**T3 Automation**

**Automation Tool for Smoke Testing for T3**

**INFOSYS TECHNOLOGIES LIMITED,**

**Pune**

|  |  |  |  |
| --- | --- | --- | --- |
| Document No. : | 01 | Version.Rev : | 1.0 |
| Created by: | Richa Khandelwal,  Rohit | Date: | 12-27-2016 |

**REVISION LIST**

| **Ver.**  **Rev** | **Date** | **Author** | **Reviewed By** | **Description** |
| --- | --- | --- | --- | --- |
| 1.00 | 12-27-2016 | Richa Khandelwal,  Rohit |  | Initial draft |

**TABLE OF CONTENTS**

* [Overview](#_Scope)
* [Scope](#_Scope)
* Proposed [Automation](#_Proposed_Automation_Execution) Execution Workflow
* [Sequence Diagram](#_Sequence_Diagram)
* [Class Details](#_Class_Details)
* [Tools Used](#_Tools_Used_1)
* [Pre-requisite](#_Pre-Requisite)
* [Initial Setup](#_Initial_Setup)
* [Execution Steps](#_Execution_Steps_2)
* [Limitation](#_Limitation_1)
* [JENKINS IMPLEMENTATION](#_JENKINS_IMPLEMENTATION)

# Overview

T3 Automation Tool is targeted for end to end testing of T3 test scenarios.

The Automation suite comprises of test data and workflow validation across OSM and BRM for various orders for different Products .10 TCs collectively for T3 has been targeted in this phase of Automation.

The Automation suite comprises of end to end testing of T3 framework which comprises of Selenium Automation, Creating Customer Account, Creating Orders, Modifying Address/ Orders, Modifying and Cancelling Order. The Order is submitted through Selenium automation by picking up the scenarios one by one from input sheet i.e. FunctionDataMap.xls. After successful order submission the order is completed by finishing all the tasks in OSM and the validation is followed by checking the changes reflected in BRM Database to make sure that changes to data in BRM and CRM are same.

Automation suite will take care of Logs and create standard Reports which includes the error snapshot and consolidated result of all status of TC’s in a single run.

Scripts are run for following Test Case Scenarios: -

* Order Creation for E Access product.
* Order modification after the order is created for E Access product.
* Order Creation for Fiber Internet product.
* Order modification after the order is created for Fiber Internet product.
* Revising the Order created for EVPL product.
* Revising the Order created for Fiber Wavelength product.
* Disconnecting the Order created for EVPL product.
* Cancelling the order created for EPL product.
* Validating the CRM to BRM functionality for change in Address.

## T3 Automation Tool functionality:

It is targeted to drive and complete end to end Test Case execution including verification and validation at various check points without any manual intervention in a multithreaded environment. It also provides control over the flow through process by recognizing the dynamic status of the order.

# Scope

Attached document covers the scope for the test case scenarios.



# Proposed Automation Execution Workflow

# Sequence Diagram

Below is sequence diagram of the application, which shows the workflow of the Automation tool done so far:

C:\Users\Rohit10.ITLINFOSYS\Downloads\Untitled Diagram (2).png

The workflow of the automation framework is as follows:-

* The Main Engine invokes the readExecutionMap method which selects the testcases marked as yes to be executed and returns the testcase object instances (in the form of threads) to execute each test case.
* The testCaseObject instances is passes as input to the TestCaseProcessor class which reads the FunctionDataMap.xls sheet corresponding to it in a multithreaded environment. A hash map containing all the business functions marked as ‘Y’ for each testCaseObject instance is created and set as functionDataMap for that testCase instance.
* The business functions for each test case are implemented in the BusinessFunctionRepository class which are invoked sequentially based on the input provided by the functionDataMap.
* Input to each business function for UI is obtained from the objectDataMap.xls sheet. The business functions as per the functionDataMap performs the steps for each testcaseObject instance.
* After customizing the order all the tasks are completed in OSM which are validated in CRM after every step. The override Price and billing Dates are validated in BRM using Database connectivity and SQL queries after an Order is placed through CRM. The same process follows modification / cancellation of any Order.

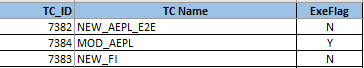
# MAPS:

The framework incorporates three types of maps:

* Execution Map
* Function Map
* Object Map

EXECUTION MAP:

Execution Map specifies which test case to be executed.

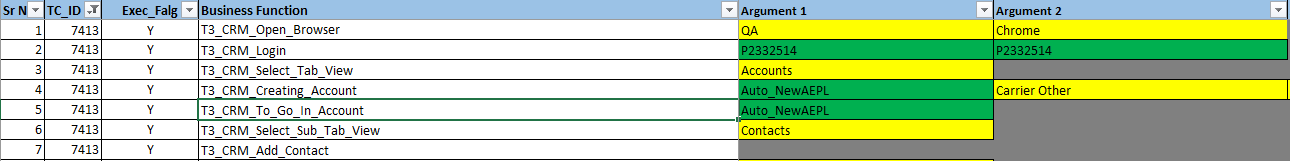


There are three tags in execution map:

* TC\_ID: This field contains the test case id as mentioned in the smoke test excel sheet.
* TC Name: This field contains the test case name for the portals.
* ExeFlag: This field contains a flag value Y or N. ’Y’ specifies that the test case needs to be executed while ‘N’ specifies that we need to skip the test case execution.

