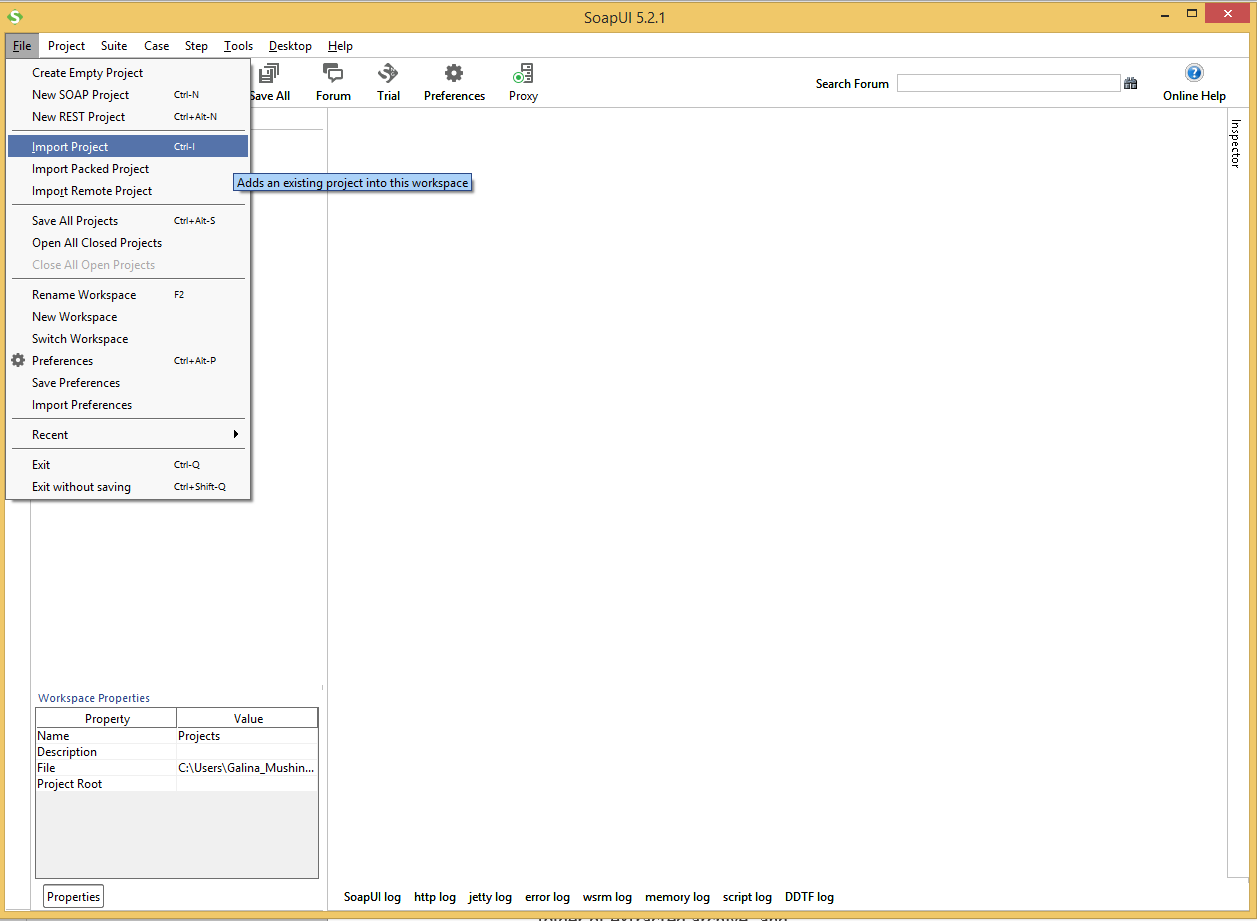


Figure “Import Project” menu item

1. Select the “DDT-Framework-soapui-project.xml” file (or Demo-DDTF-project.xml for DEMO) from the ‘\docs\examples\soap-ui-tests’ (or ‘\docs\examples\demo’ for DEMO) folder of extracted archive; and

