# Access to GIT

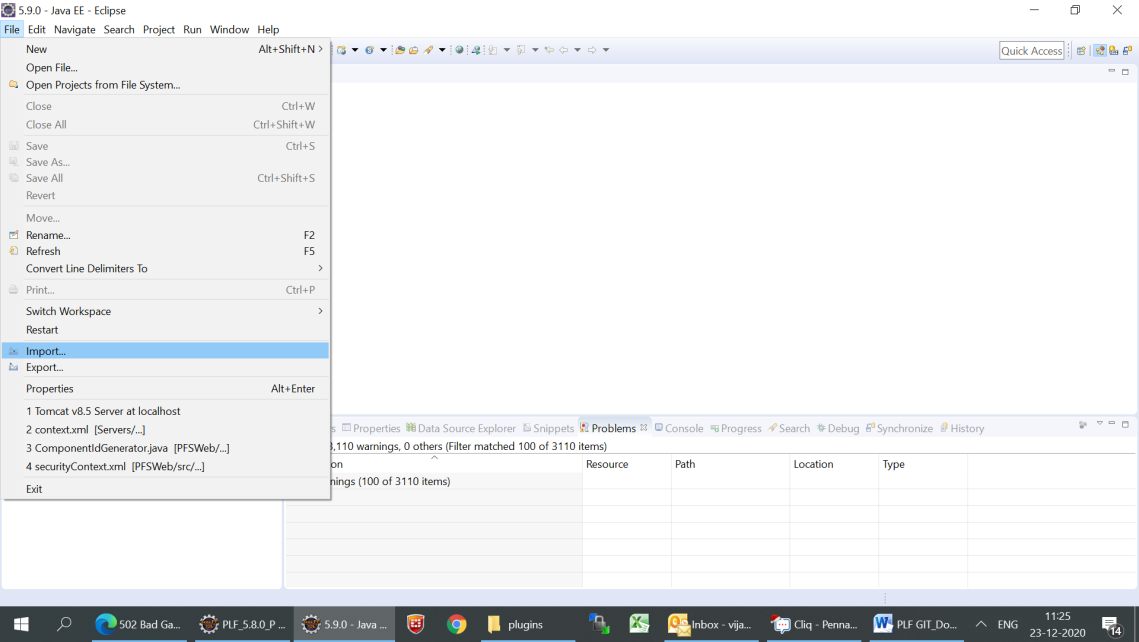
To access the GIT repository need to login to git hub for the first time using the individual LDAP credentials in the below link

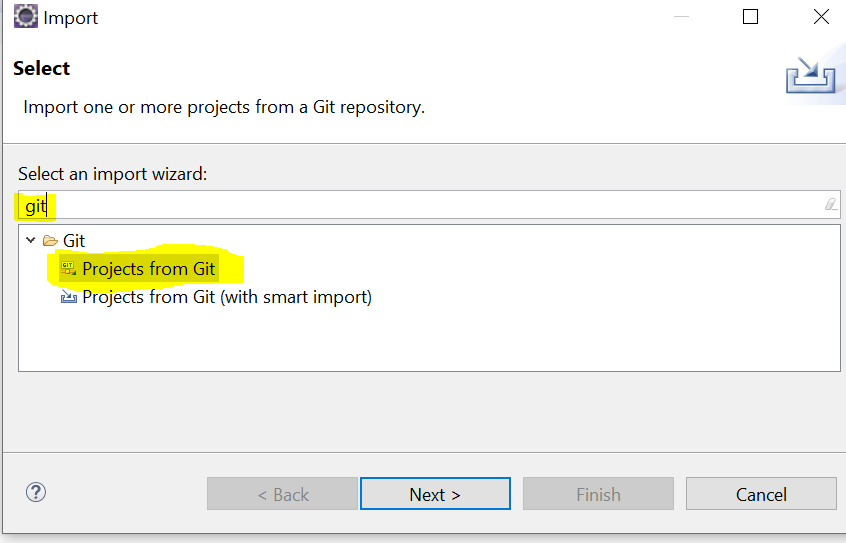
<http://gitserver/>

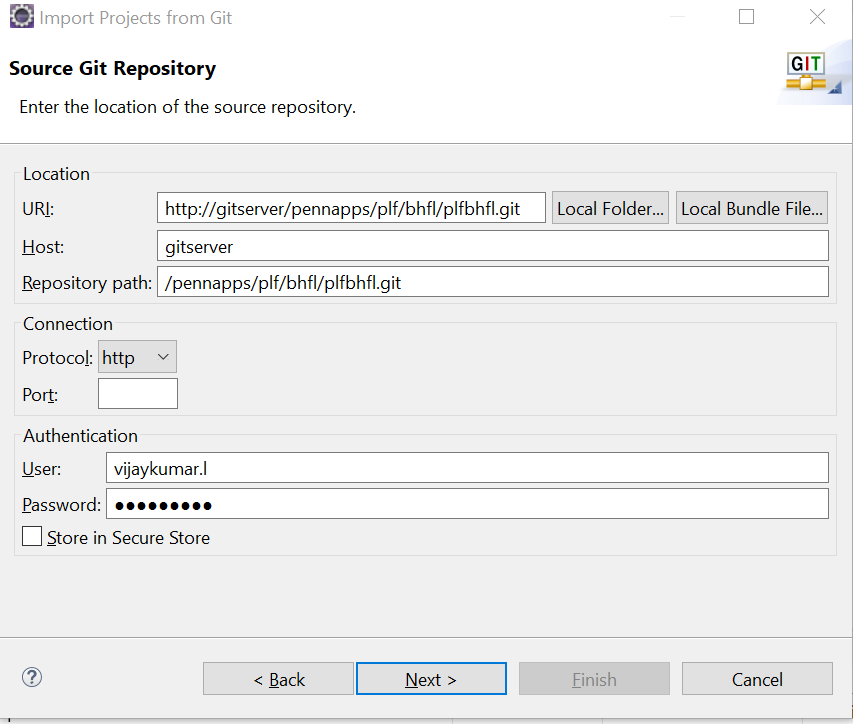
After successful login to git server using LDAP credentials Organization owners can manage individual and team access to the organization's repositories. Team maintainers (leads) can also manage a team's repository access.