Function MAP:

Function Map specifies all the functions to be performed according to the test case steps for each test case.

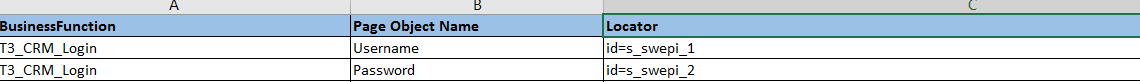


There are following tags in the function map:

* TC\_ID: This field contains the test case Id. These IDs are mapped to the IDs specified in the execution map.
* ExeFlag: This field contains a flag value ‘Y’ or ‘N’ specifying that the business function in the adjacent field for that particular test case needs to be executed or not. ‘Y’ specifies that the execution of the business function is required in that test case step.
* Business Function: For each TC\_ID there are a number of steps that needs to be executed. This field contains the business function names which are created corresponding to each step for a test case.
* Argument1, Argument2, Argument3…. Argument(n): These fields contain the input values that are passed as parameter to the business functions. It may contain a dropdown menu or user-input value. These fields are optional depending upon whether the function needs input values or not.

Object MAP:

Object Map provide the locator id and locator value of the elements of UI that are accessed by the code in the business functions for UI-flow.



There are following tags in the function map:

* Business Function: This field specifies the business function names. These names are mapped to the function names specified by the Business function field of the Function Map.
* Page Object Name: This field contains the user-defined object/element names in the UI. These names are used as keyword by the business functions to fetch the locator id and locator value from this excel.
* Locator: This field contains the UI element position description in ‘Locator ID= Locator Value’ format. For e.g. id=name

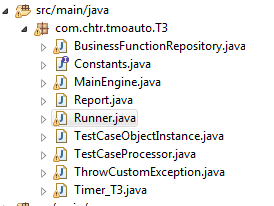
Relations between the maps:

# 

# Class Details

# Individual automation harness

These is only one main package in the framework containing all the classes:



# 

# Tools Used

* Eclipse (Which supports Maven Project)
* Maven
* Jenkins 1.6.10
* Selenium WebDriver Version 2.44.0
* GitHub Desktop Application
* SQL Developer (to connect to DB)
* HP ALM

# Pre-Requisite

* Knowledge of working with selenium in java
* Maven tool.
* Valid input(Product Name, Product characteristics ) are required for all the 10 test cases.
* No open order on any of the 10 test case scenarios. As we are creating a new Order every time before modifying/ Revising it .
* Config File need to be updated in case of any change in application login credentials.

# Initial Setup

* Install Eclipse (Preferably Eclipse Kepler or Juno) on your machine from following link
  + <https://www.eclipse.org/>
* Install JDK 1.8 and Configure your Eclipse to use jre8.To configure your Eclipse please follow following link:
  + <http://help.eclipse.org/luna/index.jsp?topic=%2Forg.eclipse.jdt.doc.user%2Ftasks%2Ftask-add_new_jre.htm>
* Copy the Automation tool to the local machine from below path:
* Separate folder is created in the Bit Bucket for each of the portals under the Test Automation Application(TAA) repository as:
* T3Auto

The code for each portal is placed in their respective folders.

User can view the code using the following url:

* + https://git.corp.chartercom.com:8443/projects/TAA

For the framework utility, the code is placed in a separate folder named as the Test Automation Framework(TAF) repository.

User can view the code using the following url:

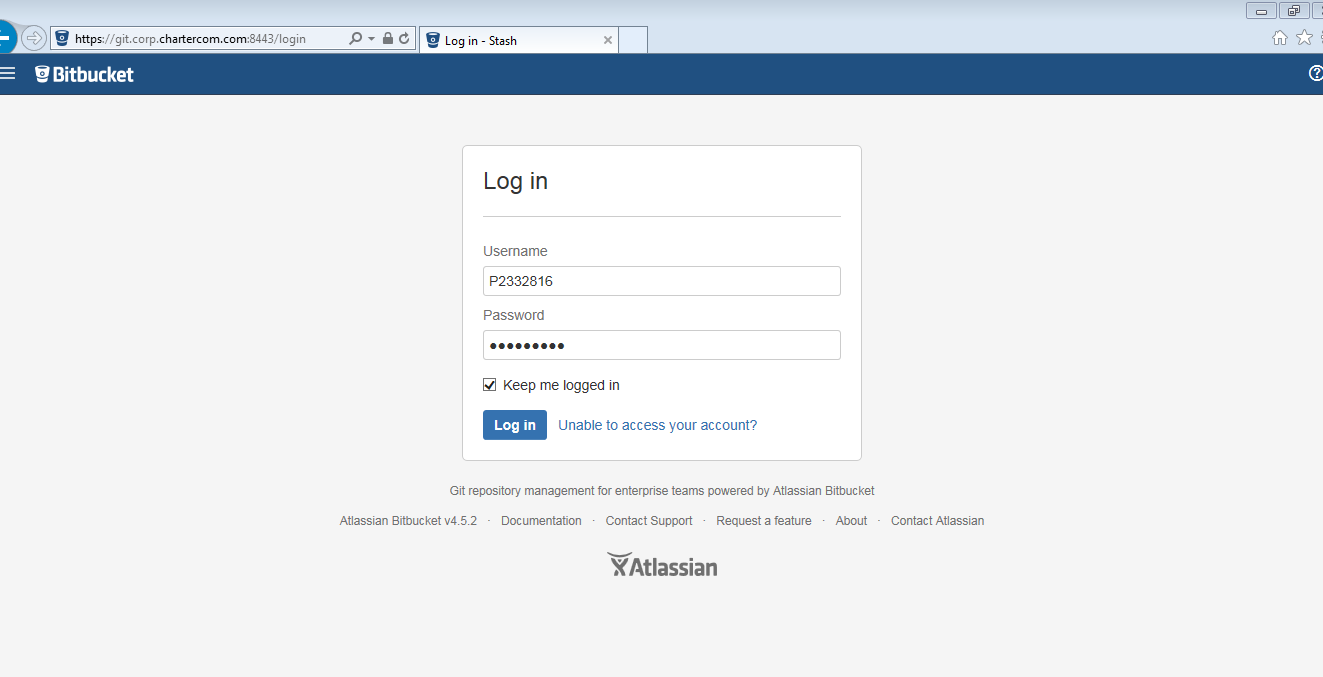
* + <https://git.corp.chartercom.com:8443/projects/TAF>