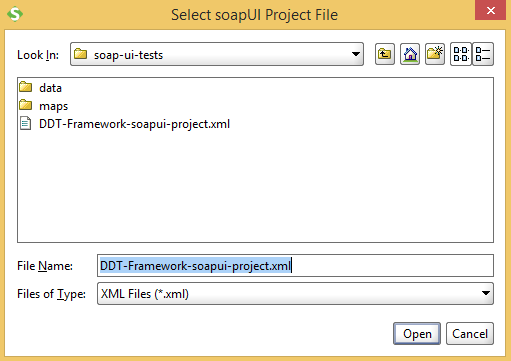


Figure “Select soapUI Project File” dialog box

1. Click on the “Open” button of the “Select soapUI Project File” dialog box.

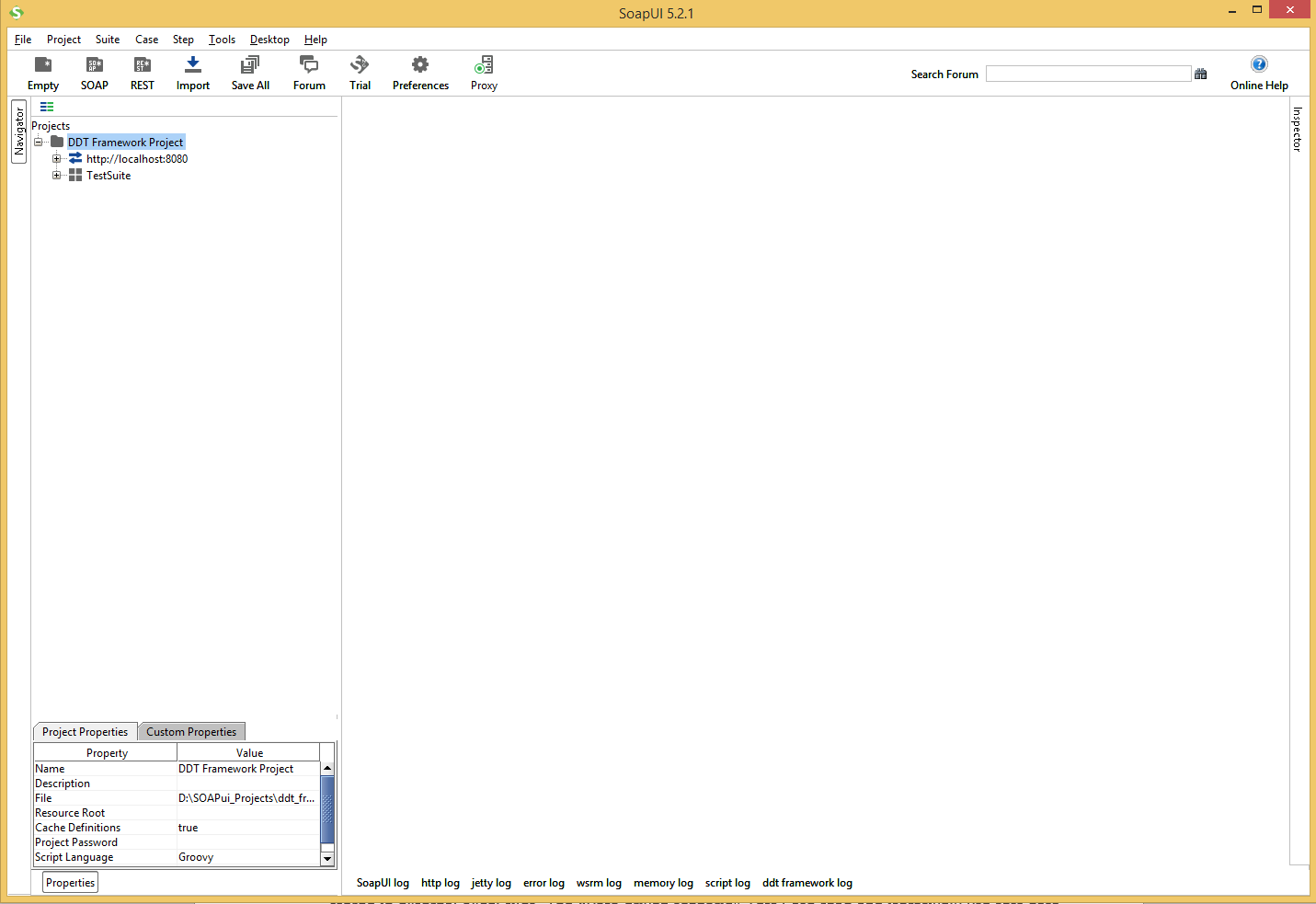


Figure View of imported SoapUI project

1. For detailed information about how to work in SoapUI, please visit the <http://www.soapui.org/> web site.

## Structure of SoapUI Project

The SoapUI project for DDT consists set of test suites. Each test suite linked to one data-file by test suite custom property ‘dataFile’ that contains test case(-s)/test scenario(-s) for story, task or bug as it’s declared in the name of test suite.

Test automation is performed with data-driven testing. Test data (input, expected output, etc.) are stored in external Excel files. The “Data driven scenario” Test Case read and iteratively use test data. Input data may contain variables defined in separate file (usually called ‘maps’) to eliminate massive update of test data files in case some systems changes. In this situation there will be need to update only the variables values or specify another file with the variables values.

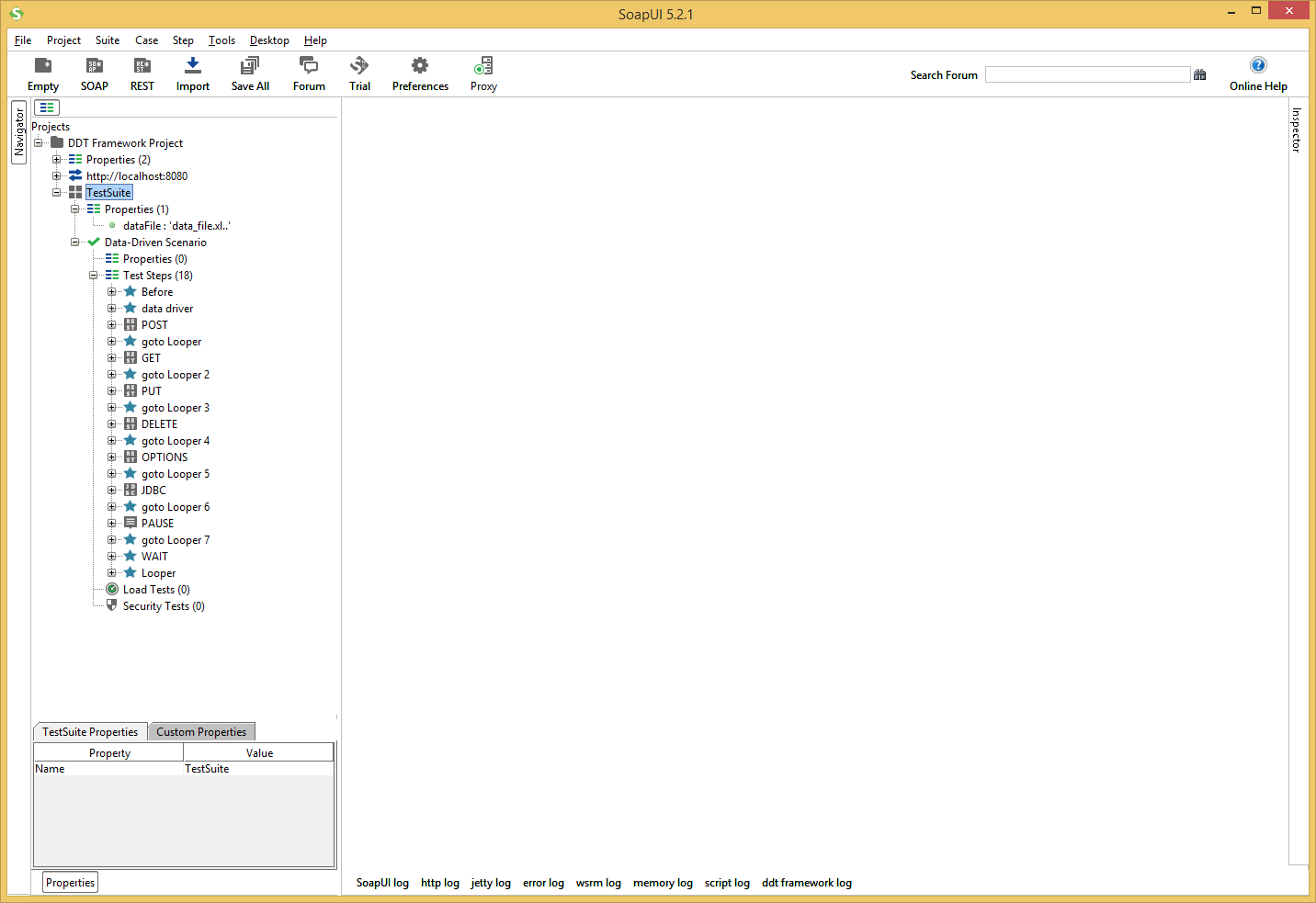


Figure SoapUI test case structure

1. Please see the “EPM-SPI\_DDT\_Framework\_Overview.docx” document from ‘docs’ folder of the for more details on structure of SoapUI project.