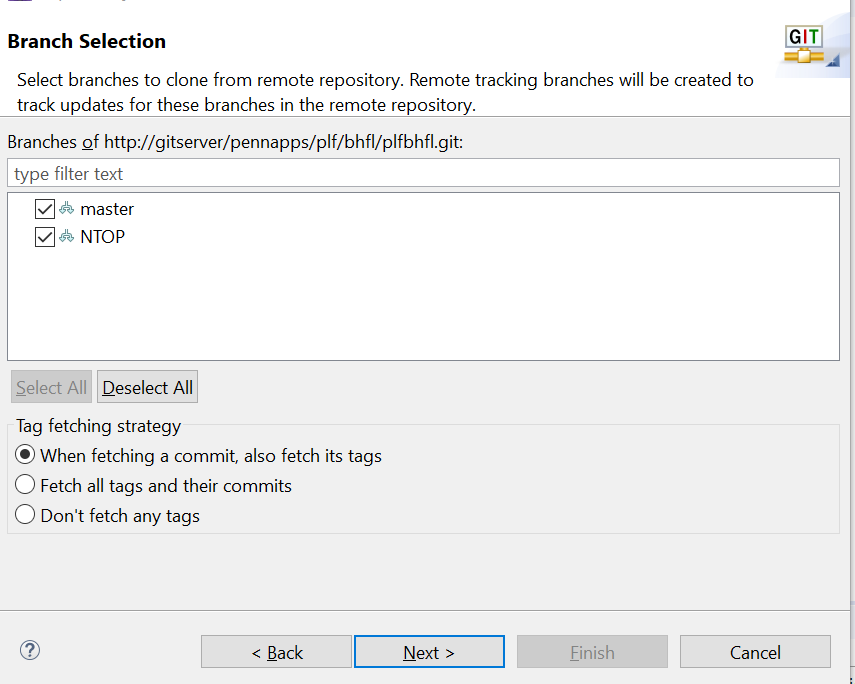
# Check out Process

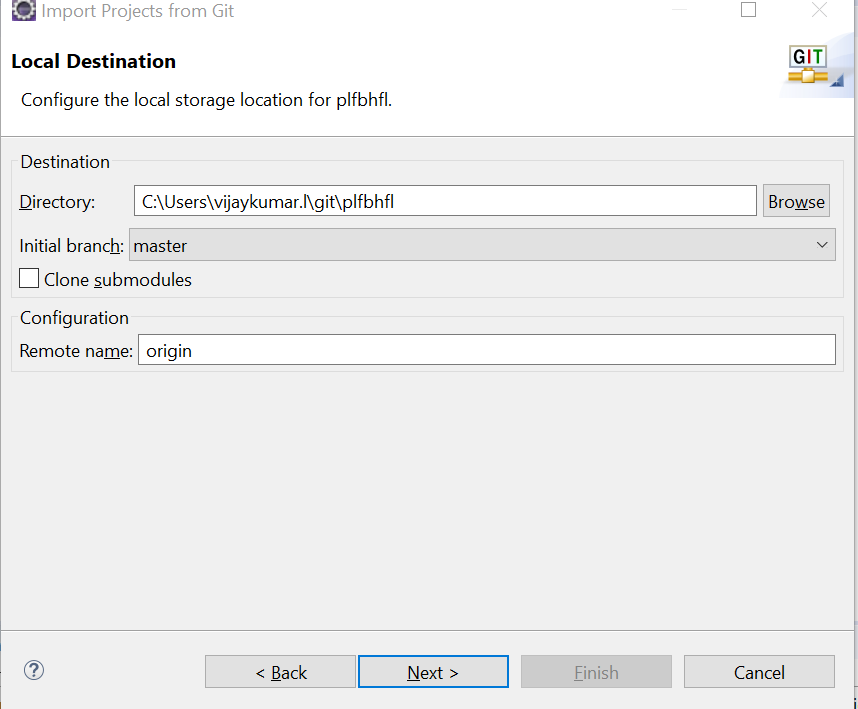
* To clone the git repo we need approval access of dev. role or Guest role required.
* Check out and adding files to .git ignore
* To check out click on File 🡪Import

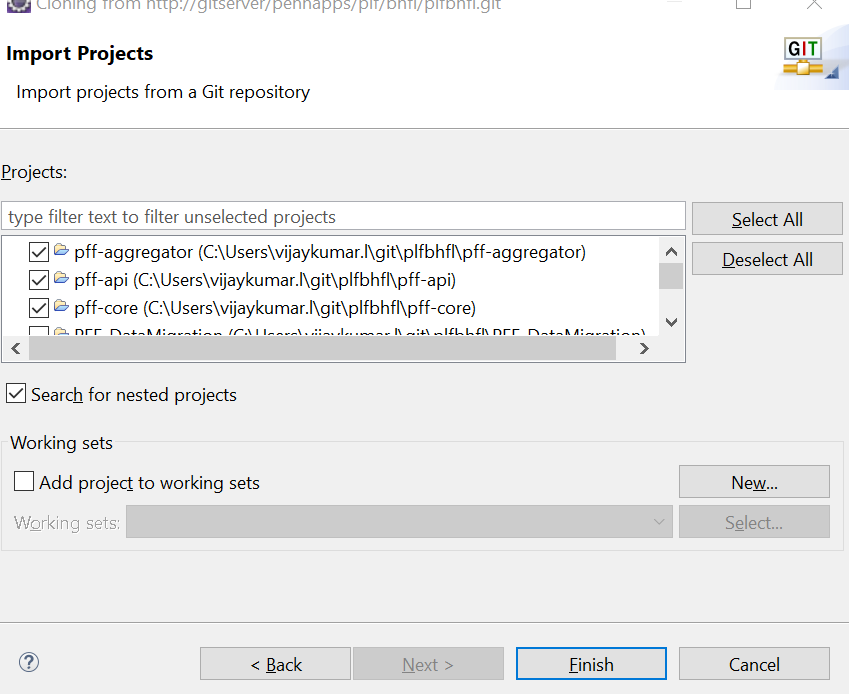


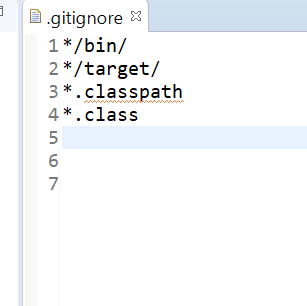












Update maven project after add below properties in .gitignore file and check team synchronise