# Execution Steps

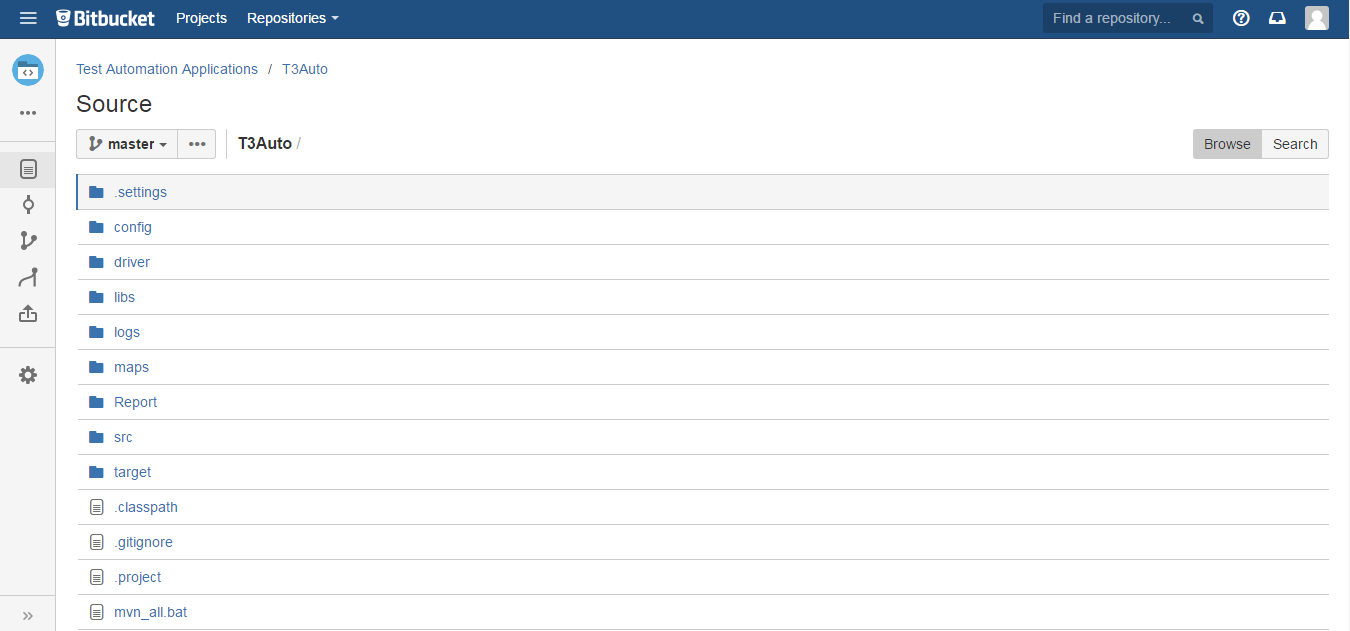
**To run the code with Eclipse:**

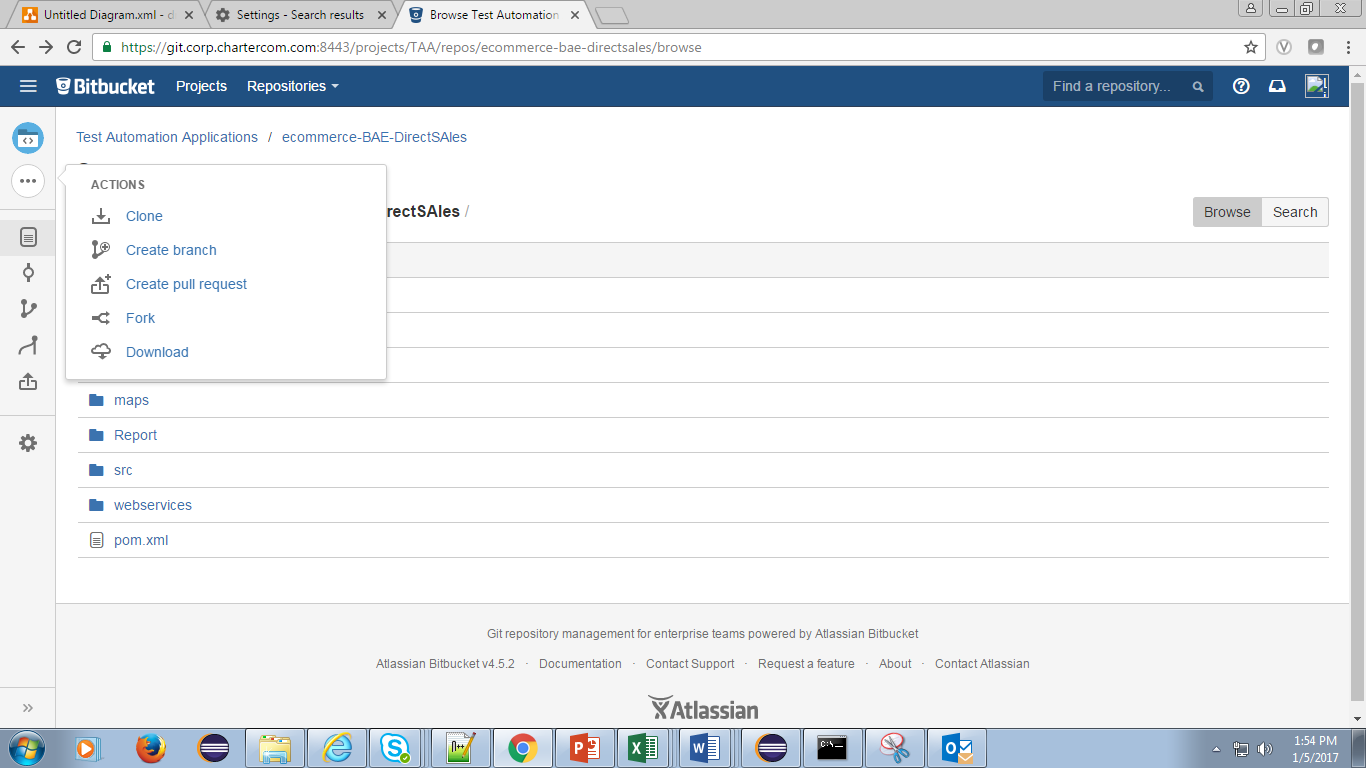
**Method 1**

* Go to Bit Bucket URL using valid credentials.



After login go to following URL in Enterprise BitBucket: - <https://git.corp.chartercom.com:8443/projects/TAA/repos/t3auto/browse>





* Click on Download. Extract after download.
* In Eclipse select import project and select the extracted code T3 Automation folder.
* Check the pom.xml file to configure the jar and to remove any error for maven dependency.
* Execute mvn\_install.bat file from cmd. This will add the jar from lib folder to the project.
* Execute mvn clean install command from the cmd using current file location.
* Go to MainEngine in under package com.chtr.tmoauto.T3.
* Right click on the file and run as java application.
* View the results in Report folder in current Date folder which contains report files in form of HTML for each test case which gives detail about the pass/failure at end of execution.

**Method 2**

Refer below document to configure Eclipse ide directly to Bitbucket

****

URLS to access other portals in the repository:

* T3 –

[https://git.corp.chartercom.com:8443/projects/TAA/repos/t3auto/browse](https://git.corp.chartercom.com:8443/projects/TAA/repos/ecommerce-charter.com/browse)

* Framework –

https://git.corp.chartercom.com:8443/projects/TAF

# Limitation

* Any updates to the T3 workflow will involve code change effort and corresponding function map, object Map modification.
* Automation code is dependent on application environment. If application is down automation may fail.

