

To develop app. we write different types of files in projects (like java, xml, jsp etc..). All these files are stored finally in a central server i.e. Code Repository. It maintains the details of code. like

- ✓ How many times code is modified?
 - ✓ Who modified what code(+/- added/removed)
 - ✓ Changes done on what date and time? Maintaining these details is known as Version Management/Version Control.
-
- a. In developement mostly used tool is GIT(gitHub).
 - b. It maintains two level repository process for cross verify and commit.
 - c. For this we need to create an account in github.com(using email id and verify email after creation).
 - d. It maintains staging area to select /verify/comment the file while adding to local repository.

Git Process:

Git Repository Types:

- a. Public : Free for every one
- b. Private : Paid version.

Remote Repository(Steps):

- a. goto <https://github.com/>
- b. SignUp(Register) with details
- c. Goto Email -and verfiy Link
- d. login with un and pwd
- e. Create Repository Type Public (Companies uses- Private Type)
- f. Copy Git Link: <https://github.com/abcd/testVen.git> it is secured with un,pwd.

Eclipse Workspace and Local Repository Steps:

In eclipse,

1. Go to window menu

2. Show Views search for "GIT" select GitRepository and GitStaging.
3. Click on Window
4. showView
5. History.

Creating local repository:

1. Right Click on Project
2. choose "Team" option
3. Share project
4. select checkbox "create or use repository"
5. select option shown below
6. click create repository button
7. finish

GIT Operations with Flow:

- add file to git staging (Click-drag-drop).
- Commit and Push
- Paste Git Link in URI input box
- enter userName and password.
- next/next/finish.
- right click on project
- Choose team
- Select pull, then ,rebase to update local from remote.

Some Interview Questions over GIT:

1) Can you provide GIT/Client/Apps URLs? No, Sorry.all details what you are asking is private and confidential. I cannot provide those.

2)Who created your git account ?

Git administrator name-"XYZ", in my company. He handles all permission and auth. details.

3)What type of Account Repository areyou using?

It is private account handled by company.

4)When you check-in or push your code?

After Implementation of complete module and UnitTesting code in my local, I'll push to remote.

5)How will you find about one file all modification??

We use history option in eclipse

->right click->team->shown in history.

we can also choose any two files to compare with each other ex:select

any history options->right click->compare with each other.

6)How do you identify code modifications in GIT?

Using Symbols (+) code is added (-) code is removed.

7)What is the difference between commit and push?

commit: update code from workspace to Local Repository

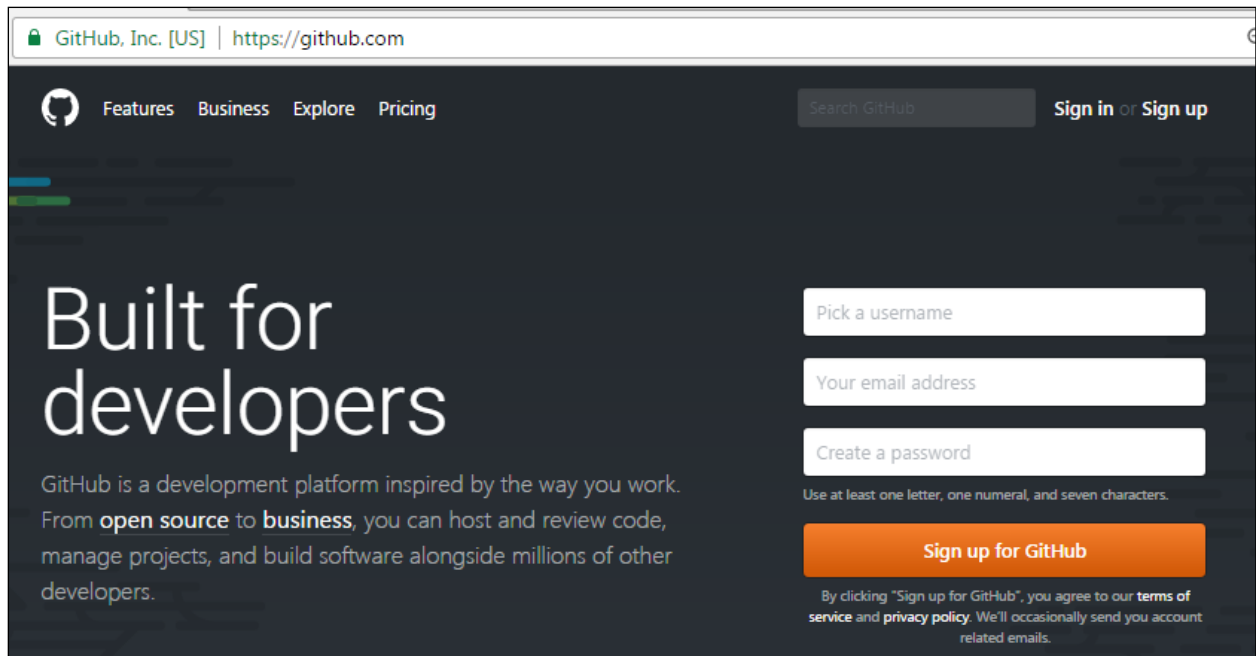
push :Update code from local to remote repository.

