AUTOMATION PRACTICES FRAMEWORK DOCUMENT

For Selenium Webdriver

Abstract

This document provides general step-by-step instructions and best practices for projects conducting automation testing or must upgrade their framework per these written guidelines.

DevSecOps Automated Testing Initiative Team

1. Table of Contents

2. Purpose:

2.1 Overview:

This document is intended to provide a set of instructions and best practices for projects intending to implement automation or already have an automation framework and are required to upgrade per the instructions provided herein.

Furthermore, this document is relevant for projects using Selenium/Webdriver for any set of tests (UI, API, etc.) Although this document is intended to be applicable for the use of any project (regardless of its use of any supplementary tools or programming language), the template used in this document provides the example of a Java project, being AOIC, using Maven as its build tool. As a side note, AOIC uses Junit for test case execution, Extent reports for reporting and Cucumber and BDD for test case development. Please substitute your project name wherever AOIC is mentioned in the following images/steps.

As mentioned, regardless of a project's supplementary tools or programming language(s), the below screenshots and steps should be relevant if using Selenium/Webdriver for automated testing.

Per IT Policy, before proceeding further, please verify whether your project is using the latest version of Selenium Webdriver or at the minimum, no more than n-1 in application versions (if available and approved). To learn the latest version, kindly conduct a web search. At the time of writing this document, the latest version of Selenium Webdriver is 4.9.1 (released on May 8, 2023).

3. Tools Requirements for automation testing:

For your project's automation testing initiative, the following tools need to be installed for automation.

- Source Code Management Tool
- Integrated Development Environment
- Java Development Kit (JDK) latest version
- Maven

Note 1: Will need to get permission to access source code control system, which is project dependent (such as Github, Git, or Rational Team Concert)

Note 2: User should check to see if tool is already installed on their system. Skip installation activity if tool is already installed.

3.1 Source Code Management Tool (i.e., GIT, Rational Engineering Work Manager, etc.)

Source code management (SCM) is used to track modifications to a source code repository. SCM tracks a running history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors. SCM is also synonymous with Version control.

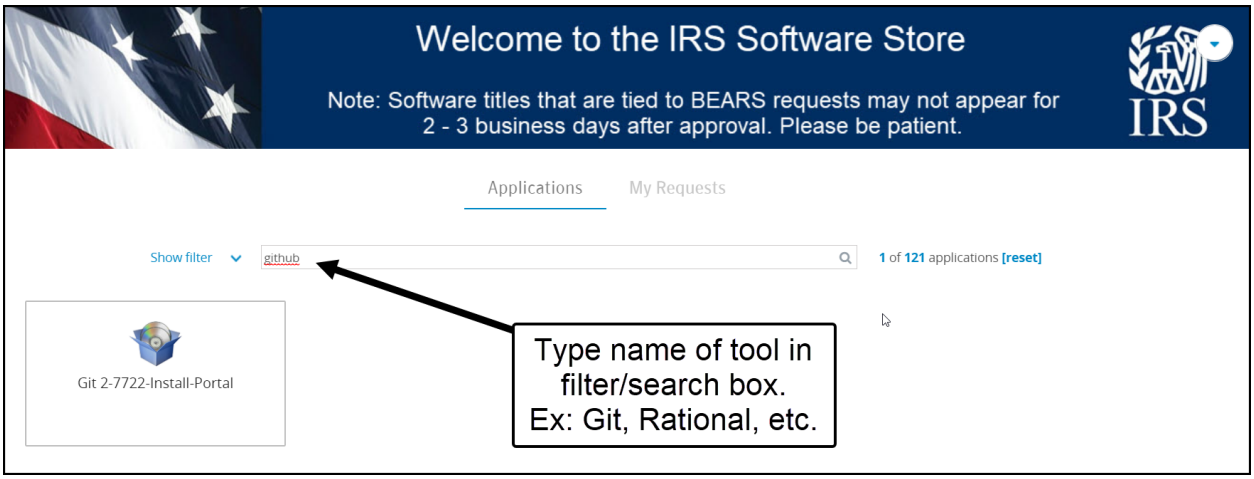
3.1.1 Authorization is needed to download and access SCM tools.

- Submit Business Entitlement Access Request System (BEARS) request for SCM tool.