# JENKINS IMPLEMENTATION

**Jenkins Implementation**

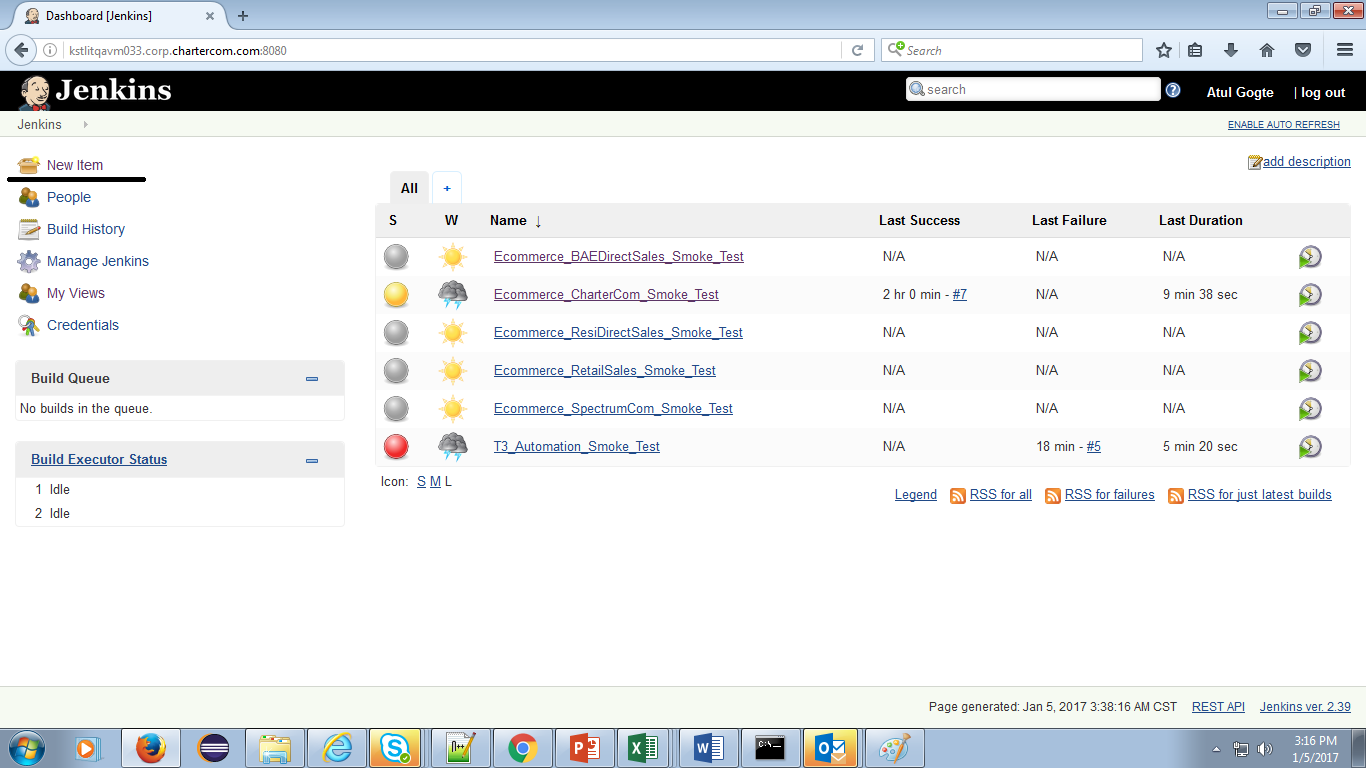
* For installation of Jenkins on your system refer to attached document.



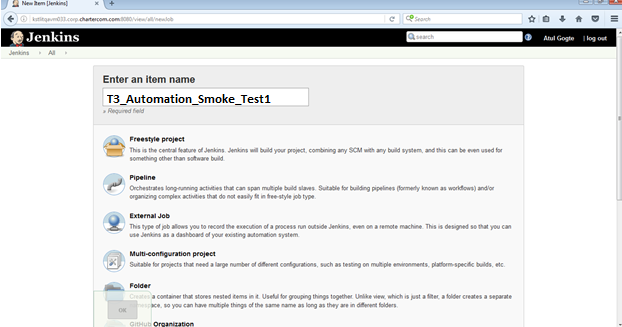
Jenkins Path : <http://kstlitqavm033.corp.chartercom.com:8080/>

To configure the Jenkins to run T3 Automation follow the below steps: -

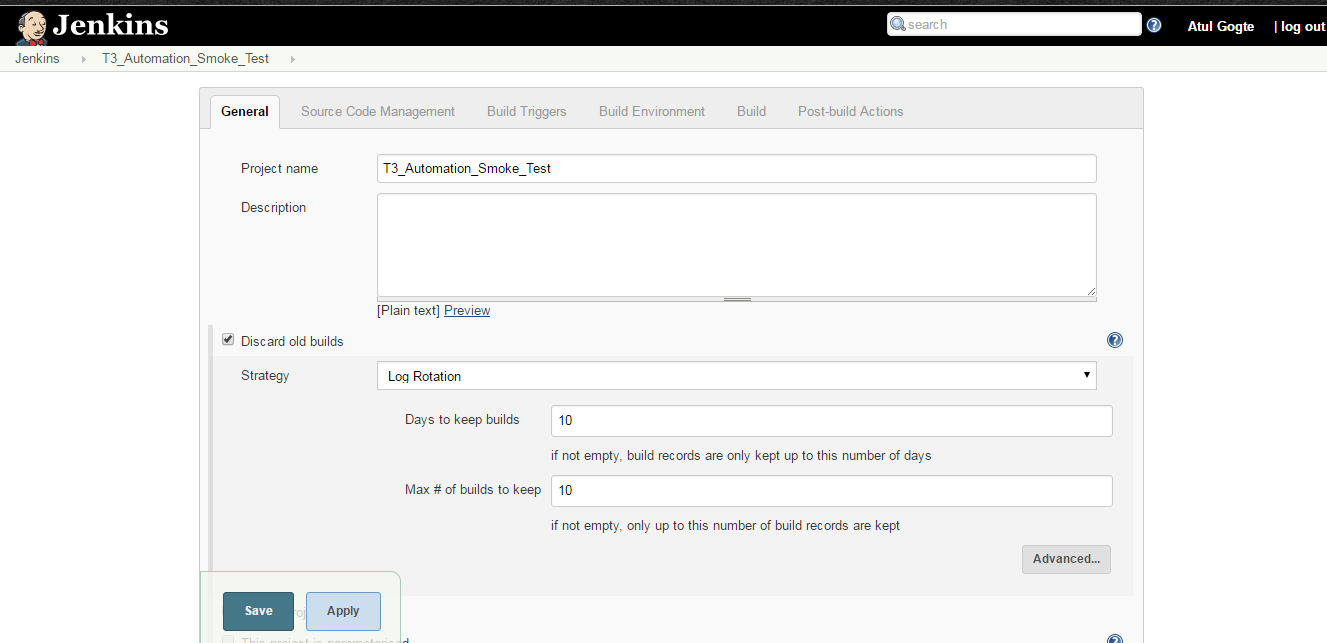
1. Open Jenkins on Mozilla Firefox.
2. Click on new Item as shown in attached snapshot.