8)How can you update you workspace/local with remote?

By using pull and rebase operations.

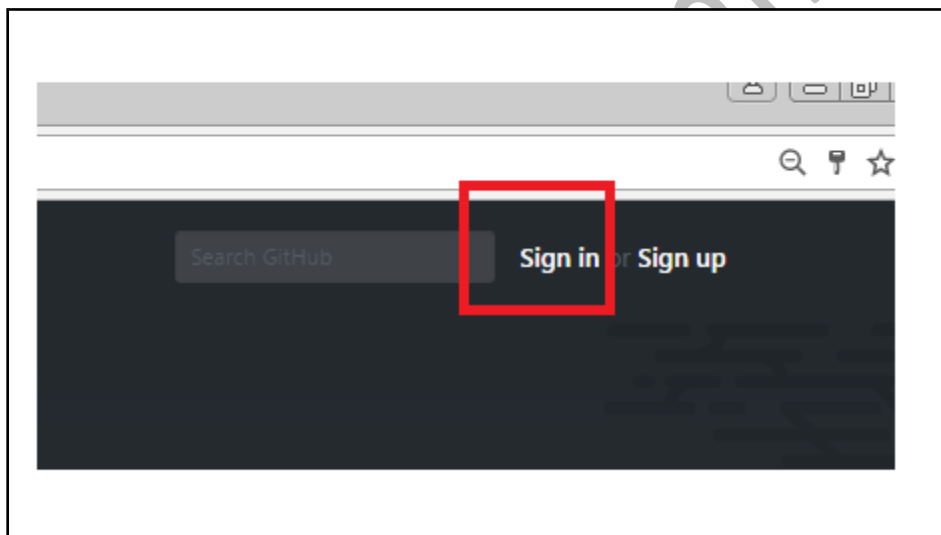
Screens Help For Complete Process:

