Automator\_V2 - Do more with less!

Automatotor\_V2 is the 2nd version of test Automator with enhanced GUI and additional features.

Automator is a utility developed to generate test artifacts for web service testing with minimum input data.

# New Features:

Additional features for SOAP service artifact generation.

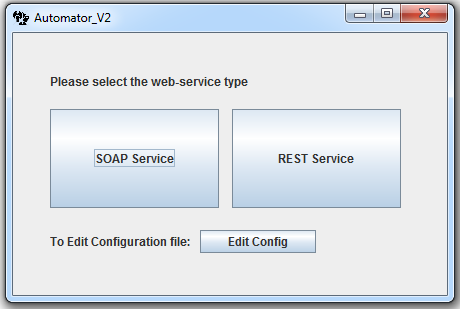
* XML request and response parsing to fetch input and output node names.

Enhanced the functionalities for REST service artifact generation (Initial stage).

Usage Details:(double click on the Automatotor\_V2 jar file to start)

Home screen of Automatotor\_V2 appears as below, contains 3 buttons –

1. ***SOAP Service***: click to generate artifacts for SOAP services
2. ***REST Service***: click to generate artifacts for REST services
3. ***Edit Config***: click to edit the Automatotor\_V2 configuration file, contains properties for different actions to be performed by the utility.



***SOAP Service window:***

SOAP service artifact generation window contains below features,

**Remove Repeated Nodes**: checkbox to specify if the parser should omit the repeated nodes from request/response XML.

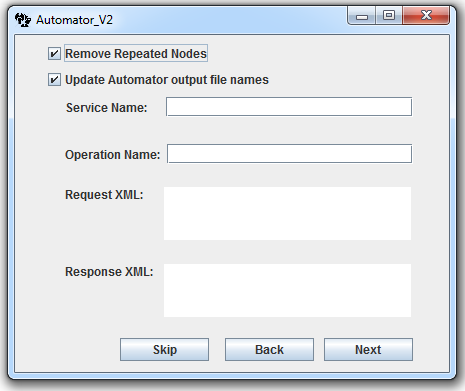
**Update Automator output file names:** to update the output file names according to Service name and operation name.

**Service Name**: input field for service name

**Operation Name:** input field for operation name

**Request XML:** input field for soap request XML

**Response XML:** input field for response XML



**Buttons:**

**Skip:** will go to next screen.

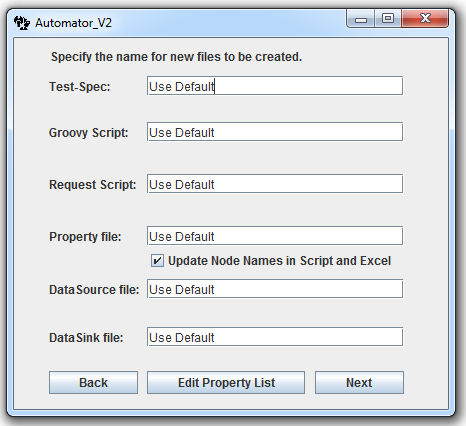
**Back:** will go to the previous screen

**Next:** will go to next screen and perform actions using input data provided in this screen (Updates output file name, Request/Response parsing)

***New File name window:***

On clicking Next button from SOAP service window, it goes to *‘New file name window’.*

User can specify the names for new files, else it will generate artifacts with default names.



**Buttons:**

**Back:** will go to previous window

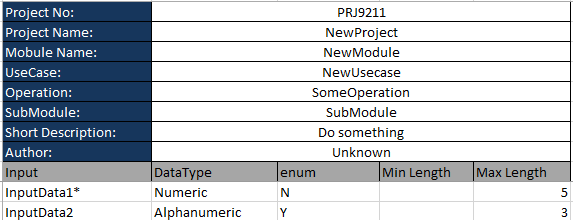
**Next:** will go to next screen

**Edit Property List:** will open the input data excel sheet, containing parsed node names. User can update/edit these details in this file. Also user has to provide additional details in this file,

1. Update the short description field: *Short description is used while creating test case description.*

**Ex:** if short description = *Create account*, then valid test case description will be *Create account with all input fields….*

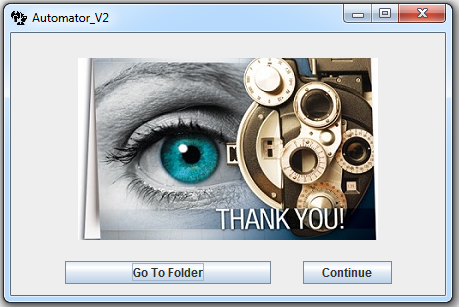
1. Also if the user wants to generate test cases, user has to fill the details in columns DataType, enum, MinLength and Maxlength.

~~~~

**End Screen:**

Once all the files are generated it goes to the *end screen,* containing two buttons

1. **Go To Folder:** will go to the folder where new files are generated.
2. **Continue:** will go to the home screen**.**



**Generated Files:**

**Test Case/Test-Spec file:**

Test Automator config file contain property to generate test case file or disable generating test cases. If it is set to ‘Y’, it will generate test case file.

**Test Data files:** contains an input data file and an output template file

**Property File**: A property file which can be used as Data-Source property file in SoapUI.

**Groovy Script:** validation groovy script.

NOTE: By default it uses CommonUtils groovy library functions (attached below) in validation script, it is required to configure the common script path while exicuting it in soapui.



# Configuring common script path in SoapUI:

Need for Script Library: Used to save common functionalities as methods and can be referred in different groovy scripts. This will reduce repeatability and increase reusability.

* Make new folder (e.g. "C:\MyProject\GroovyLib" – preferably in the project directory), Add groovy class file into the folder, “CommonUtils.groovy”.
* Set **File** > **Preferences** > **SoapUI Pro** tab > **Script Library** to library folder. (I would set that to "C: \MyProject\GroovyLib" in my example.)
* Then restart SoapUI Pro to pick up the library script.
* Call Groovy as follow (new groovy file)

// CommonUtils.groovy

c = new CommonUtils()

# Design:

