CA - Experiment 6 - Continuous Integration with Jenkins: Setting Up a CI Pipeline, Integrating Jenkins with Maven/Gradle, Running Automated Builds and Tests

---

Experiment 6: Continuous Integration with Jenkins

Objective

To set up a Continuous Integration (CI) pipeline in Jenkins, integrate it with Git, and run Selenium Java tests using Maven.

Prerequisites

Before proceeding, ensure the following:

✅ **Jenkins is installed and running** on your system. If not, refer to [Experiment 5].
✅ **Git is installed** and configured in Jenkins. (Verify with `git --version`).
✅ **Maven is installed** and configured in Jenkins. (Check with `mvn -version`).
✅ **Selenium Maven Project is ready** with test cases (`src/test/java`).
✅ **Project is stored in two places**:

- Locally on your system (e.g., `D:\Idea Projects\MVNGRDLDEMO`).
- Pushed to **GitHub** with a valid repository link.
  ✅ **Jenkins has access to the GitHub repository** (via credentials).

---

1. Configuring Jenkins & Git Integration

Step 1: Verify Git Installation in Jenkins

1. Open **Jenkins Dashboard** → **Manage Jenkins** → **Global Tool Configuration**.
2. Under **Git**, verify the installation path (e.g., `C:\Program Files\Git\bin\git.exe`).
3. Click **Save**.

Step 2: Add GitHub Credentials in Jenkins

1. Navigate to **Manage Jenkins** → **Manage Credentials**.
2. Select **Global credentials (unrestricted)** → Click **Add Credentials**.
3. Choose **Username with password** or **SSH Key**, provide details, and click **OK**.

---

2. Running a Selenium Java Test from a Local Maven Project

Step 1: Create a New Jenkins Job

1. Go to **Jenkins Dashboard** → Click **New Item**.

2. Enter a project name → Select **Freestyle Project**.
3. Click **OK**.

Step 2: Configure the Build Step

1. Scroll to **Build** → Click **Add build step** → **Execute Windows Batch Command**.
2. Enter the following commands (**ensure correct navigation to project directory**):

```
cd D:\Idea Projects\MVNGRDLDEMO
mvn test

```

3. Click **Save** → Click **Build Now** to execute the test.

---

3. Running Selenium Tests from a GitHub Repository via Jenkins

Step 1: Set Up a New Jenkins Job for GitHub Project

1. Go to **Jenkins Dashboard** → Click **New Item**.
2. Enter a project name → Select **Freestyle Project**.
3. Click **OK**.

Step 2: Configure Git Repository in Jenkins

1. Under **Source Code Management**, select **Git**.
2. Enter your GitHub repository URL (e.g., `https://github.com/your-repo-name.git`).
3. Select the **Git credentials** configured earlier.

Step 3: Add Build Step for Maven

1. Scroll to **Build** → Click **Add build step** → **Execute Windows Batch Command**.
2. Enter the Maven test command:

```
mvn test

```

3. Click **Save**.

Step 4: Trigger the Build

1. Click **Build Now** to fetch the code from GitHub and execute the Selenium tests.
2. Check the **Console Output** to verify test execution.

Important Notes

📌 **Prerequisites are crucial!** Make sure Jenkins, Git, Maven, and Selenium projects are set up correctly before proceeding.
📌 **Always navigate to the project directory** before running `mvn test` from a local system.
📌 **Use webhooks** in GitHub to automatically trigger builds when new code is pushed.
📌 **Configure email notifications** in Jenkins for build status updates.

🔗 **Jenkins & Git Integration Video:** [To Be Added - Version 1.2]
🔗 **Running Selenium Tests in Jenkins:** [To Be Added - Version 1.2]