<https://github.com/>

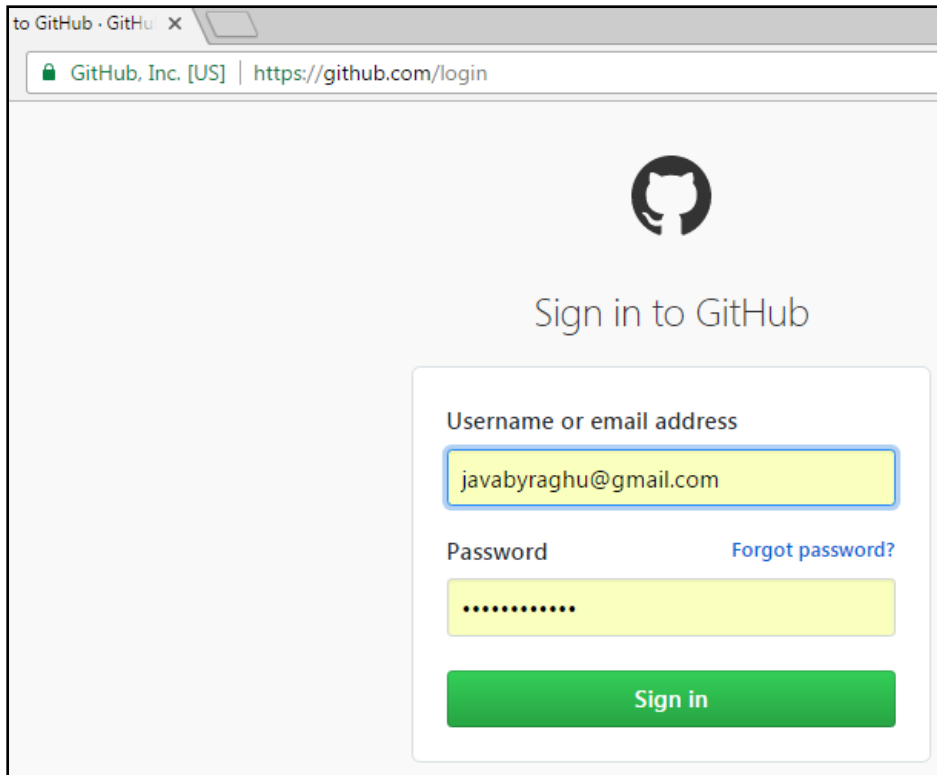


The screenshot shows the GitHub homepage with a dark theme. The header includes the GitHub logo, navigation links (Features, Business, Explore, Pricing), a search bar, and links for 'Sign in' or 'Sign up'. The main content area features the text 'Built for developers' and a description of GitHub as a development platform. On the right side, there is a sign-up form with three input fields: 'Pick a username', 'Your email address', and 'Create a password'. Below the password field, there is a note: 'Use at least one letter, one numeral, and seven characters.' An orange button labeled 'Sign up for GitHub' is positioned below the form. At the bottom of the sign-up section, there is a disclaimer: 'By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.'

Goto email account and click on verify link. Click on sign in



Enter user name and password.



to GitHub · GitHub ×

GitHub, Inc. [US] | <https://github.com/login>

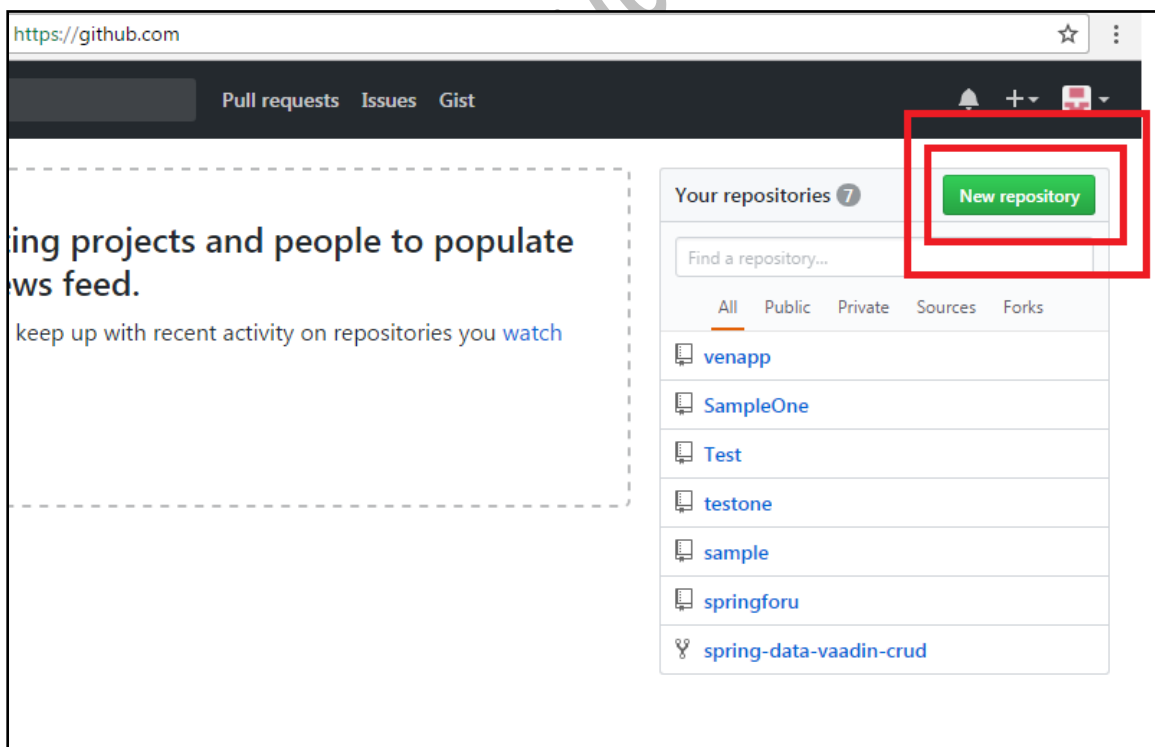
Sign in to GitHub

Username or email address

Password [Forgot password?](#)