\*.DS\_Store

bin

bin/src/main/

bin/.svn/

bin/.settings

bin/target/

target

.classpath

.project

.settings

bin

classes/

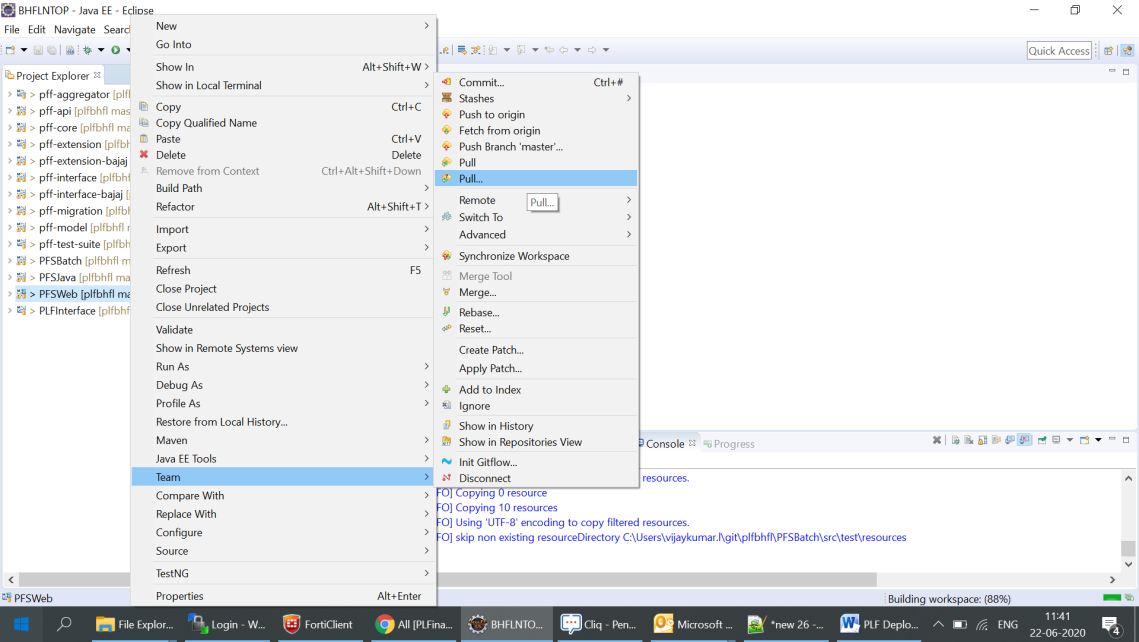
/build

target/

\*.iml   
.gitignore

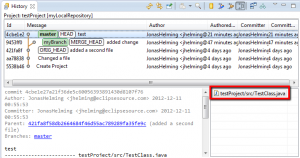
# Synchronizing workspace

Right-click on a project in the Package Explorer and select Team > Pull or right-click on a repository in the Git Repositories view and select Pull to pull new changes from the upstream branch your local branch is tracking. This also works if resources are selected from more than one repository.



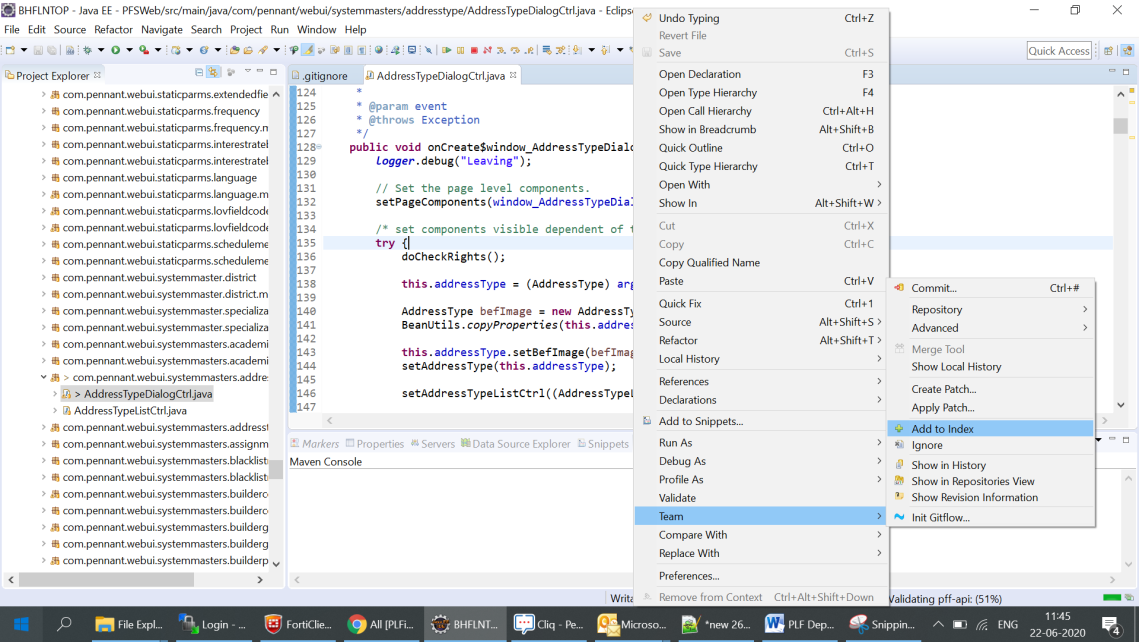
# History View

To show any shared file’s history, right click it and select *Team => Show in History*. This will open the History View, giving an overview of the commits and allowing you to perform several actions (compare, creating branches/tags, reset…). Every commit you select comes up with a revision comment and revision details. The revision comment (bottom left corner) includes parents, children, commit message and changes whereas the revision details (bottom right corner) name the changed files and the actions performed upon them (A=ADD, M=MODIFY, D=DELETE). Selecting a file in the revision details will scroll the revision comment to the changes to that file.

