

Last updated: Dec 7, 2021

Selenium Java Training - Session 35 - Git, GitHub and SauceLabs

Git and GitHub

1. In real time, we work as a team, hence we need Git and GitHub
2. Create an account at GitHub, where we can host our code
3. Create a new repository say 'GitDemoRepo'
4. At a high level Git is used to move the code between local machines and GitHub Repo
5. Download git and install
6. Open Git Command Prompt and trigger the below commands:
 1. Create a local folder in your System for Git Repos
 1. Navigate to this folder using command prompt
 2. Initialize the git folder as local git repository
 1. git init
 2. .git will be created to confirm this
 3. Register yourself with Git
 1. git config --global user.name "Arun"
 2. git config --global user.email "arun.motoori@gmail.com"
 4. Clone the GitHub repo to the local repo
 1. git clone <https://github.com/arunmotoori/GDR.git>
 5. Go inside the cloned project folder
 1. cd GDR
 6. Copy all the Project files to the Local Git Repo folder
 7. We have to commit before pushing the code to GitHub
 1. There are two levels of commit
 1. staging and commit
 2. Adding all the project files to staging
 1. git add *
 3. Committing the code
 1. git commit -m "first commit"
 8. Giving the address of GitHub where we need to push the local repository code
 1. git remote add origin git@github.com:arunmotoori/GDR.git
 2. We can find this command ready for us in the GitHub page
 9. Push the code to GitHub
 1. git push origin main
 10. Import this code into Eclipse IDE:
 1. Launch Eclipse IDE from a different workspace
 2. Import the Project from GitLocalRepo
 3. Do some changes to the LocalRepo code
 11. Check the changes, add to staging and commit

4. git commit -m "second commit comment"
5. git status
6. git push origin main
12. Modify directly from GitHub - Assuming other person has changed code
 1. Get the latest code
 1. git pull origin main
13. Create a new branch and switch to it
 1. git checkout -b sbranch (create branch and switch)
 2. Note: git checkout sbranch (Will only switch but not create)
 3. git branch (To check the current branch)
 4. Update some code and push to subbranch
14. Switch to master branch
 1. git checkout main
 2. git pull origin master (Get the latest code from master)
 3. git status (Once everything clear we can go to next step)
15. Merge the branch to main
 1. git merge devbranch
 2. Merges to the active branch

Jenkins (Continued)

1. Launch jenkins from command line using the below command
 1. keep jenkins.war in any folder
 2. Run the jenkins using the command - java -jar jenkins.war
2. Access localhost:8080 and login using the below credentials
 1. arunmotoori
 2. 12345
3. Install Maven Integration Plug-in in Jenkins
4. Create a new Maven Job in Jenkins
5. Select 'Git' under 'Source Code Management' and give the page URL path of GitHub Repo
6. Uncheck any selected checkbox options under 'Build Triggers'
7. Under post steps > select invoke top level Maven targets
8. Apply and Run the Job
9. Explain about build periodically

SauceLabs for Cloud Testing

1. Why Cloud?
2. Create a SauceLabs account
 1. arunmotoori
 2. Second@123
3. Create a new Maven Project

2. Search for 'SauceLabs Platform Configurator' and get the auto-generated capabilities code
3. Paste the auto-generated code
4. Search for 'saucelabs getting started with selenium website testing'
5. And copy paste three lines of code directly inside the class
6. Modify the username in the code
7. Go to Account > User Settings and copy the Access Key and paste into the code
8. Write some sample selenium code
9. View the code [here](#)
10. Execute the code and watch under SauceLabs > Automated > Test Results

By,
Arun Motoori