[Sign in](#)

Click on new Repository



<https://github.com>

Pull requests Issues Gist

ing projects and people to populate
ws feed.

keep up with recent activity on repositories you [watch](#)

Your repositories 7

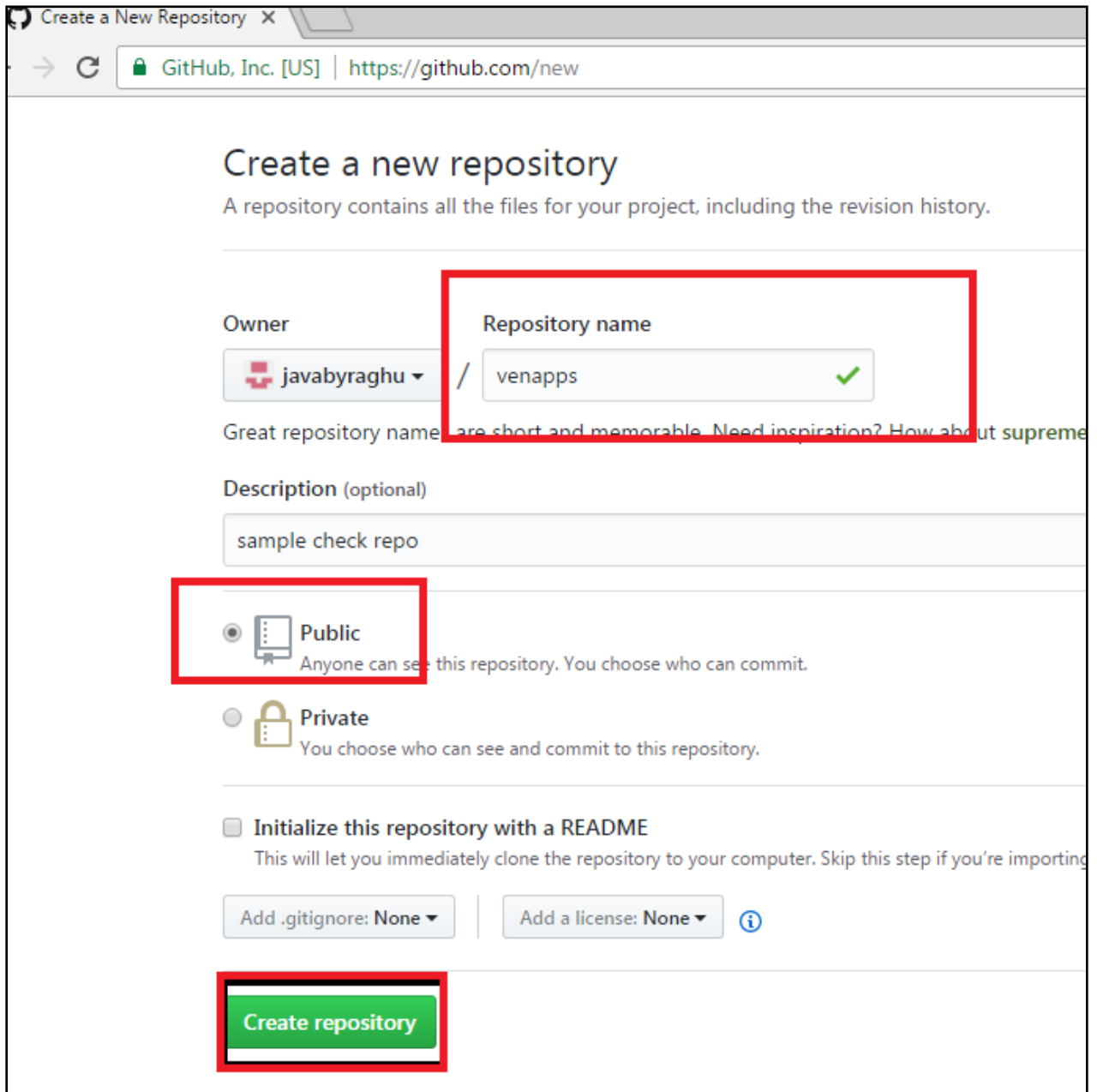
[New repository](#)

Find a repository...