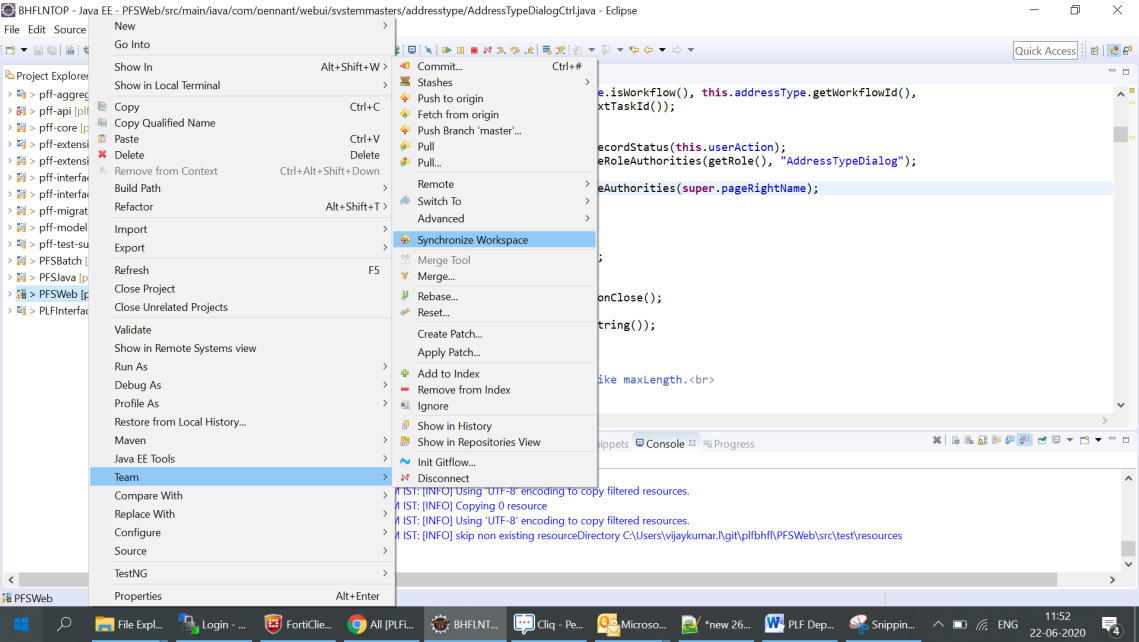
[](https://eclipsesource.com/wp-content/uploads/2012/12/23.png)

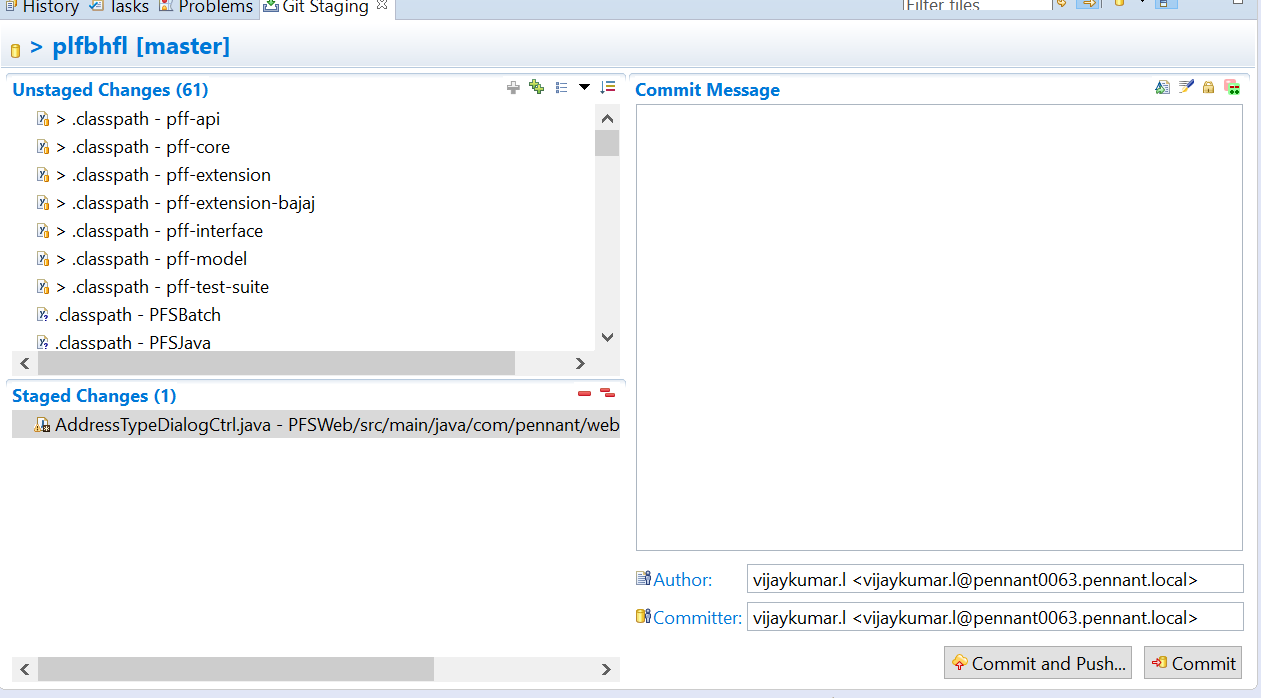
# Commit

* Open the *Git Staging* view, via Window Show View Other…​  Git  Git Staging. In this view drag all files into the *Staged Changes* area, write a meaningful commit message and press the commit button.
* If you need any code changes one file, use add to index for staging file to commit.

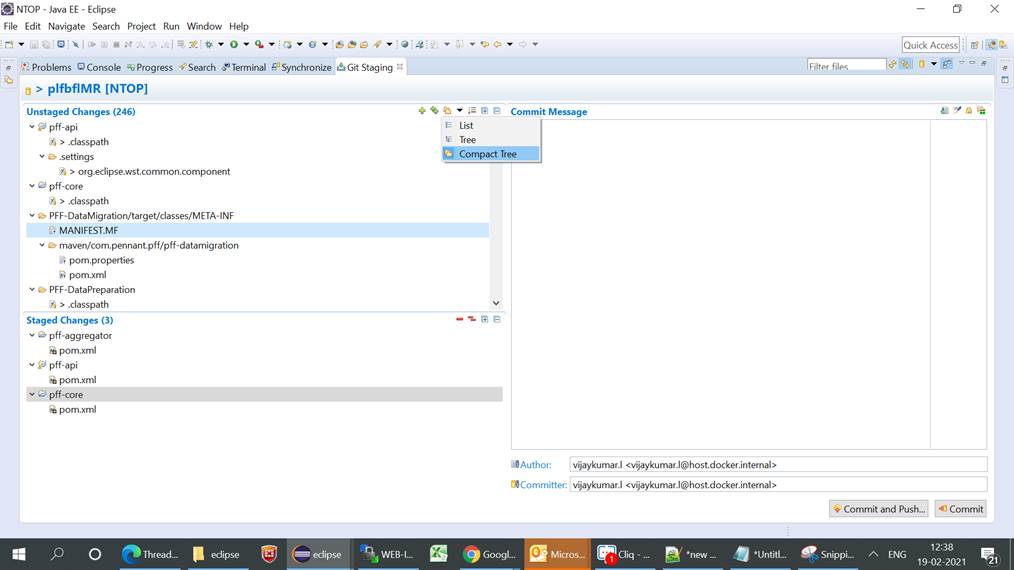


* Right to click on project, use option Team and use Synchronise workspace as shown below .