# Test Execution and Result Validation

Before test execution, make sure that DDT Framework library integrated to SoapUI and SoapUI Project configured properly.

## Start Test Suite Execution

It is possible to execute any test suite according to the following instruction:

1. Expand project tree in SoapUI;
2. Double click on selected Test Suite to open Test Cases window;

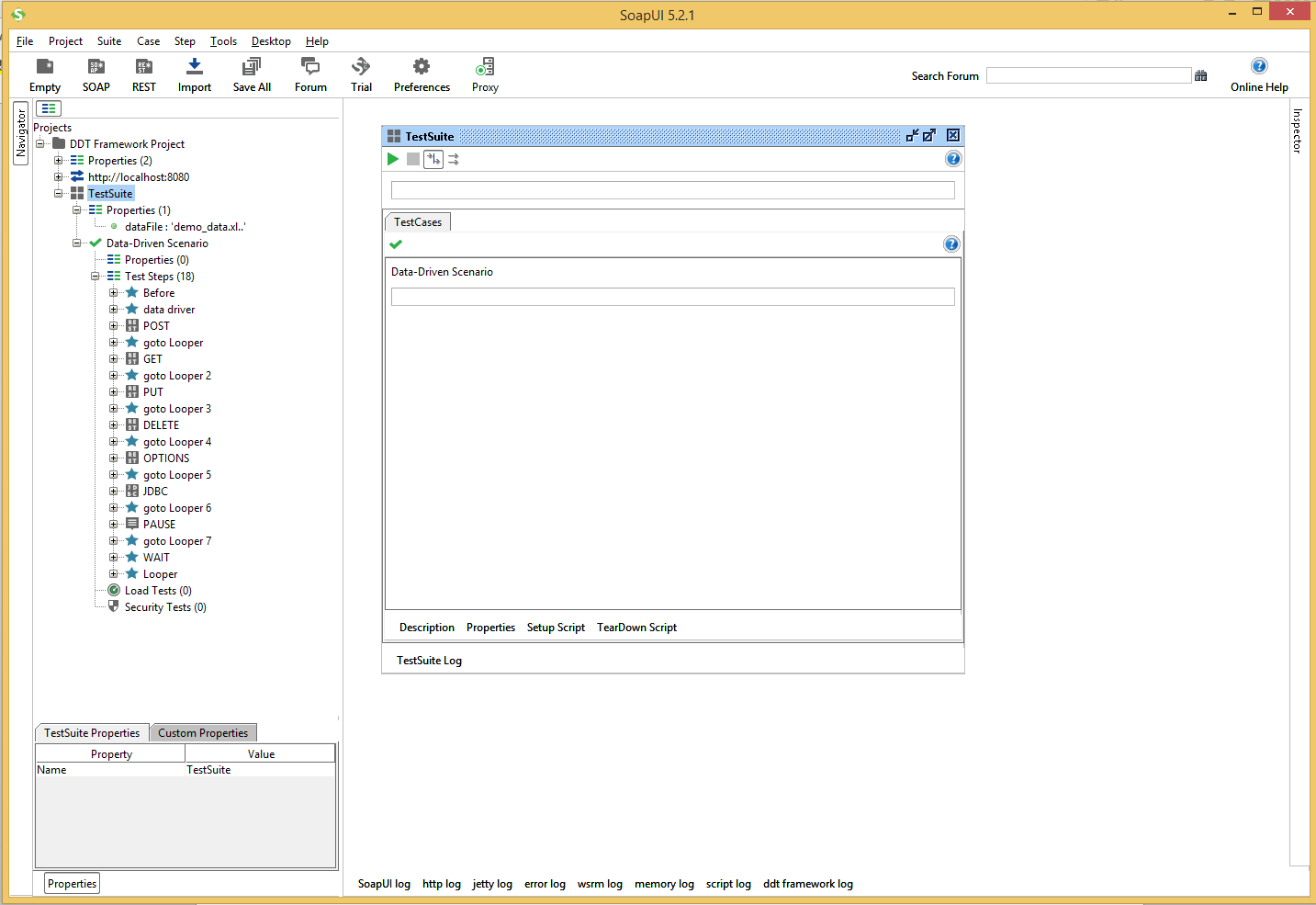


Figure TestSuite window

1. Click on the ‘Runs the selected TestCases’ button (green arrow) to start Test Suite execution.

## View Test Suite Execution Log

All information on status of TestCases execution is available in ‘TestSuite Log’ at bottom of the Test Cases window.