Note: there are different types of entitlement groups for each SCM tool in BEARS (i.e., Admin, Dev User, Prod User, etc.). The specific entitlement group to select will be based on user need.

3.1.2 Request and install tool from Symantec Software Portal.

Note: Only available in portal if IRS has approved the SCM tool for user self-installation.



.

3.2 Integrated Development Environment (i.e., Eclipse, Selenium IDE, JBoss, etc.) Luna

Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment.

3.2.1 Authorization is needed to download and access IDE tools.

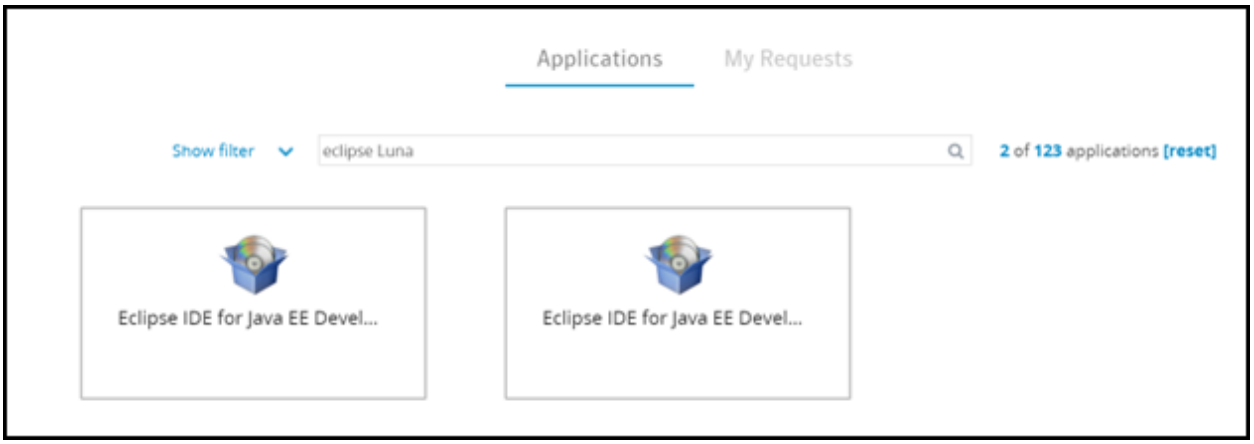
- Submit Business Entitlement Access Request System (BEARS) request for the IDE tool.

Note: there are different types of entitlement groups for each IDE tool in BEARS (i.e., Admin, Dev User, Prod User, etc.). The specific entitlement group to select will be based on user need.

3.2.2 Request and install tool from Symantec Software Portal

Note: only available in portal if IRS has approved the SCM tool for user self-installation.

Below example is for self-installation of Eclipse



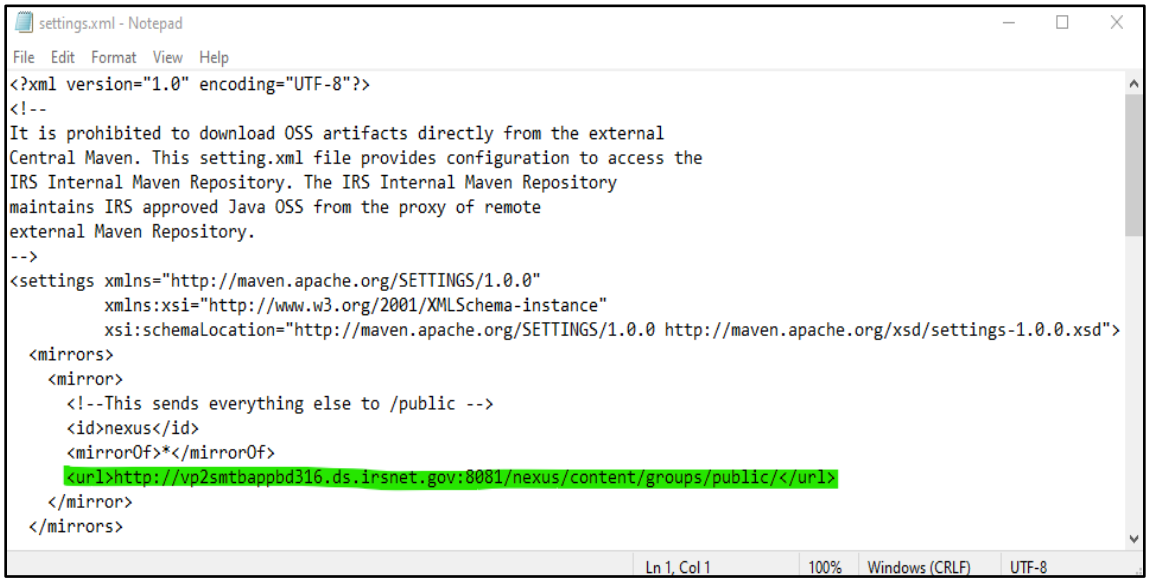
	Eclipse IDE for Java EE Developers 4.4 (Luna)_9631-Install - Portal	nct001ma4432250	4/17/2023 3:37:37 PM	4/17/2023 3:37:38 PM	Succeeded	Approved (Complete)