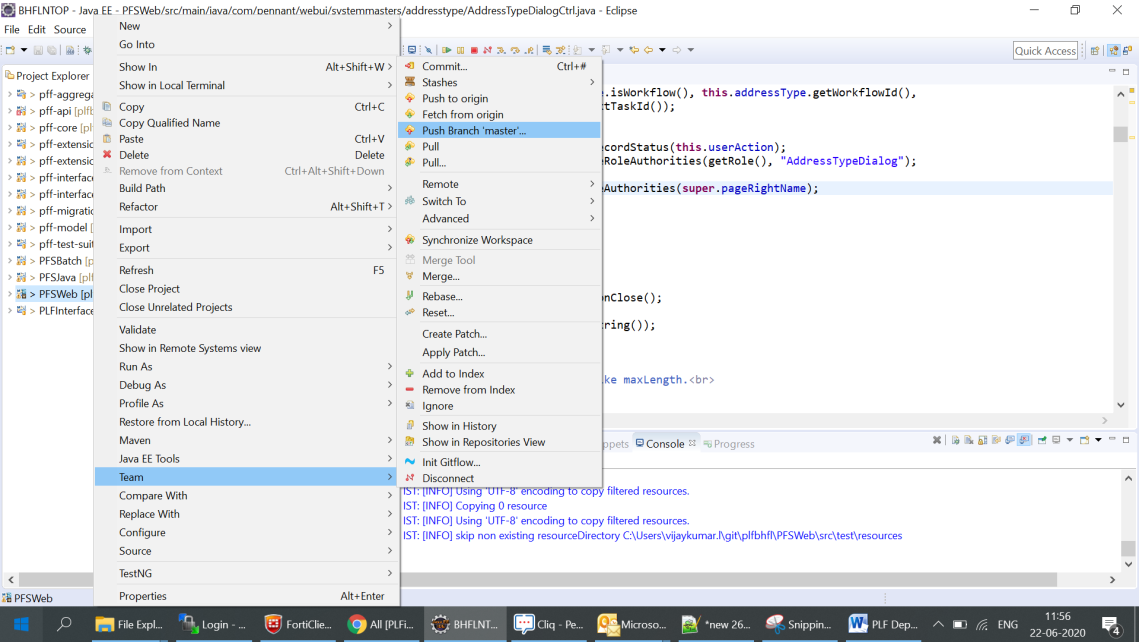




In Git staging , please select compact tree or tree

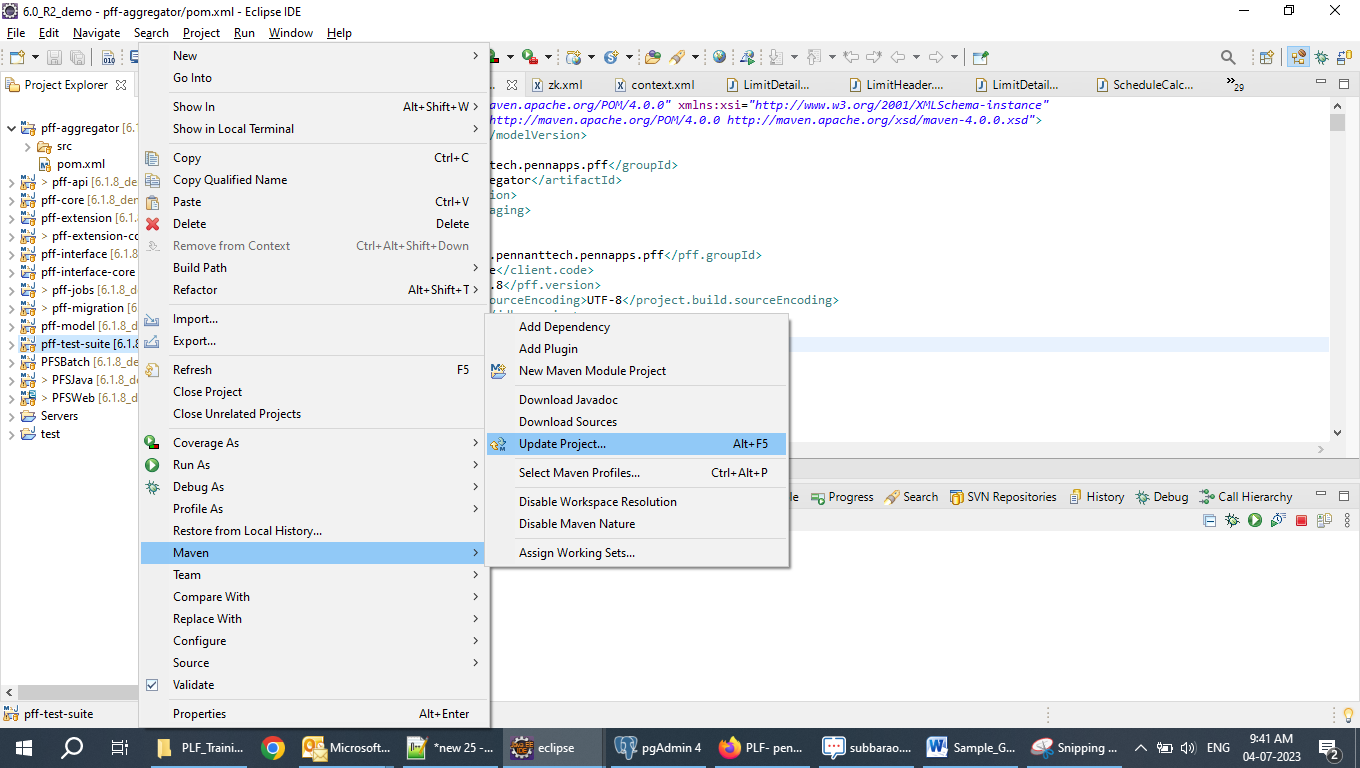


* Commit the changes to local repository and later push changes to required branch as below .