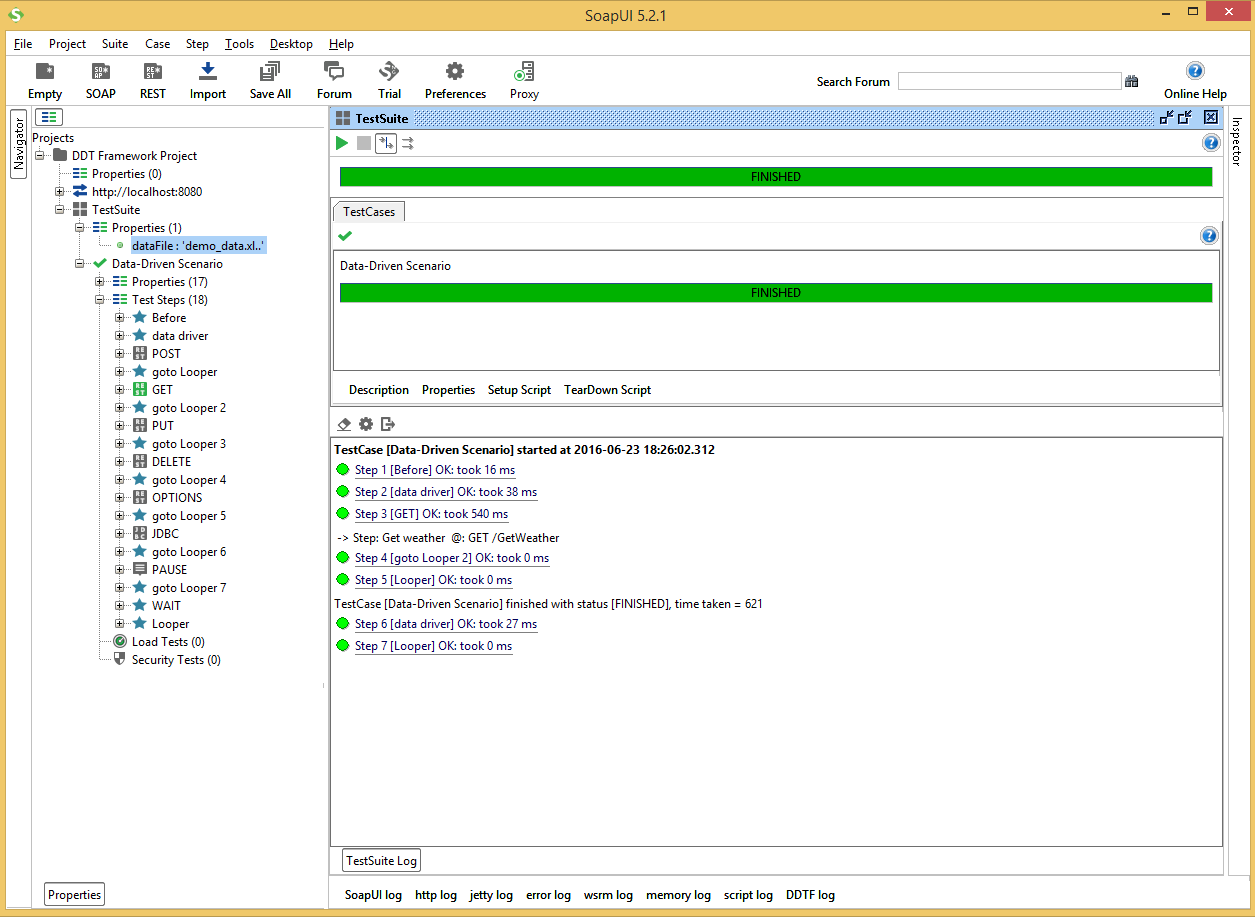


Figure TestSuite Execution Log

## View DDT Framework Log

All information on test data, test step parameters and result assertion is logged by the DDT Framework Library and available in ‘DDTF log’ tab at bottom of the SoapUI window.

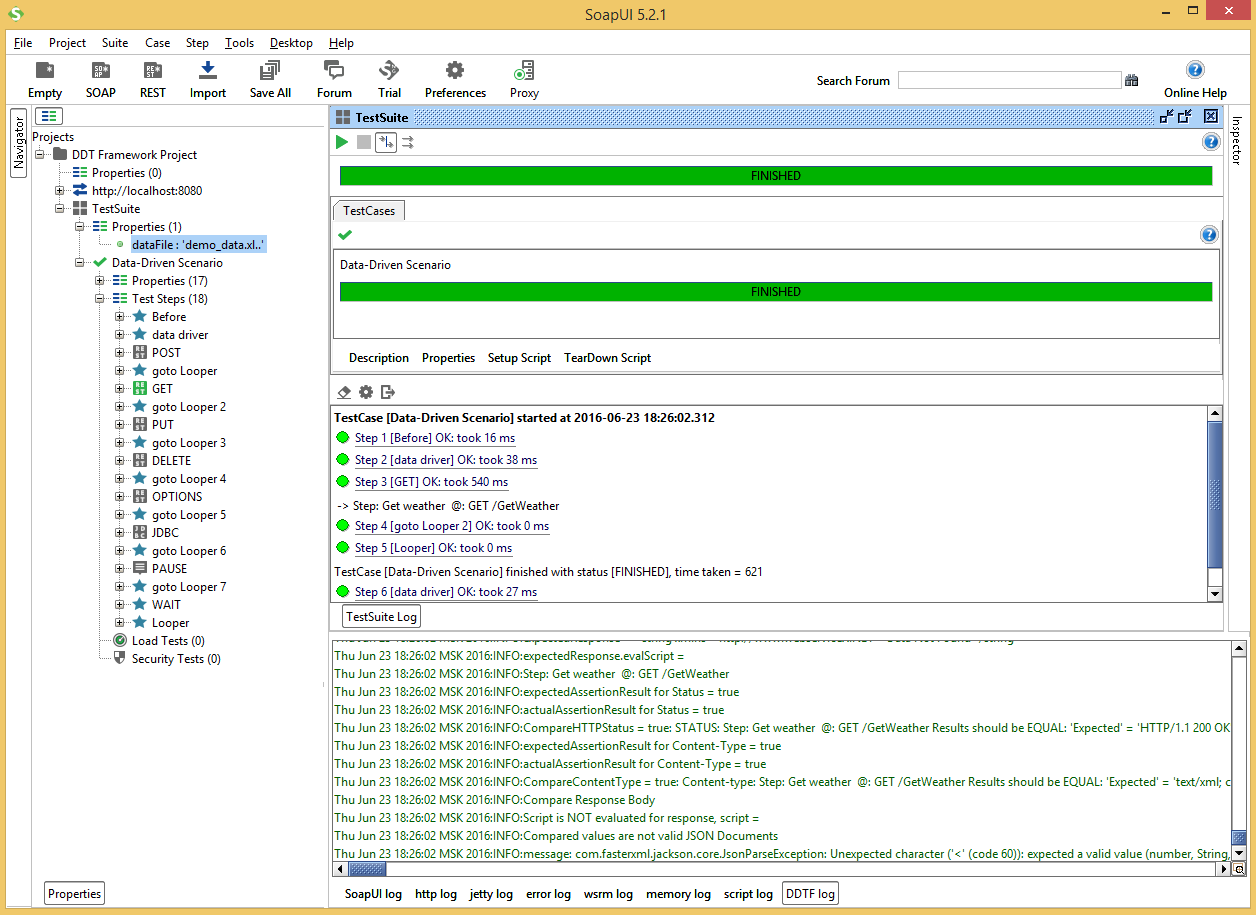


Figure DDT Framework Library Log

1. It is possible to configure the maximum number of rows displayed in ‘DDTF log’ with the “Set Max Rows” menu item of context menu from the ‘DDTF log’ tab.

## Test Step Result Validation

Test suite log contains results for each executed test step. All passed steps marked with green colour and all failed steps marked with red colour.

1. It is possible to open each REST request step with double click to view its Request Message, Response Message and Properties.

