Experiment No 7

Date of Performance:
Date of Submission:

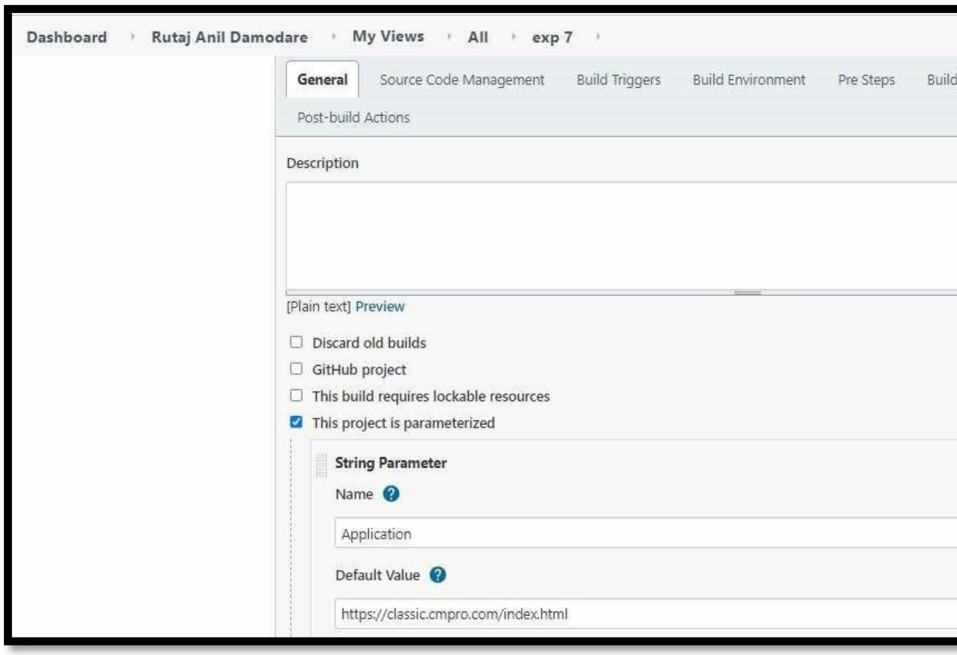
Aim: How to Run Selenium Tests in Jenkins Using Maven

Software required: Jenkins, Selenium

Theory: Jenkins and Selenium are two popular tools that are often used together in the context of automated testing for web applications. By integrating Selenium with Jenkins, teams can achieve the following benefits:

1. **Automated Testing:** Selenium can be used to write test scripts that automate web application testing. Jenkins can schedule these tests to run automatically, triggered by code changes or on a predefined schedule. This ensures that tests are executed consistently, which is especially crucial in agile development environments with frequent code updates.
2. **Cross-Browser Testing**: Selenium allows you to run tests on different web browsers, ensuring your web application works correctly on various platforms and browser versions. Jenkins can orchestrate these tests across multiple browser configurations, increasing the coverage of your testing process.
3. **Parallel Testing:** Selenium tests can be time-consuming, especially when you have a large test suite. Jenkins can distribute test execution across multiple agents or nodes, enabling parallel testing and reducing the overall testing time.
4. **Reporting and Notifications**: Jenkins provides detailed reports of test results, allowing teams to quickly identify and address issues. It can also send notifications (e.g., email alerts or chat messages) to relevant stakeholders when tests fail, enabling rapid feedback and quick resolution of problems.
5. **Integration with CI/CD Pipelines:** By integrating Selenium tests into the Jenkins CI/CD pipeline, you can ensure that your application's test suite is run automatically whenever changes are made to the codebase. This helps catch bugs early in the development process and maintains a high level of software quality

<u>To setup and run selenium tests in Jenkins using maven:</u>


Dashboard › Rutaj Anil Damodare › My Views › All › exp 7 ›

General | Source Code Management | Build Triggers | Build Environment | Pre Steps | Build
Post-build Actions

Description

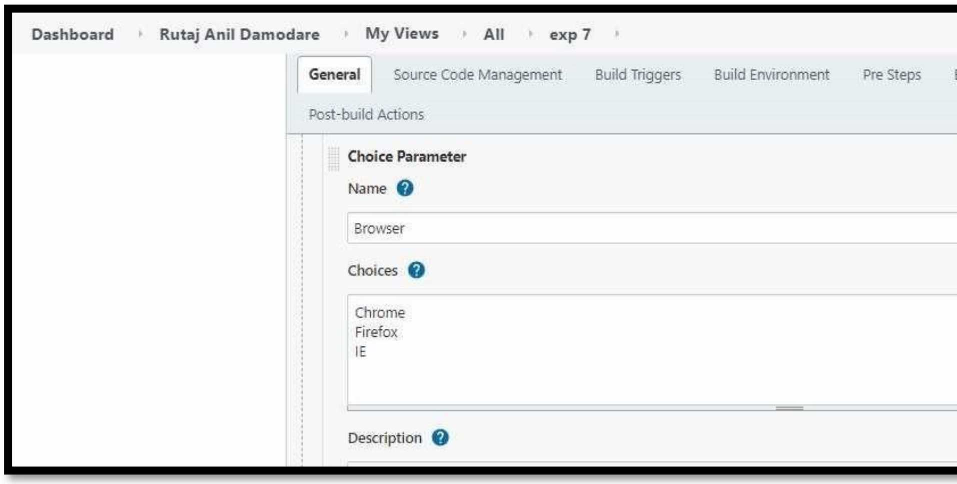
[Plain text] Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] This build requires lockable resources
- [x] This project is parameterized