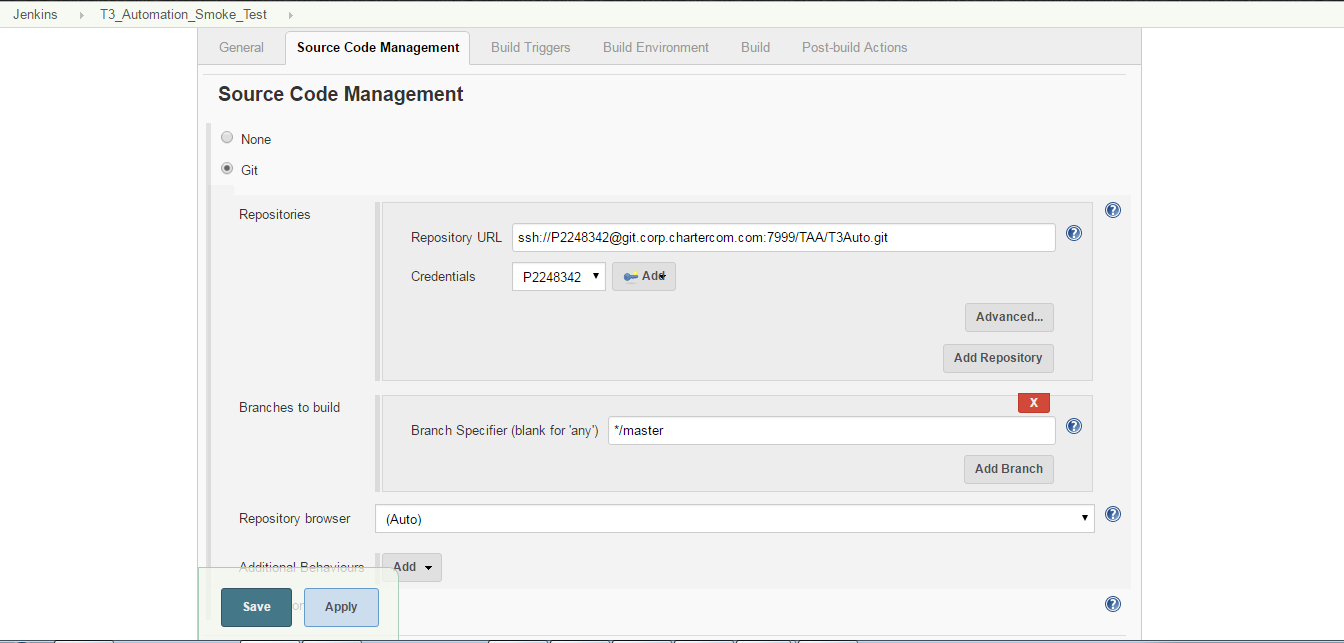
1. Give the name and select freestyle project. Refer the snapshot



1. Click on Project T3\_Automation\_Smoke\_Test.
2. Configure the project.



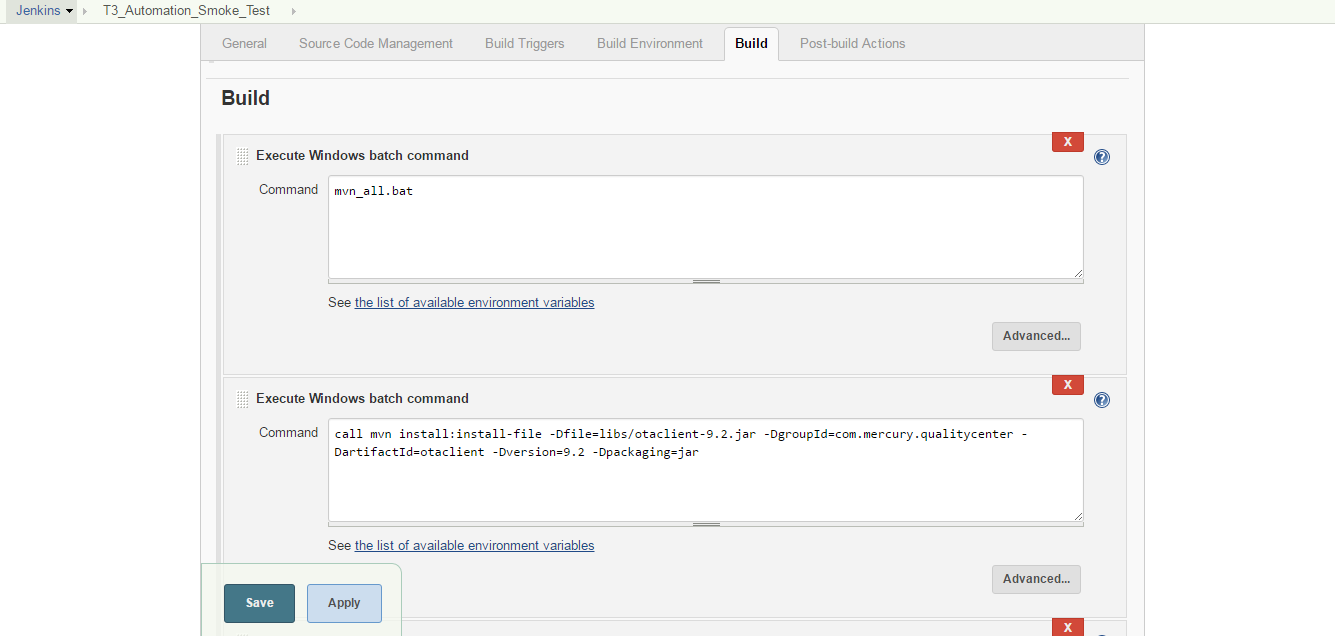
1. Go to Source Code Management. Click on GIT radio button. Provide the repository URL and credentials.