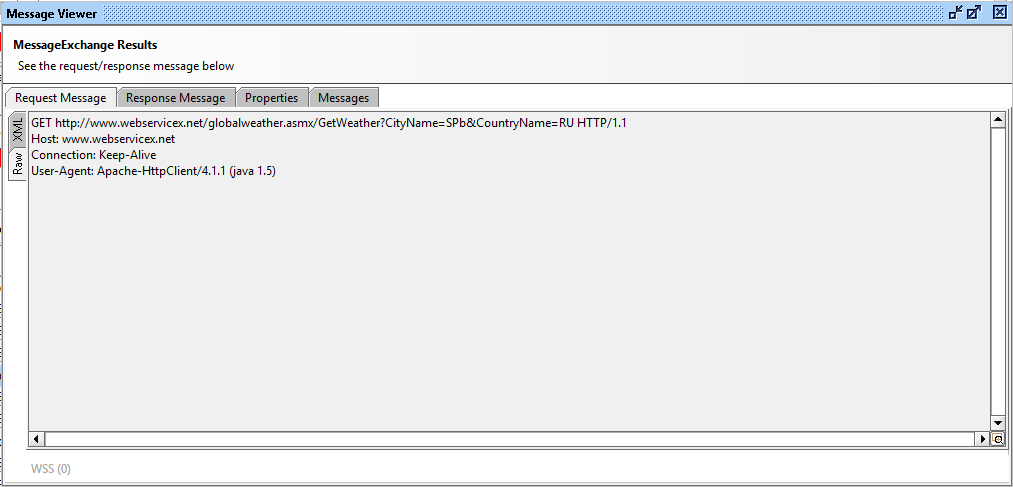


Figure Message Viewer window

1. All failed REST requests contain assertion message.

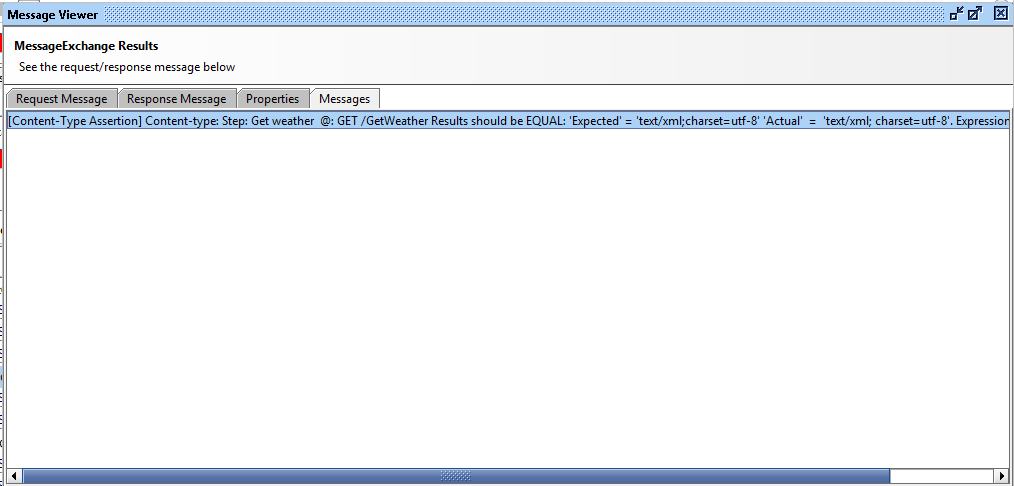


Figure Messages tab with assertion messages

1. It is possible to copy the assertion message from the “Message Viewer” dialog with the “Ctrl+C” keys combination and paste it to a text editor. It is more convenient to analyze of assertion result in test editor. The “Message Viewer” dialog in SoapUI is not support text formatting and indentations for the assertion message.

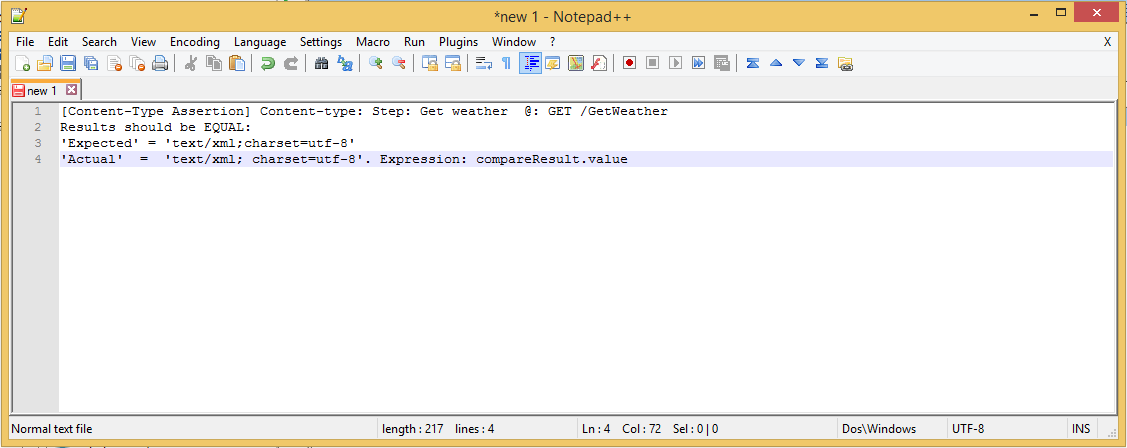


Figure Assertion message in text editor

1. All failed test steps contains step description. Step description start with combination of CELL\_ DESCRIPTION, CELL\_METHOD and CELL\_PATH.