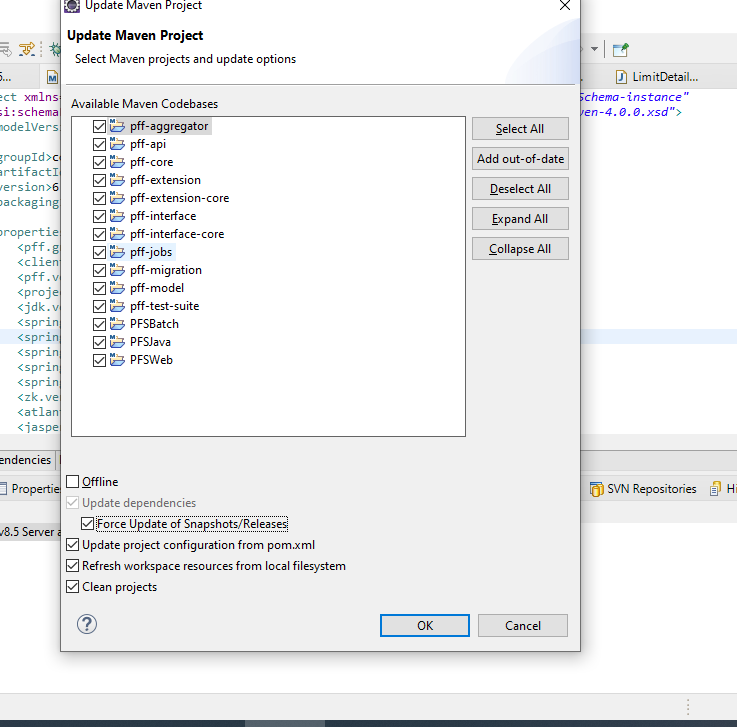




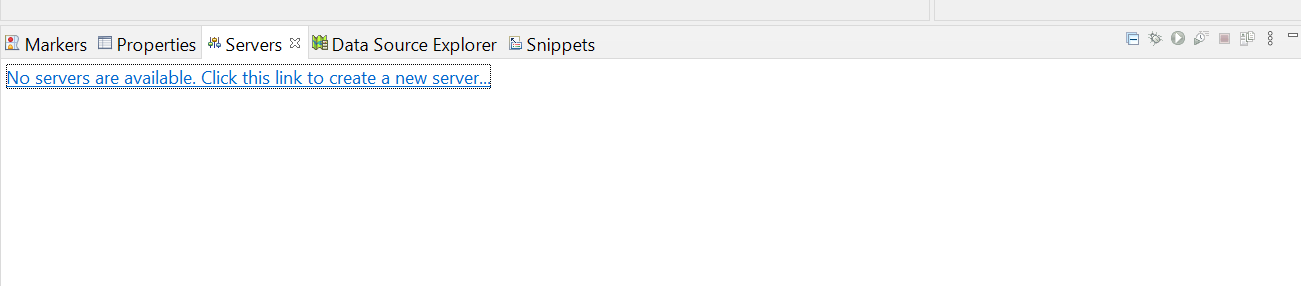
# Maven update.



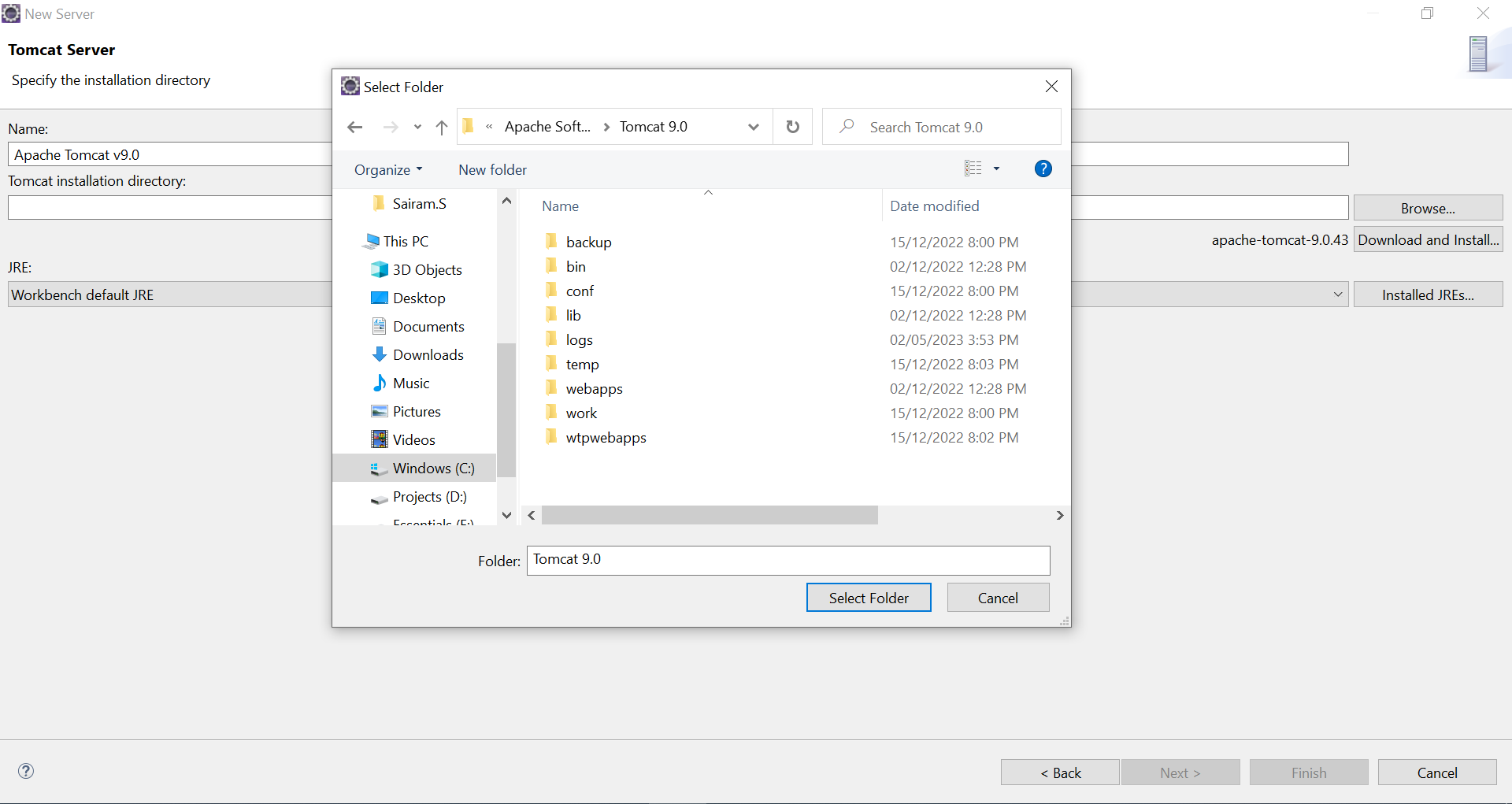
Click on select all and force update Snapshots/Releases and click on OK and wait till the maven update is completed as per below SS