All Public Private Sources Forks

- [venapp](#)
- [SampleOne](#)
- [Test](#)
- [testone](#)
- [sample](#)
- [springforu](#)
- [spring-data-vaadin-crud](#)

Provide Repository Details




Create a New Repository

GitHub, Inc. [US] | <https://github.com/new>


Create a new repository


A repository contains all the files for your project, including the revision history.

Owner:  javabyraghu / Repository name: ✓

Great repository names are short and memorable. Need inspiration? How about **supreme**

Description (optional):

☒  **Public**
Anyone can see this repository. You choose who can commit.

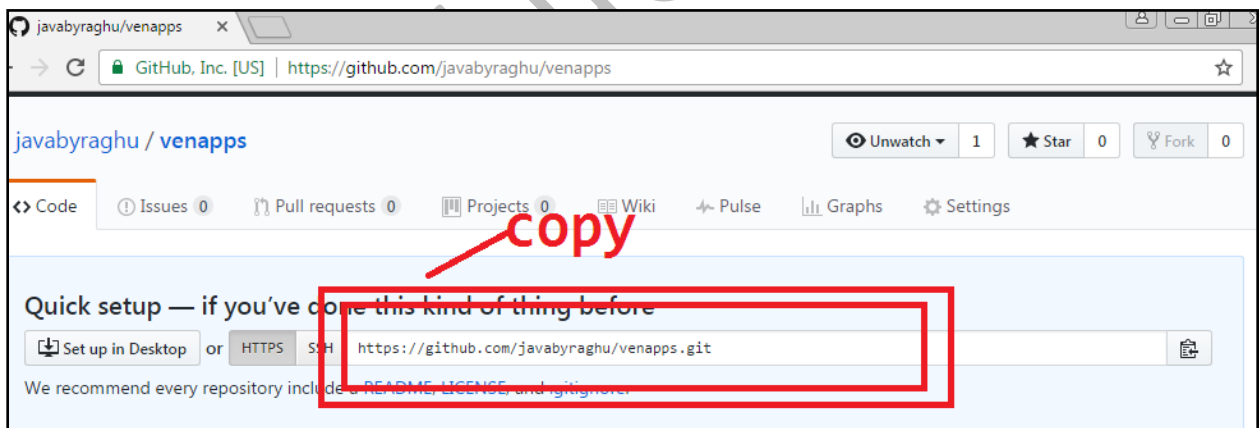
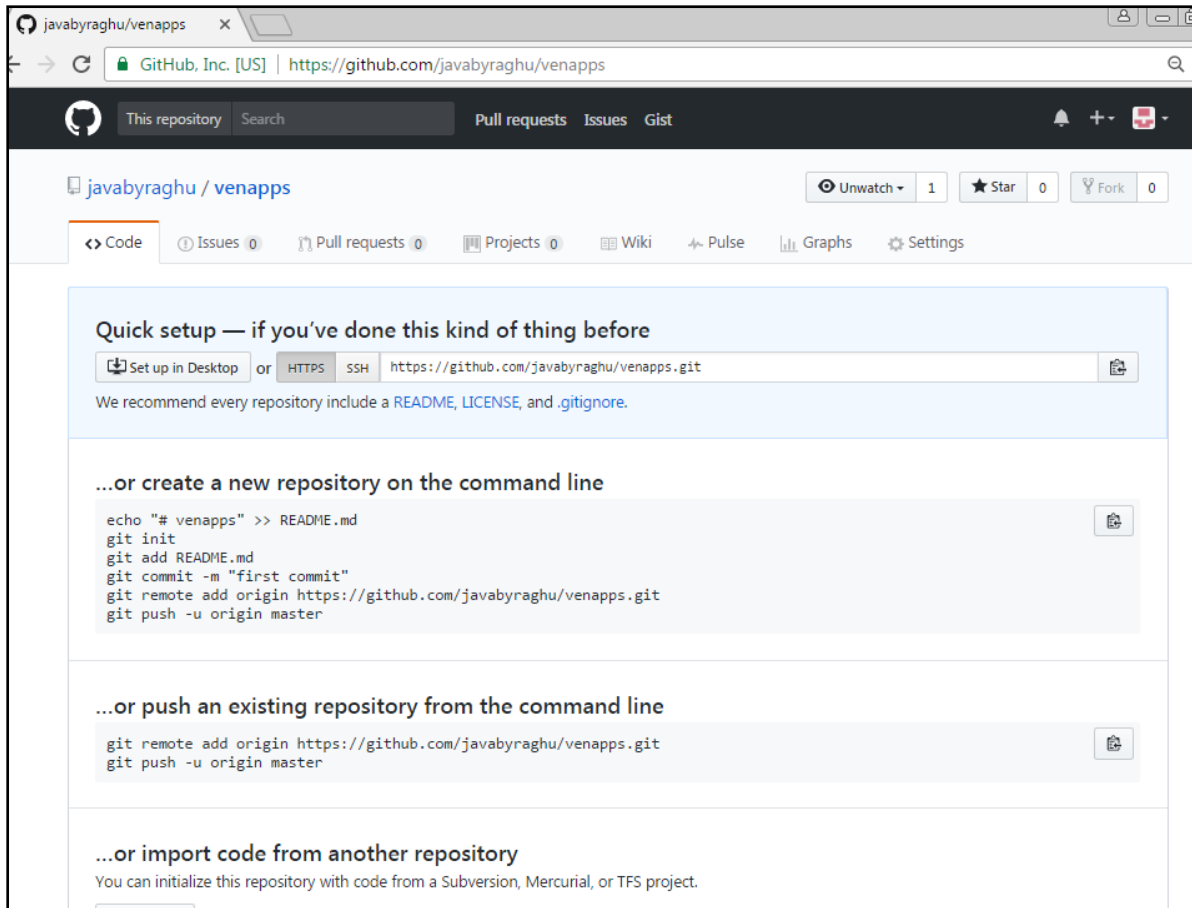
☐  **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