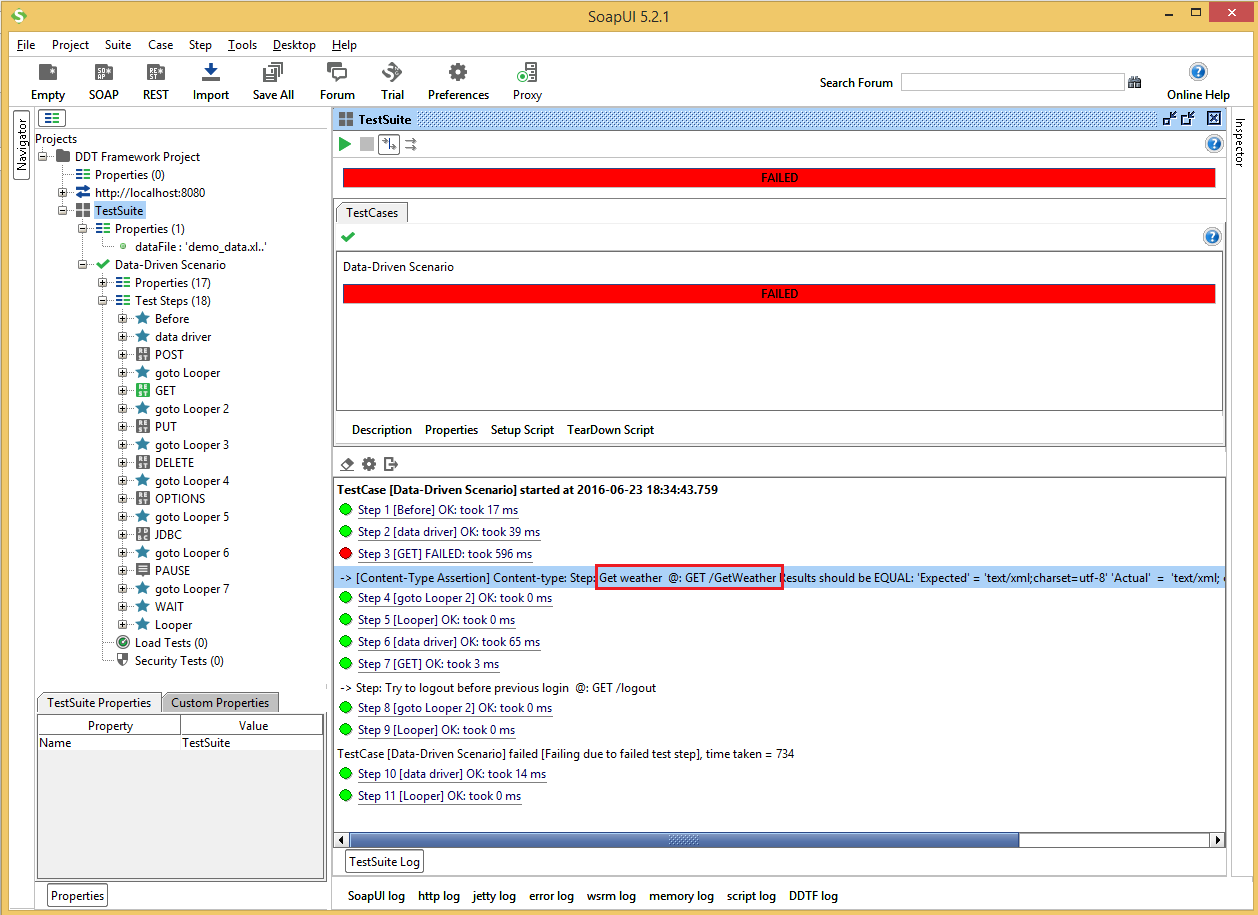


Figure Description of failed test step in TestSuit log

1. All assertion messages start with the same combination of CELL\_ DESCRIPTION, CELL\_METHOD and CELL\_PATH for efficient navigation on test data file.

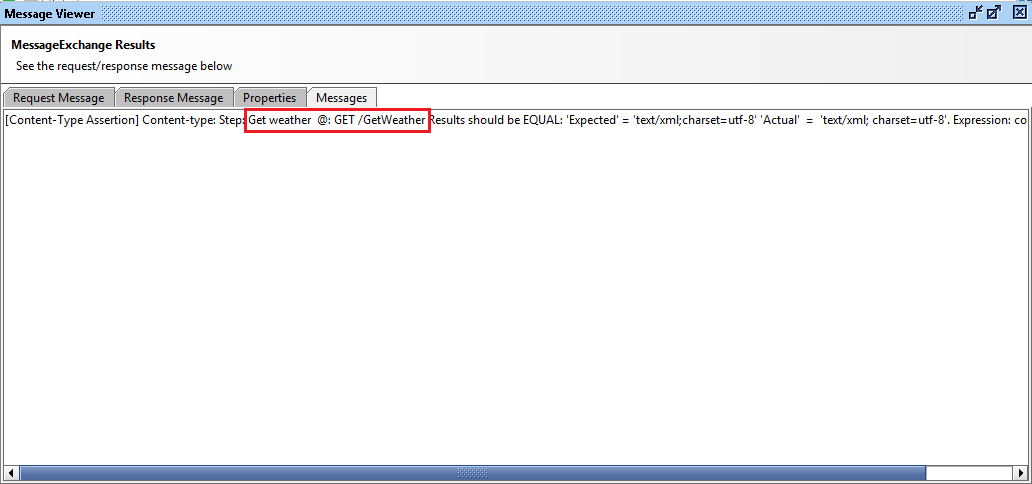


Figure Description of failed test step in assertion message

1. It is possible to verify corresponding test step in test data file using this combination with the appropriate data file.

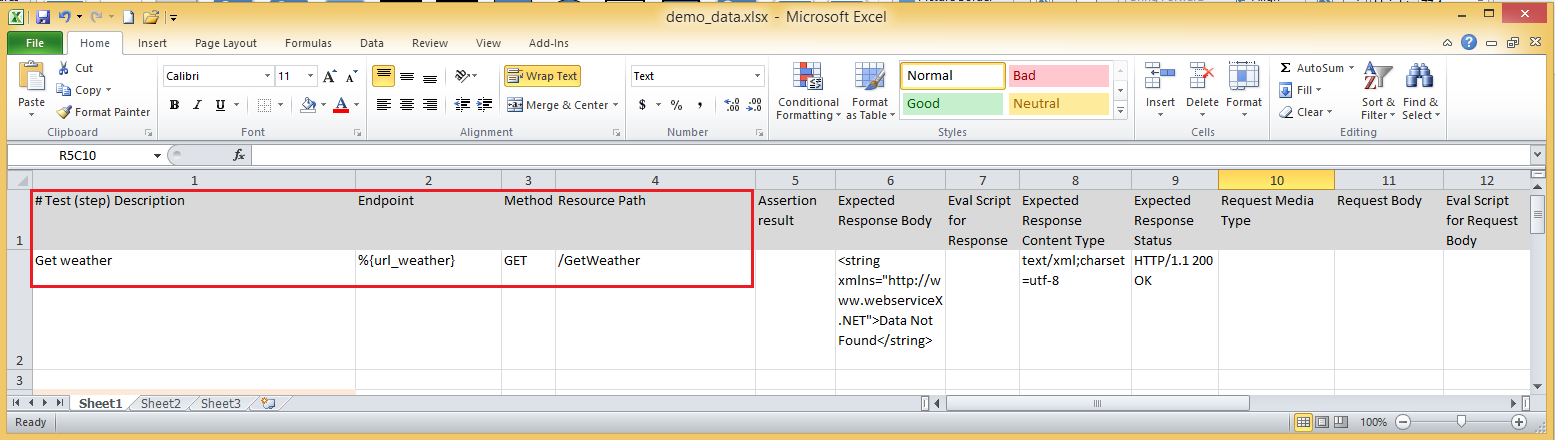


Figure Combination of CELL\_ DESCRIPTION, CELL\_METHOD and CELL\_PATH in Test Data file