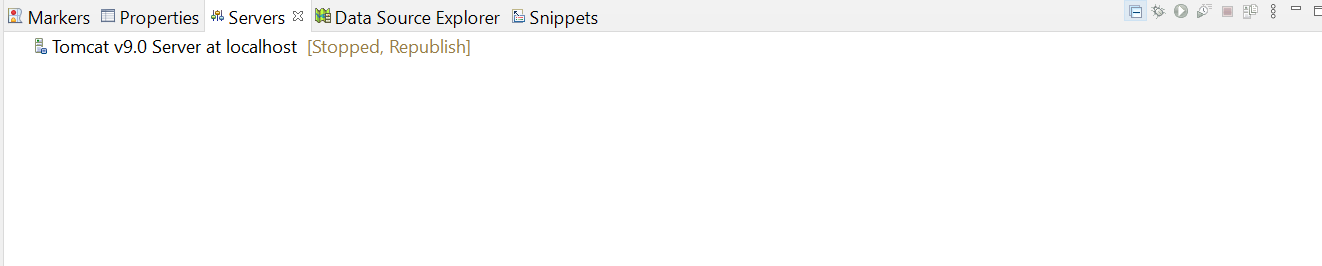
# Adding Tomcat Server.



Click on add a new server and select the respective tomcat version

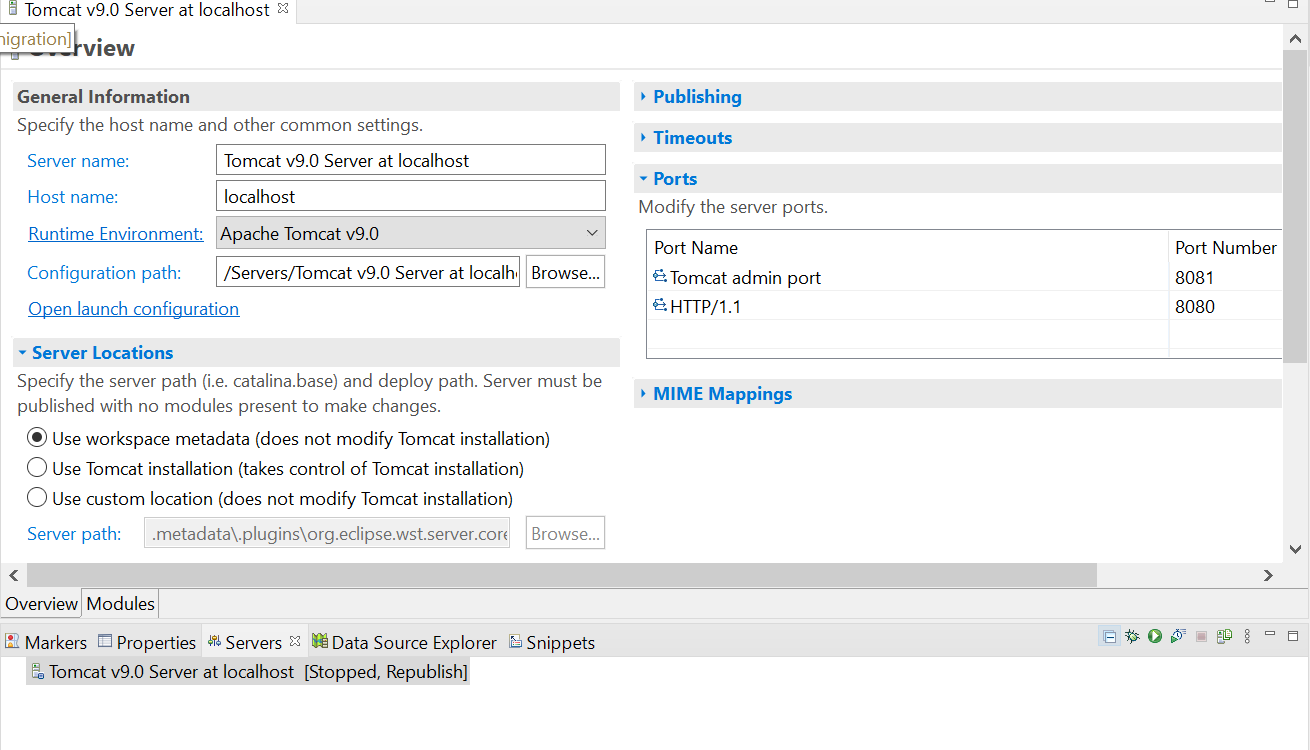


Click on Finish.