On click create repository

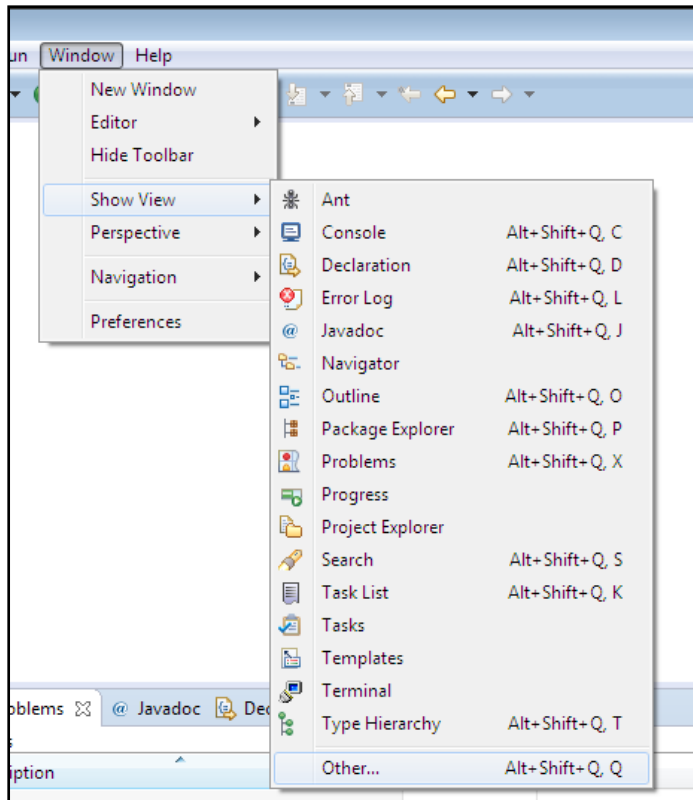


Ex : (Repository Link)

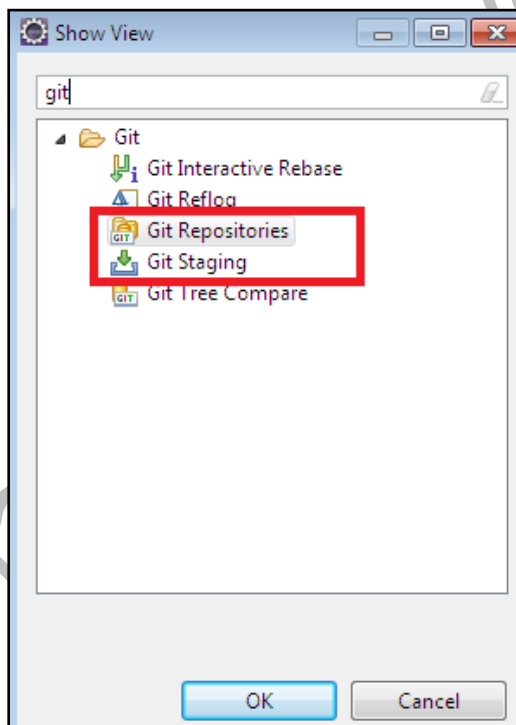
<https://github.com/javabyraghu/venapps.git>