1. Each test data file can contain description of all test scenarios on the “TestCases” sheet of document.

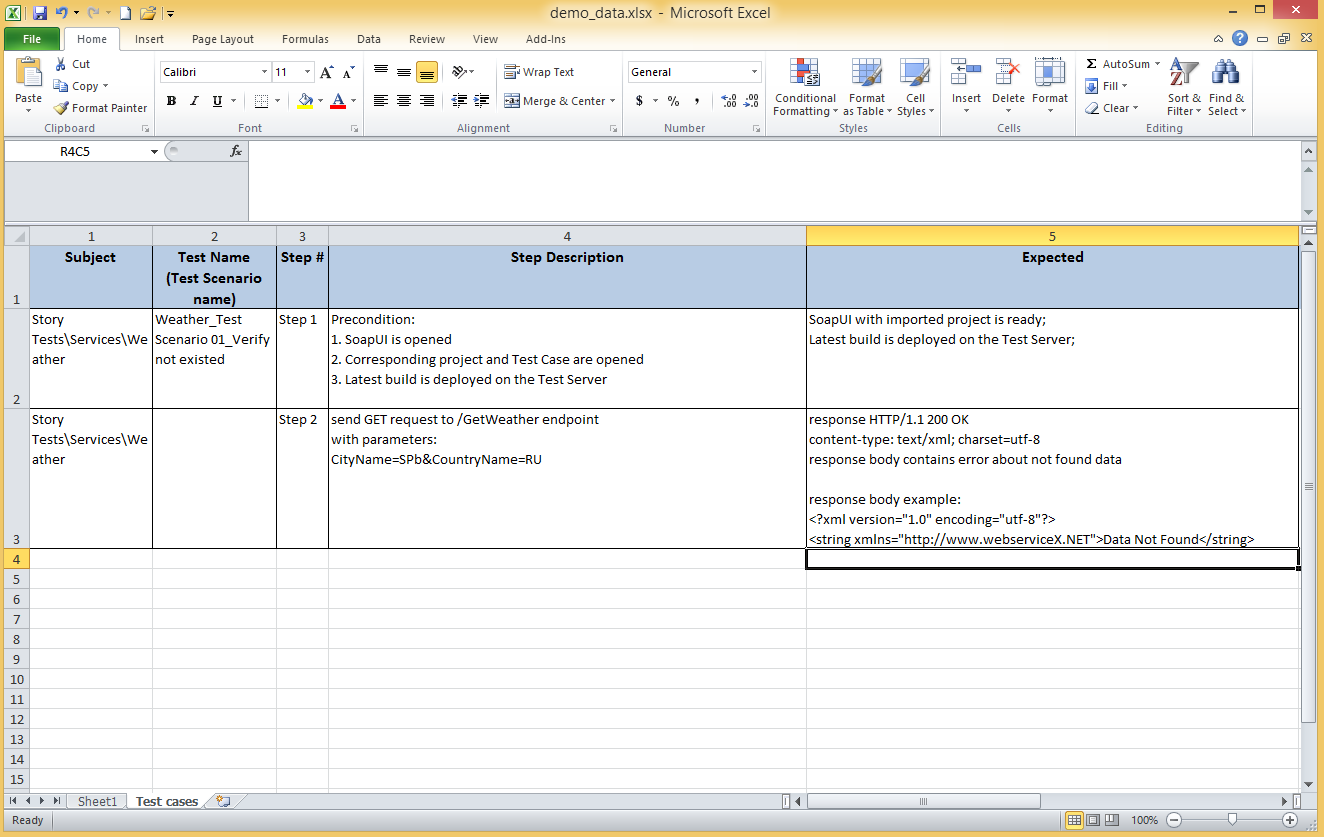


Figure Test scenarios on the “TestCases” sheet of document in Test Data file

1. Test step description contains reference to appropriate test scenario, e.g. “TS 04\_...” means “Test Scenario 04, Step …”.
2. The “Step Description” and “Expected” fields on the “TestCases” sheets might by updated with short delay. Please use the test data from the first sheet as it contains up to date information.

It is possible to view information on REST request step in different formats for more convenient step validation.

Table Tabs and message formats on Message Viewer window

| Available message formats | |
| --- | --- |
| Message tab | Message format |
| Request Message | * XML * RAW |
| Response Message | * XML * JSON * HTML * RAW |
| Step properties | * SoapUI table |
| Assertion message | * String |

Request and Response Messages in RAW format contain information on HTTP headers.

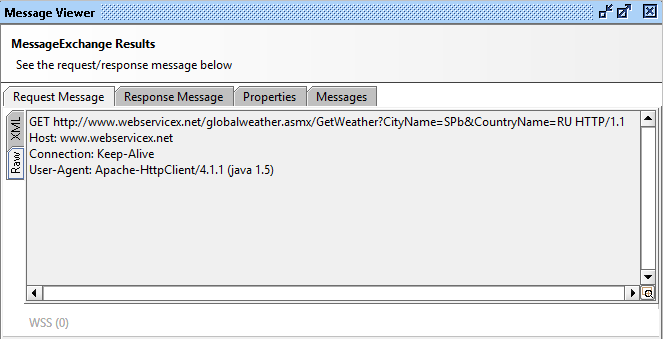


Figure RAW format of Request Message

## Closing SoapUI

There is a common rule about SoapUI – keep your projects ‘closed’ in SoapUI workspace. So, it’s better to close project before closing SoapUI tool. Highlight the project, view context menu by right mouse click and choose ‘Close Project’ menu item. You can also close SoapUI without saving changes. To do that please click on the “File / Exit without saving” menu item.

1. Troubleshooting

This chapter contains the list of known troubles that may occurs during DDT Framework installation.

* 1. Build Task for DDT Framework Failed

If build process is failed with some error at *“:test”* point try to use the “gradlew.bat jar” command instead to eliminate the library tests execution.

1. If it does not helps please contact to the DDT Framework Team for further support.
   1. Properties from Configuration File is not available in SoapUI Project

Please make sure that configuration file is located at the same folder as the SoapUI project file itself. The DDT Framework Library searches configuration file in directory where SoapUI project file located.