String Parameter

Name: Application

Default Value: https://classic.cmpro.com/index.html


Dashboard › Rutaj Anil Damodare › My Views › All › exp 7 ›

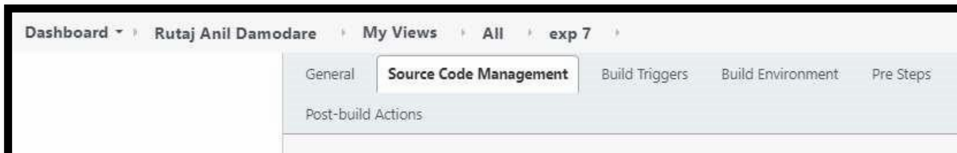
General | Source Code Management | Build Triggers | Build Environment | Pre Steps
Post-build Actions

Choice Parameter

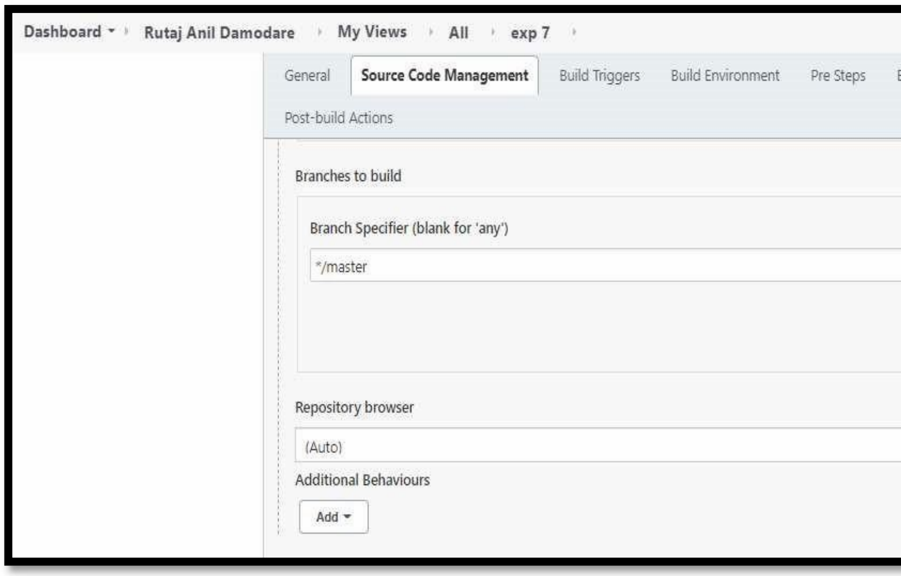
Name: Browser

Choices:
```
Chrome
Firefox
IE
```

Description


Dashboard › Rutaj Anil Damodare › My Views › All › exp 7 ›

General | Source Code Management | Build Triggers | Build Environment | Pre Steps
Post-build Actions



Dashboard › Rutaj Anil Damodare › My Views › All › exp 7 ›

General | Source Code Management | Build Triggers | Build Environment | Pre Steps

Post-build Actions

Branches to build

Branch Specifier (blank for 'any')