Got Eclipse:

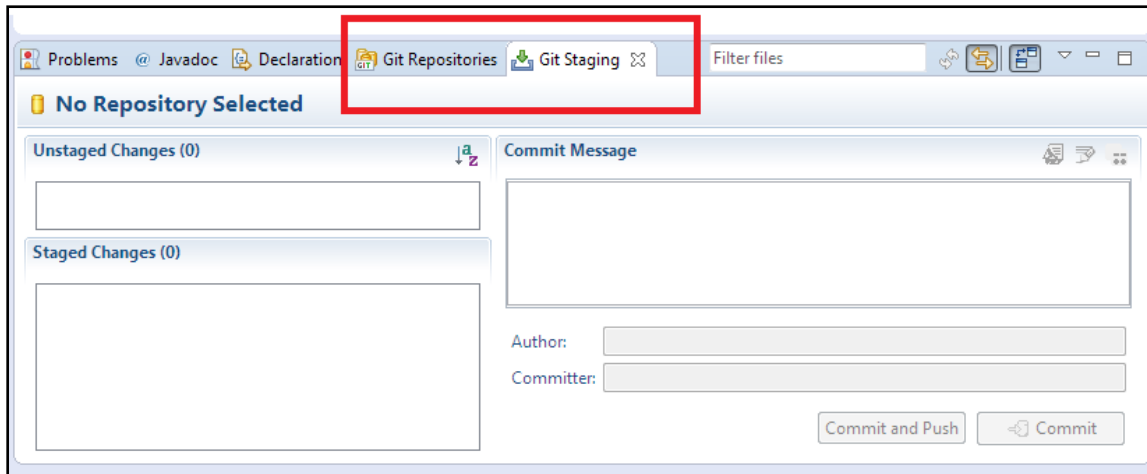
- Create A new Java Project (**project name and repository name Should be same**)
- Add Git Views