3.2.3 Get access to the User token.

- Submit Business Entitlement Access Request System (BEARS) request
- DEV USER NEXUS SN (INTERNAL REPOSITORY NEXUS)

3.2.4 Prepare Settings.xml for each test user

- Download the setting.xml
- Open the settings.xml in notepad



- Navigate to server's tag section as below and Update username and password in settings .XML as shown in screenshot below.

3.3 JDK latest version

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It offers a collection of tools and libraries necessary for developing Java applications.

3.3.1 Installing JDK

- Note: JDK must be installed **before Maven installation**
- Navigate to OSGetServices by clicking on https://irworks.for.irs.gov/esc

3.3.2 Verify in your workstation and the configuration after installation.

- IRS software versioning goal is current release – 1

3.4 Maven

Maven is a build automation tool used primarily for Java projects. It is designed to manage dependencies and the software lifecycle, and to work with plugins that allow users to add other tasks to the standard compile, test, package, install and deploy tasks.

Note: Most of newer versions of IDEs already come with Maven plug-in, so there may not be a need to install a standalone Maven client.

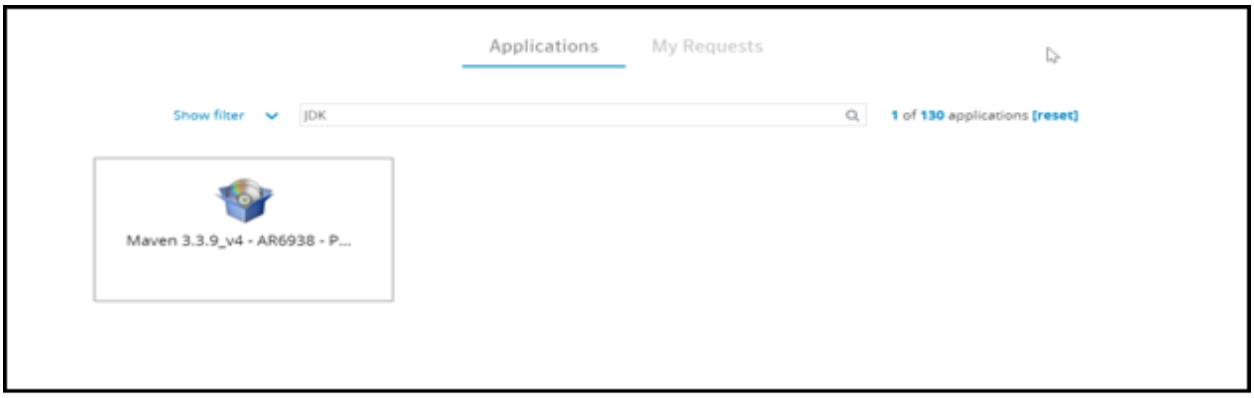
3.4.1 Get authorization to download and access Maven tool.

- Submit Business Entitlement Access Request System (BEARS) request

Note: there are different types of entitlement groups in BEARS (i.e., Admin, Dev User, Prod User, etc.). The specific entitlement group to select will be based on user need.