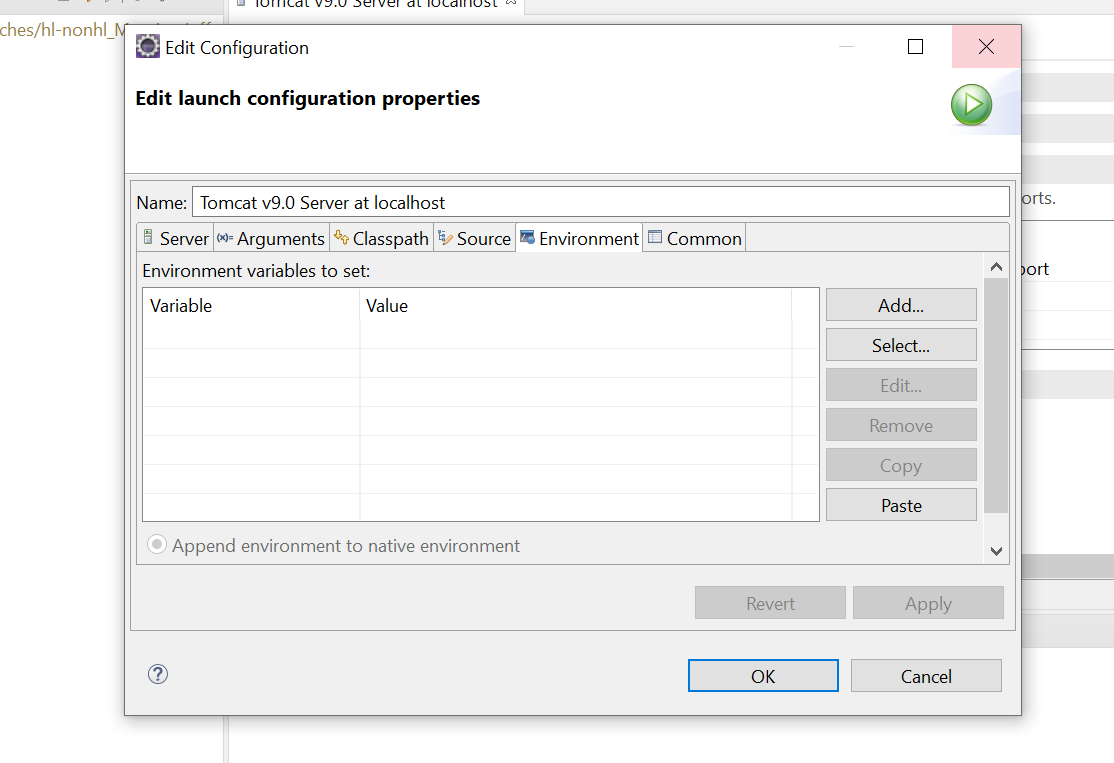


Adding the environmental variables to the tomcat server.

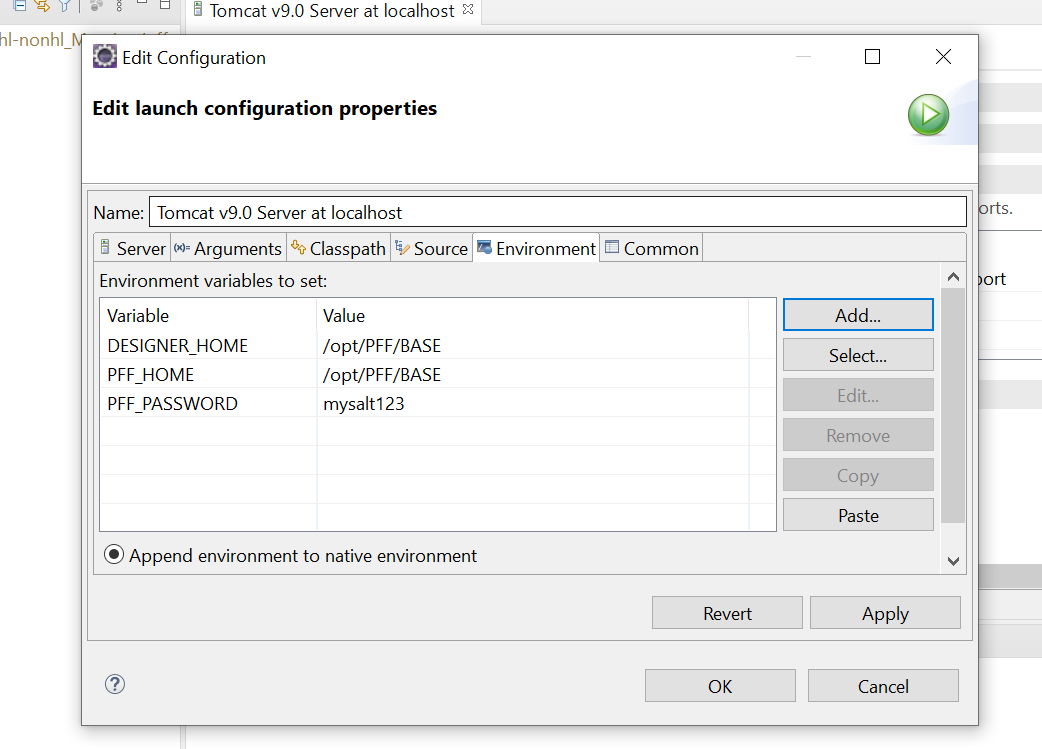
Double click on the server.



Click on the open launch configuration.



Click on Add and provide the PFF\_HOME,PFF\_PASSWORD,DESIGNER\_HOME variables as shown below.



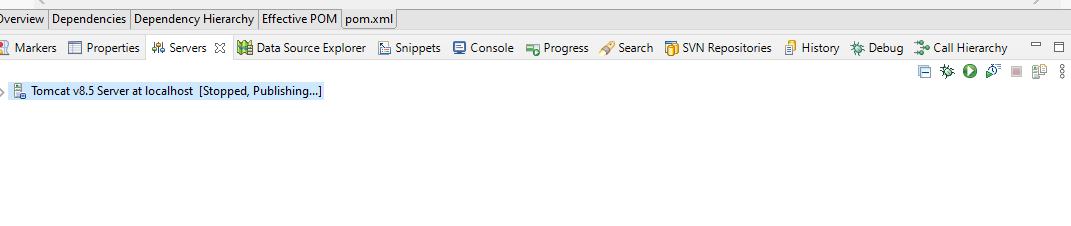
Click on OK to save the changes.

# Adding Tomcat Server context changes.

Open the context.xml file and add the changes as shown below. Context file is shared in the download link.



# Application Start:



After the application start up is completed below is the URL and credentials to be used

Url : <http://localhost:8080/PFSWeb/loginDialog.zul>

Credentials : admin / test

