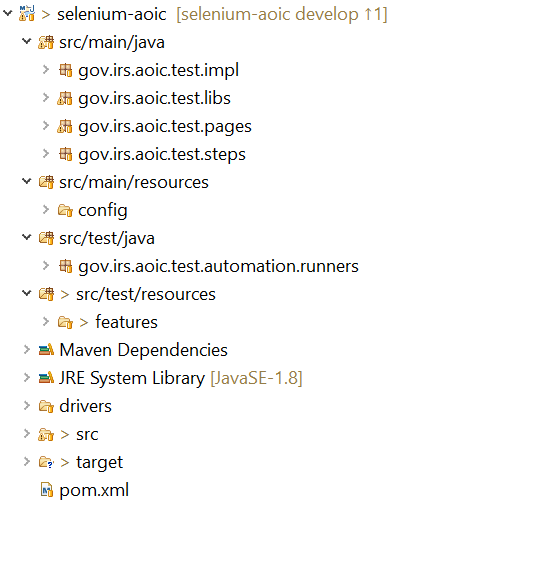
# Functional Test Automation using Selenium/WebDriver

## Overview:

AOIC project uses Selenium/Webdriver for functional test automation of UI cases. This template can be used for automating any tests (UI, API etc.). Current template is a Java project using Maven as the build tool. The project uses Junit for test case execution, Extent reports for reporting, Cucumber and BDD for test case development.

## Project Structure:



**Src/main/java**: This consists of packages that has all the automation code

*Gov.irs.aoic.test.impl* – This package consists of code related to any business logic. Some applications do not need this package and don’t have any classes defined here

*Gov.irs.aoic.test.libs* – This package consists of code related to shared libraries. This will not have any application specific code.

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

*Gov.irs.aoic.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 files, environment specific parameters etc.

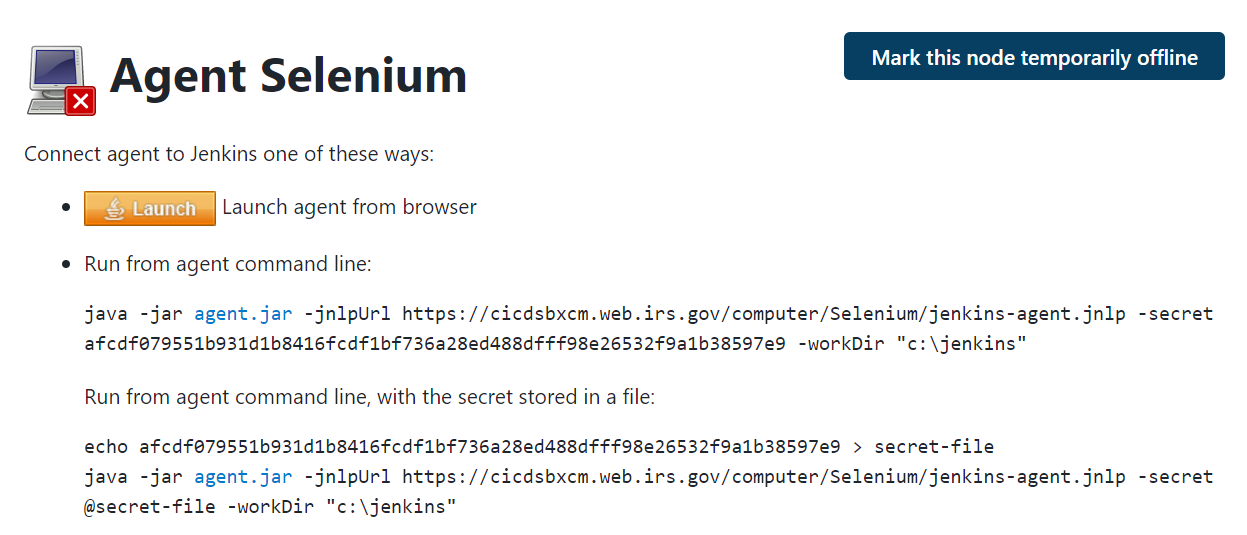
**Src/test/java:** This has all the testrunners that are executed for executing different tests