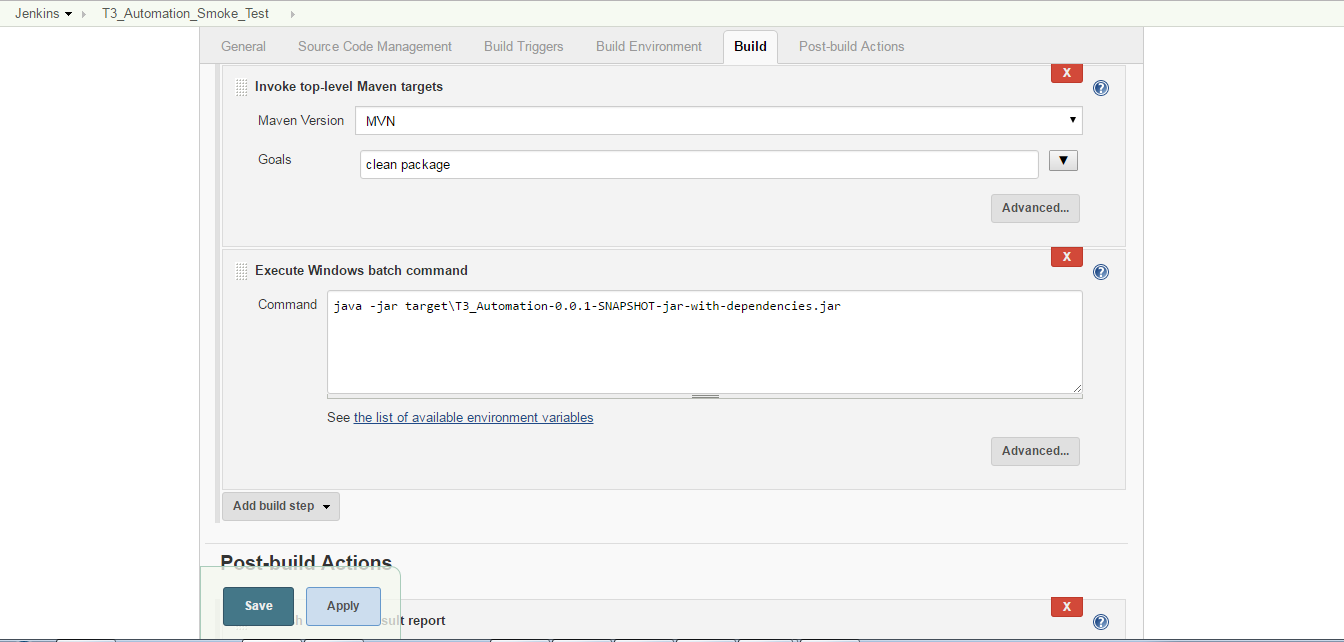
1. In Build:

For Execute Window batch command and provide the below commands:

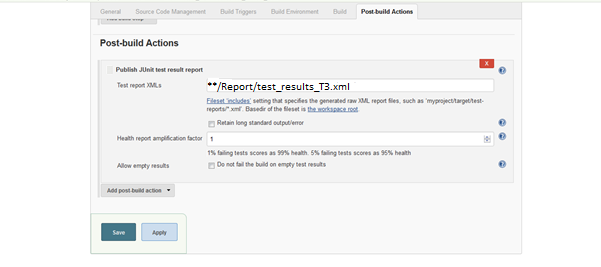
* Call mvn install:install-file -Dfile=lib/otaclient-9.2.jar -DgroupId=com.mercury.qualitycenter -DartifactId=otaclient -Dversion=9.2 -Dpackaging=jar
* call mvn install:install-file -Dfile=lib/com4j-1.0.jar -DgroupId=com.mercury.qualitycenter -DartifactId=com4j -Dversion=1.0 -Dpackaging=jar