3.4.2 Request and install tool from Symantec Software Portal.

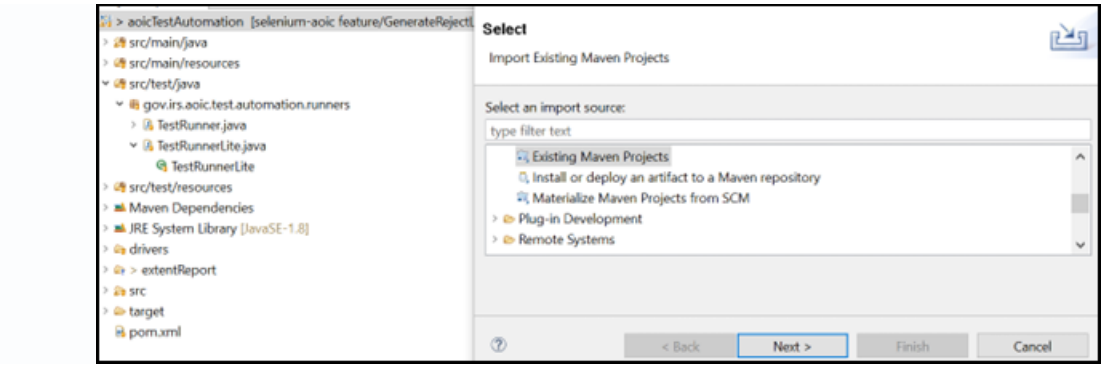
Note: Maven tool is a self-installation application



3.4.3 Import Maven Project:

The IDE should have a Maven plug-in

- Inside Eclipse, go to "File" > "Import
- In the "Import" dialog, choose "Existing Maven Projects" and click "Next
- Locate the Project Root Directory Click on the "Browse" button and navigate to the root directory of your Maven Selenium project.
- Select the Project:
- You should see your project listed in the "Projects" section. Ensure it is checked and click "Finish."



4. Build Selenium Automation:

- This section helps developers to set up Test Automation Project in Eclipse for the project that they are working on.

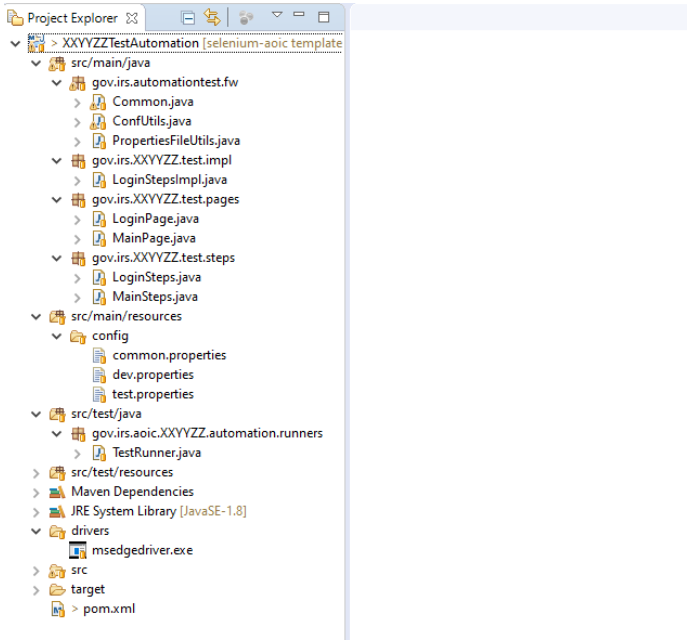
Note 1: Automation code should not be browser specific. Should be developed in a manner that works with multiple browsers.

Note 2: Automation code should be developed to work in Windows and Linux environments.

4.1 Project Structure:

selenium-automation.zip

- Download and extract the above zip file.
- Import as maven project in eclipse.
- Please check below screenshot after importing as Maven project



src/main/java: This consists of packages that have all automation code

gov.irs.automationtest.fw – This package consists of framework methods to be used to implement selenium automation testing any project.