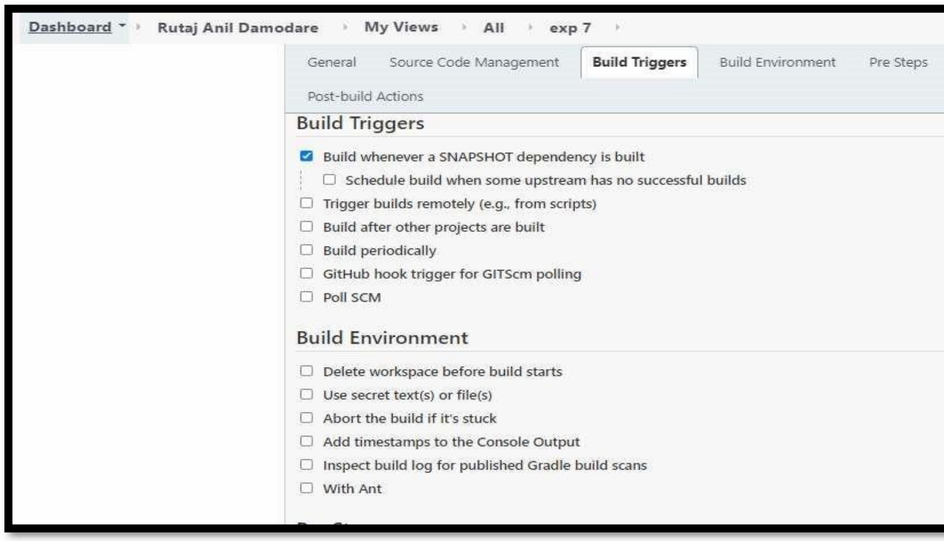
*/master

Repository browser

(Auto)

Additional Behaviours

Add



Dashboard › Rutaj Anil Damodare › My Views › All › exp 7 ›

General | Source Code Management | Build Triggers | Build Environment | Pre Steps

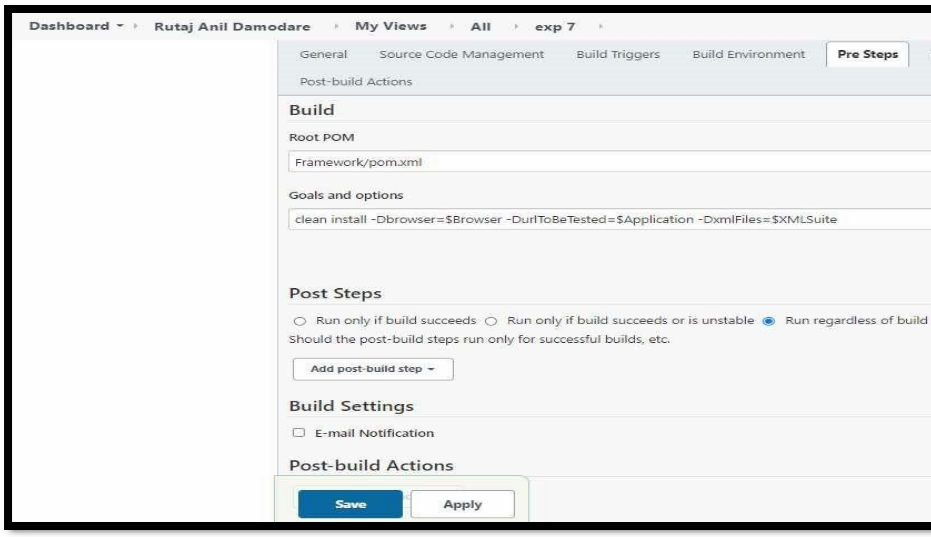
Post-build Actions

Build Triggers

- [x] Build whenever a SNAPSHOT dependency is built
  - [ ] Schedule build when some upstream has no successful builds
- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [ ] Build periodically
- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Build Environment

- [ ] Delete workspace before build starts
- [ ] Use secret text(s) or file(s)
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Inspect build log for published Gradle build scans
- [ ] With Ant



Build

Root POM: Framework/pom.xml

Goals and options: clean install -Dbrowser=$Browser -DurlToBeTested=$Application -DxmlFiles=$XMLSuite

Post Steps

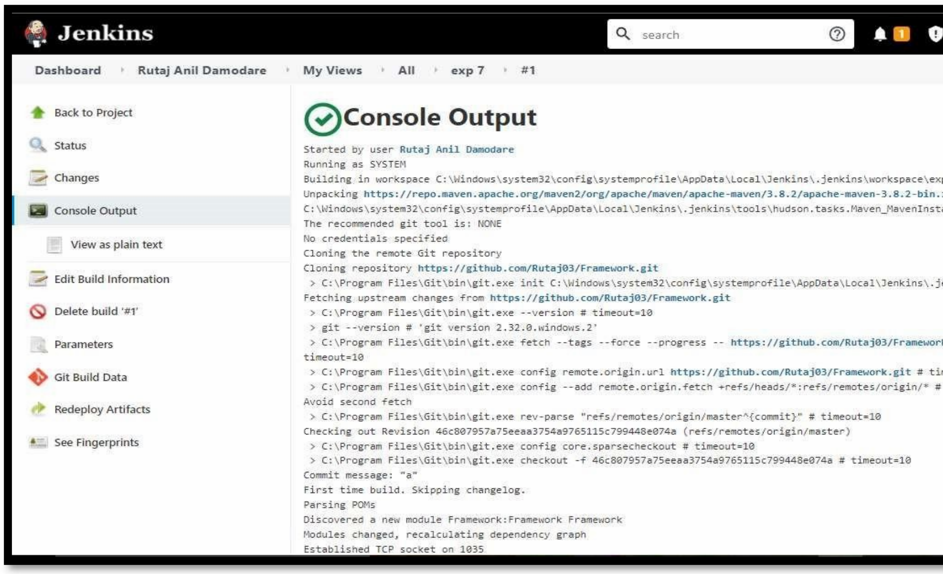
○ Run only if build succeeds ○ Run only if build succeeds or is unstable ● Run regardless of build

Should the post-build steps run only for successful builds, etc.