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

## 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.

**CICD Setup**:

* Jenkins job can be setup using pipeline script and TestRunner can be executed using Maven Goals.
* Windows Agent should be setup as a slave and tests should be restricted to be executed on the windows agent. The agent should have required software to run tests
  + 
* Another way to set up windows machine is to use Selenium Grid and set it up as host and node. Make changes in the automation code to use RemoteDriver to connect using Grid.
* Headless browsers can also be used to execute UI automation. Headless browsers need to be installed on Jenkins workers
  + [Getting Started with Headless Chrome - Chrome Developers](https://developer.chrome.com/blog/headless-chrome/)

**Existing CICD Setup:**

1. Standalone pipeline job to execute tests: This is just a pipeline script and not part of the CD Pipeline

[SELENIUM-TEST-PIPELINE-LITE [zz-TestArea] [Jenkins] (irs.gov)](https://cicdsbxcm.web.irs.gov/job/zz-TestArea/job/SELENIUM-TEST-PIPELINE-LITE/)

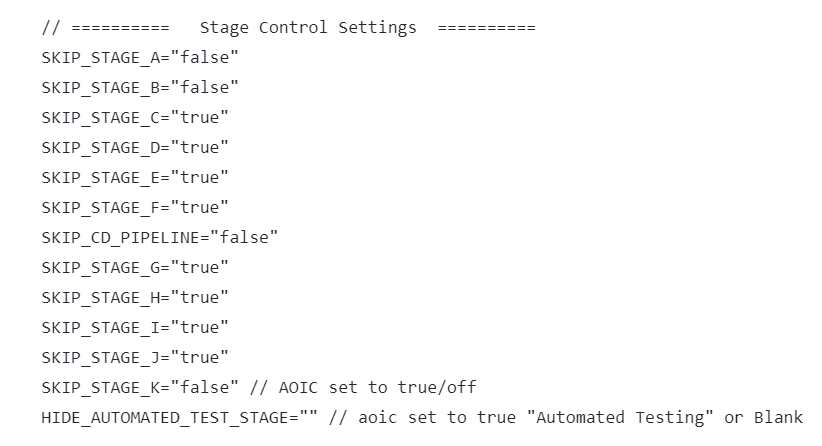
1. CICD as part of the pipeline: This Jenkins folder consists of CI and CD jobs that are part of the AOIC CICD pipeline. CD pipeline just consists of Automated Stage

[All [zz-TestArea » SelAutoTest] [Jenkins] (irs.gov)](https://cicdsbxcm.web.irs.gov/job/zz-TestArea/job/SelAutoTest/)

**CICD Code updates for executing automation as part of CD Pipeline:**

1. Created new branch (cicdmtadevelop) in CICD-ONBOARDING library. Created new jenkinsfile AOIC\_SelAuto\_Jenkinsfile which has the stages needed for running automation.

[cicd-onboarding/AOIC\_SelAuto\_Jenkinsfile at cicdmtadevelop · EOPS/cicd-onboarding (irs.gov)](https://github.enterprise.irs.gov/EOPS/cicd-onboarding/blob/cicdmtadevelop/project-Jenkinsfiles/CICD/AOIC_SelAuto_Jenkinsfile)



1. Created new branch (aoicautodevelop) and updated CICD\_JAVA\_TEMPLATE to execute automated tests in StageK. Also created 2 functions to execute automation and send email ([RunAutomatedTest.groovy](https://github.enterprise.irs.gov/EOPS/cicd-shared-library/blob/aoicautodevelop/vars/RunAutomatedTest.groovy), [SendAutoTestEmail.groovy](https://github.enterprise.irs.gov/EOPS/cicd-shared-library/blob/aoicautodevelop/vars/SendAutoTestEmail.groovy))

[cicd-shared-library/vars at aoicautodevelop · EOPS/cicd-shared-library (irs.gov)](https://github.enterprise.irs.gov/EOPS/cicd-shared-library/tree/aoicautodevelop/vars)



**Creating Core Shared Library for all projects:**

Core Automation shared library is like a template that can be built and pushed to nexus. It has all common libraries that will be needed for automation. All automation projects can add this as a dependency and get its features. Below is a sample project

<dependency>

<groupId>gov.irs.automation</groupId>

<artifactId>template</artifactId>

<version>0.0.1-SNAPSHOT</version>

</dependency>