If project file has moved from other directory, please make sure it is re-import successfully into SoapUI. SoapUI has link to the project file in its workspace so it is required to re-import all moved project files to update link in SoapUI workspace or it is possible to switch to the new workspace with imported file.

* 1. Assertion Messages for Failed Steps are too long

The “Message Viewer” dialog in SoapUI is not support text formatting and indentations for the assertion message. However, it is possible to copy the assertion message from the “Message Viewer” dialog with the “Ctrl+C” keys combination and paste it to the text editor. It is more convenient to analyze of assertion result in test editor.

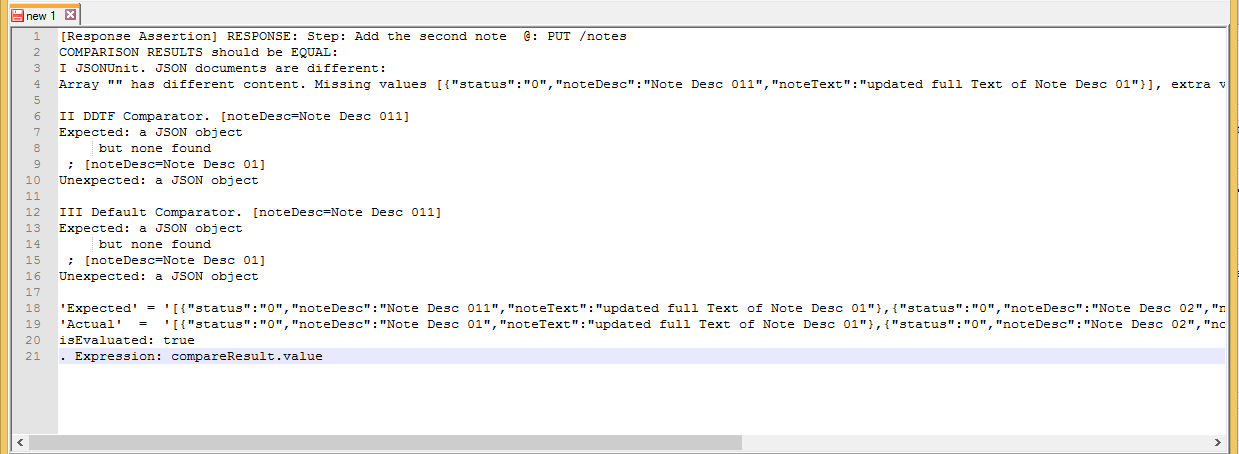


Figure 5 Assertion message in text editor

* 1. Tests are not work or Data-Driver is not works

It is possible to investigate SoapUI log file to figure out what goes wrong. SoapUI creates “soapui.log” and “soapui-errors.log” files in the “SOAPUI\_HOME/bin” directory by default. The “soapui.log” contains detailed logs for data-driver runs with test steps descriptions and parameters. The “soapui-errors.log” contains detailed errors stack traces for SoapUI errors.

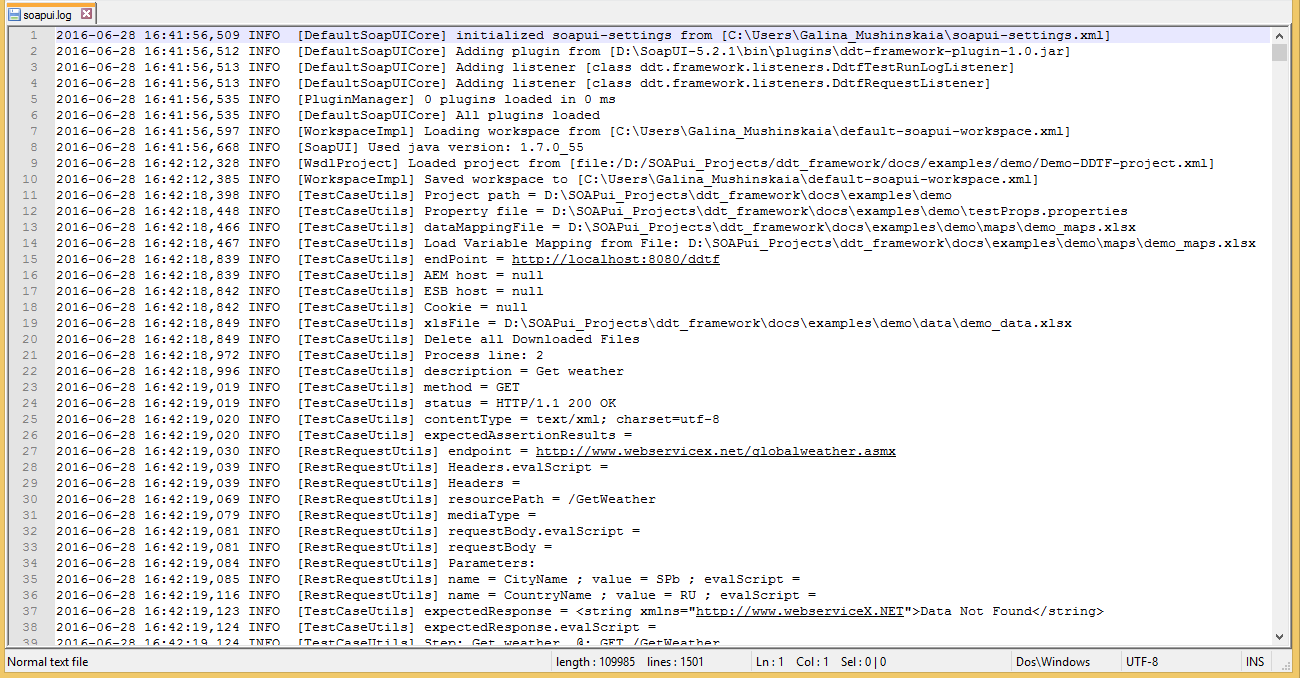


Figure 36 DDT Framework Library log in soapui.log file

* 1. The “soapui.log” File is Missed

This problem can have a place when SoapUI tool is installed on System driver.

Please make sure that SoapUI have access and rights to create and modify files in its “bin” directory.

* 1. Nothing is helped

Please feel free to contact the DDT Framework Team for further support and help.

| REVISION HISTORY | | | | | |
| --- | --- | --- | --- | --- | --- |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| n.n |  |  | dd-mmm-yyyy |  | dd-mmm-yyyy |
|  |  |  |  |  |  |
|  |  |  |  |  |  |