Build Settings

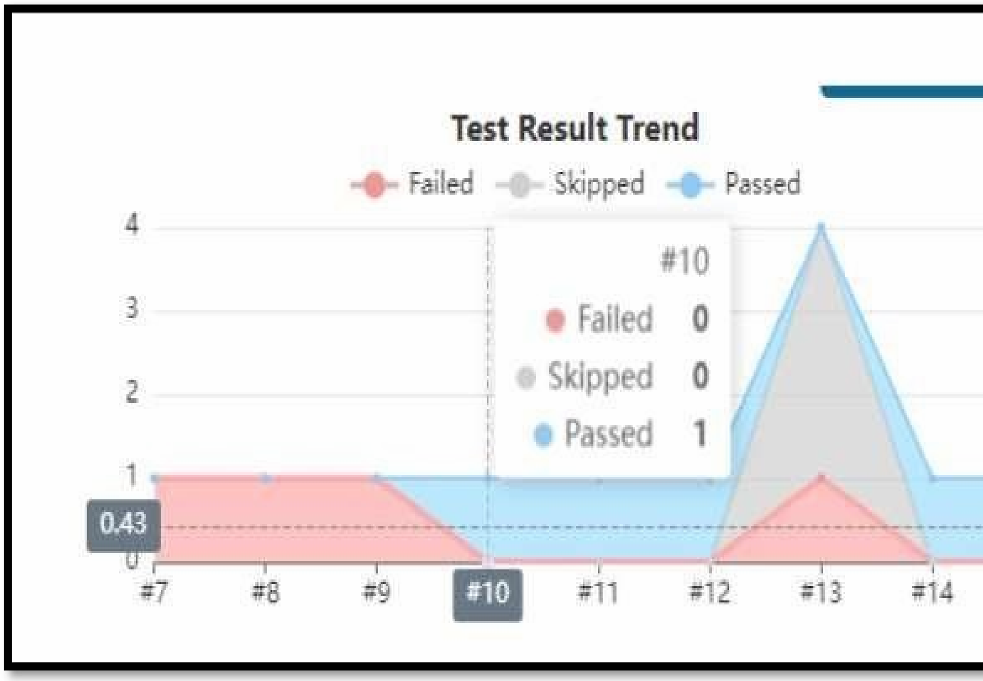
☐ E-mail Notification

Post-build Actions



Console Output

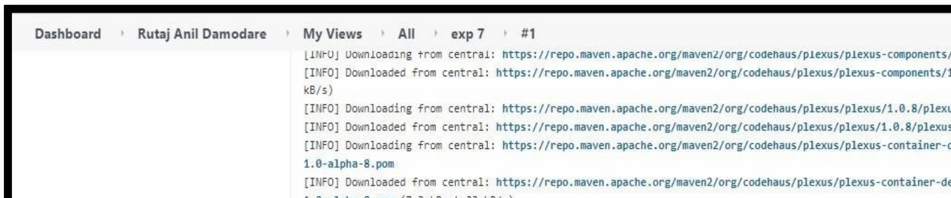
```
Started by user Rutaj Anil Damodare
Running as SYSTEM
Building in workspace C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\exp
Unpacking https://repo.maven.apache.org/maven2/org/apache/maven/apache-maven/3.8.2/apache-maven-3.8.2-bin.
C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\tools\hudson.tasks.Maven_MavenInsta
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Rutaj03/Framework.git
 > C:\Program Files\Git\bin\git.exe init C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins\.j
Fetching upstream changes from https://github.com/Rutaj03/Framework.git
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > git --version # 'git version 2.32.0.windows.2'
 > C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/Rutaj03/Framewor
timeout=10
 > C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/Rutaj03/Framework.git # ti
 > C:\Program Files\Git\bin\git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* #
Avoid second fetch
 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 46c807957a75eeaa3754a9765115c799448e074a (refs/remotes/origin/master)
 > C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
 > C:\Program Files\Git\bin\git.exe checkout -f 46c807957a75eeaa3754a9765115c799448e074a # timeout=10
Commit message: "a"
First time build. Skipping changelog.
Parsing POMs
Discovered a new module Framework:Framework Framework
Modules changed, recalculating dependency graph
Established TCP socket on 1035
```



Conclusion: Thus, we successfully
Jenkins using Maven.

SIGN AND REMARK:
DATE:

R1 (3 Marks)	**R2 (5 Marks)**



```
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1
kB/s)
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexu
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexus
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-d
1.0-alpha-8.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-de
```