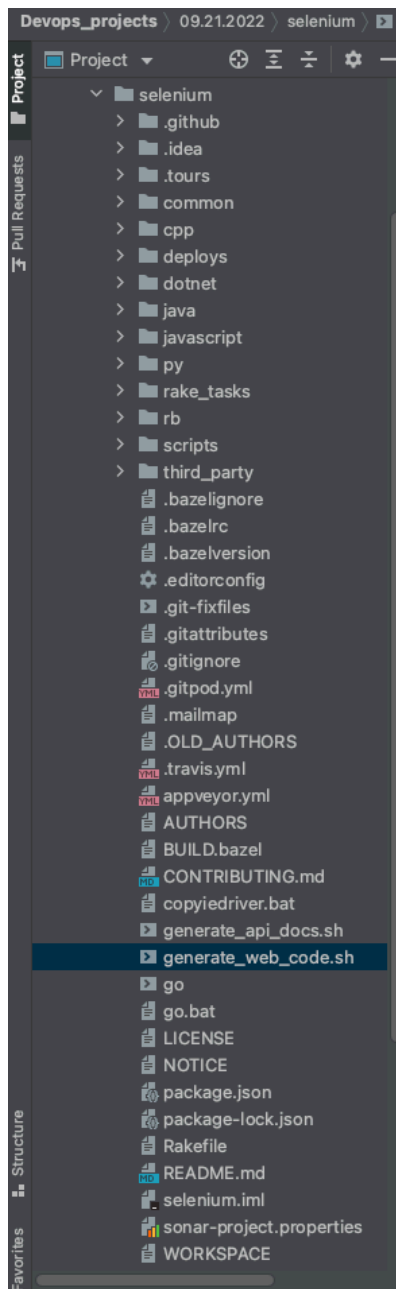


Git

1. Clone Selenium Webdriver project

Command: `git clone https://github.com/SeleniumHQ/selenium.git`



2. Clone Appium project.

Command: `git clone https://github.com/appium/appium.git`
(I didn't do it in real because it was quite heavy and it was the same as exercise nor 1)

3. Create a new Python project (any) and upload it to Github

`https://github.com/sarahkossmann/Devops_projects`

4. Create a branch in your project repository

```
(venv) sarahkossmann@Sarahs-MacBook-Pro:Devops_projects/09.21.2022 <main*>$ git branch a-branch
(venv) sarahkossmann@Sarahs-MacBook-Pro:Devops_projects/09.21.2022 <main*>$ git checkout a-branch
M      .idea/vcs.xml
Switched to branch 'a-branch'
(venv) sarahkossmann@Sarahs-MacBook-Pro:Devops_projects/09.21.2022 <a-branch*>$ git checkout main
M      .idea/vcs.xml
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
(venv) sarahkossmann@Sarahs-MacBook-Pro:Devops_projects/09.21.2022 <main*>$
```

5. Create a conflict between your Master and branch

```
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <main>$ git merge a-branch
CONFLICT (add/add): Merge conflict in 09.21.2022/conflict_file.txt
Auto-merging 09.21.2022/conflict_file.txt
Automatic merge failed; fix conflicts and then commit the result.
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <main*>$ git add .
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <main*>$ git commit -m "conflicts solved"
[main 3eeb5be] conflicts solved
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <main>$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 407 bytes | 407.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:sarahkossmann/Devops_projects.git
   f8cd62a..3eeb5be  main -> main
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <main>$
```

6. Enter detached head status

```
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <main>$ git checkout 3eeb5be8bd123996edc914715276951bc4bbe074
Note: switching to '3eeb5be8bd123996edc914715276951bc4bbe074'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 3eeb5be conflicts solved
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <3eeb5be>$
```

7. Reset (hard) to a specific commit

```
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <main>$ git reset --hard 3eeb5be8bd123996edc914715276951bc4bbe074  
HEAD is now at 3eeb5be conflicts solved  
(venv) sarahkossmann@Sarahs-MacBook-Pro:PycharmProjects/Devops_projects <main>$
```

2.

