

Live Project

Follow the below steps to start working on the real time project:

1. Download the Existing Framework from here - [Click here](#)
 2. Extract the zip file and copy the project folder along with its items in any folder location
 3. Create a GitHub Repository say DemoQAFoxLiveProject (Include ReadMe file)
 4. Clone the GitHub Repo into your local machine
 5. Copy the files available in the Framework which we have downloaded in step 1
 6. Import the Project into Eclipse IDE and resolve the errors if any
 7. Open GitBash inside the cloned Project folder
 8. Upload the updated code/files to GitHub repository
 - git status
 - git add .
 - git commit -m "Uploading Initial automation framework"
 - git push origin master
 9. Delete any existing feature files and Create Register.feature file and first scenario under src/test/resources > Features folder - [View Scenario here](#)
 10. Create Register.java file under src/test/java > stepdef package and implement all the steps in first scenario
 - First run the feature file without implementation to get the high level implementation in output
 - Remove the errors and unnecessary comments/code which is auto-generated
 - Write the code for opening the Application URL in the browser and Understand
 - Copy the Hooks under the step-def package - [Download Hooks class here](#)
 - Understand the methods in Hooks class
 - Remove the Hooks related methods from Base Class
 - Close and open the Register.feature file and observe that I am getting an error
 - To overcome this error, I will remove Cucumber-eclipse add-on from Eclipse IDE and in place of it I will install Cucumber JVM eclipse plugin from <https://marketplace.eclipse.org/content/cucumber-jvm-eclipse-plugin>
 - Update the Runner class to execute the so far implemented scenario - [View Runner Class here](#)
 - `@RunWith(Cucumber.class)`
 - `@CucumberOptions(features={"classpath:FeatureFiles/Register.feature"},`
`glue={"classpath:com.tutorialsninja.automation.stepdef"},`
`plugin={"html:target/cucumber_html_report"},`
`tags={"@Register", "@One"})`
 - Execute the Runner class using JUnit and check the result
 - If everything goes well, upload the updated code to GitHub
-