Note: If there is a need to add new methods or update in existing methods, please schedule meeting to finalize the requirement. Please do not do any code changes in this package unless everyone agrees to do the changes in the meeting. In future, once it stabilizes, we can move this out of this template project and create new framework project and upload in nexus so that others can include as dependency in pom.xml

gov.irs.XXYYZZ.test.impl – This package consists of implementation classes for each test case.

gov.irs.XXYYZZ.test.pages – This package consists of classes related to each application page. All the objects and functions that can be performed on a page are defined here.

gov.irs.XXYYZZ.test.steps – This package consists of code that connects the BDD test cases written in Gherkin using Cucumber. Each test case executed first runs the functions defined here

src/main/resources: This consists of all config property files with environment specific parameters etc.

src/test/java: This has all the testrunners that are executed for executing different tests

src/test/resources: This has all the feature files that have test cases defined in gherkin format

- Replace all occurrences of XXYYZZ with your project name.
- Update *.java, *.feature, *.properties files based on the test case requirement
- Implement automation of test cases referring the below API docs

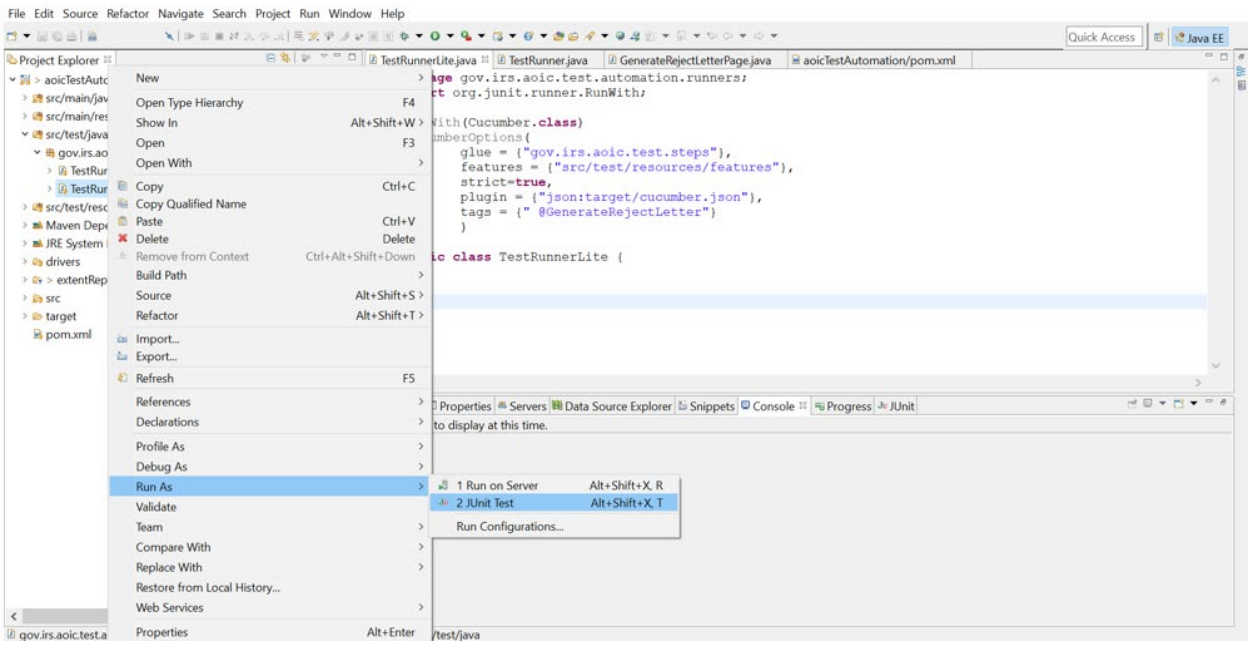
AutomationFrameworkDocs.pdf

4.2 Test Case Execution:

Test cases can be executed by running the TestRunners. Test Runner can be executed locally on any machine or by using CICD pipeline/Jenkins job.

5. Run the Test Automation Programs:

- Right-click on your test class in Eclipse and select "Run As" > "Junit Test"
- The test automation program will execute, and the browser will open, performing the specified actions.
- View Test Results
- Observe the output in the Eclipse console to check if the test passed or failed
- Repeat for Other Test Cases


File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Project Explorer
TestRunnerLite.java
TestRunner.java
GenerateRejectLetterPage.java
aoicTestAutomation/pom.xml
New
Open Type Hierarchy F4
Show In Alt+Shift+W
Open F3
Open With
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
References
Declarations
Profile As
Debug As
Run As
Validate
Team
Compare With
Replace With
Restore from Local History...
Web Services
Properties Alt+Enter
1 Run on Server Alt+Shift+X, R
2 JUnit Test Alt+Shift+X, T
Run Configurations...
Properties
Servers
Data Source Explorer
Snippets
Console
Progress
JUnit
to display at this time.
/test/java