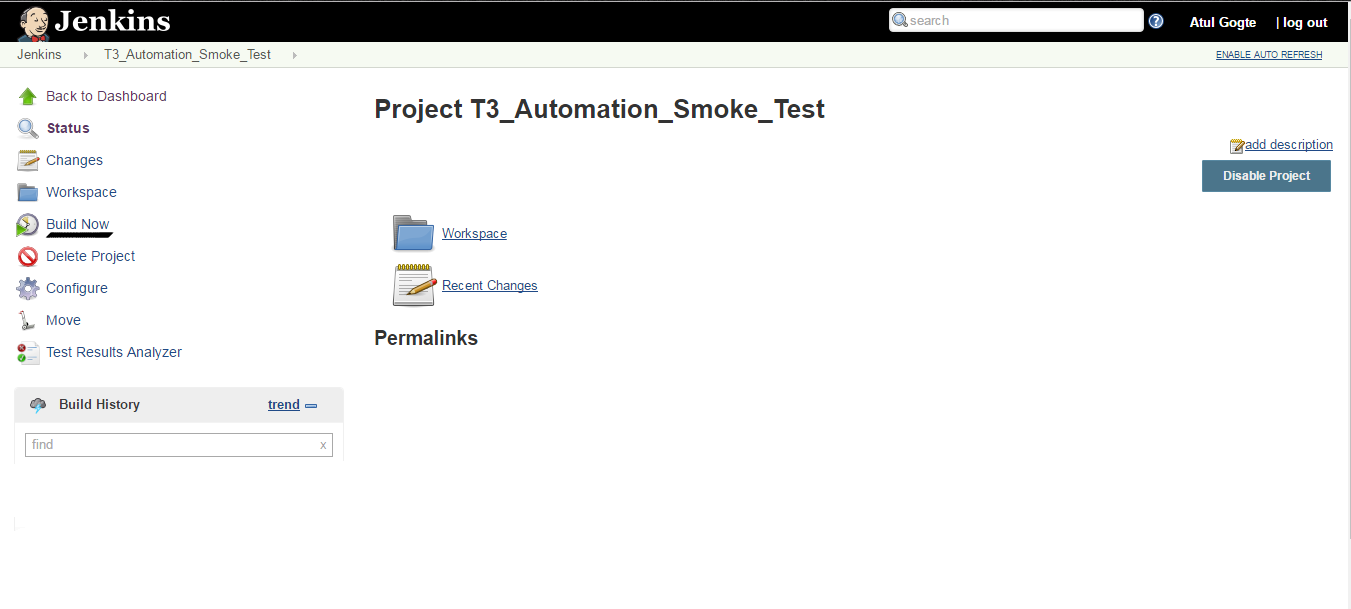
* java -jar target\T3\_Automation-0.0.1-SNAPSHOT-jar-with-dependencies.jar



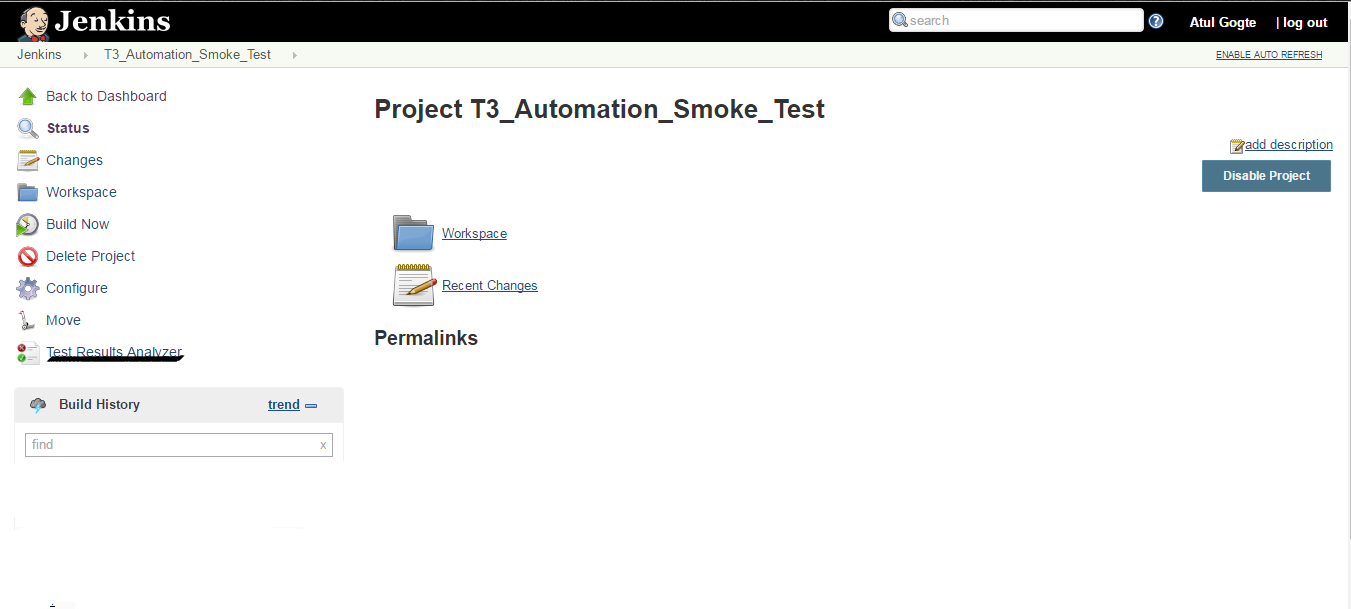
1. In Post Build Section click on Publish JUnit test result report and provide the xml name for viewing the results in Jenkins using TestResultanalyzer.



1. Click on Build now present on the left hand side.



1. Once the execution of Automation code is completed. To view the results on Jenkins click on Test Result Analyzer.



1. Click on Get Build Report.