Search with Git, select Git Repository, Staging

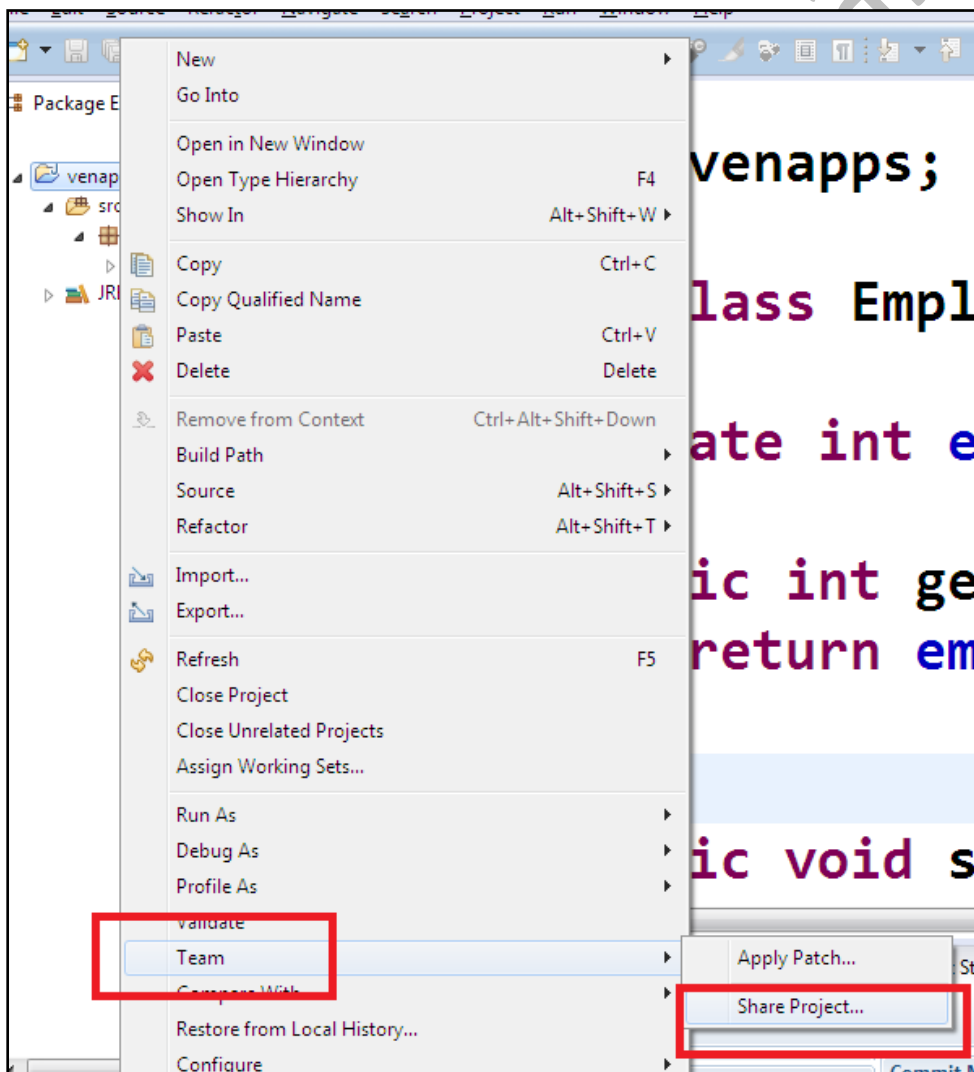


Observer options like

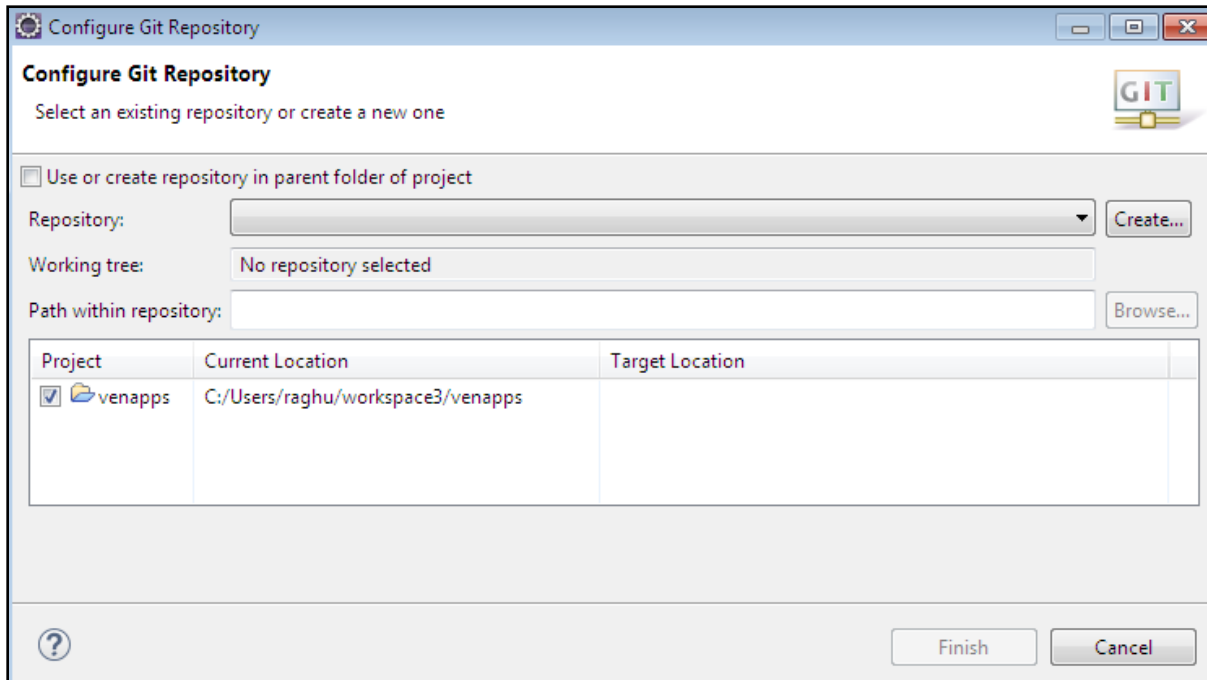


After writing of some code ex: Employee.java

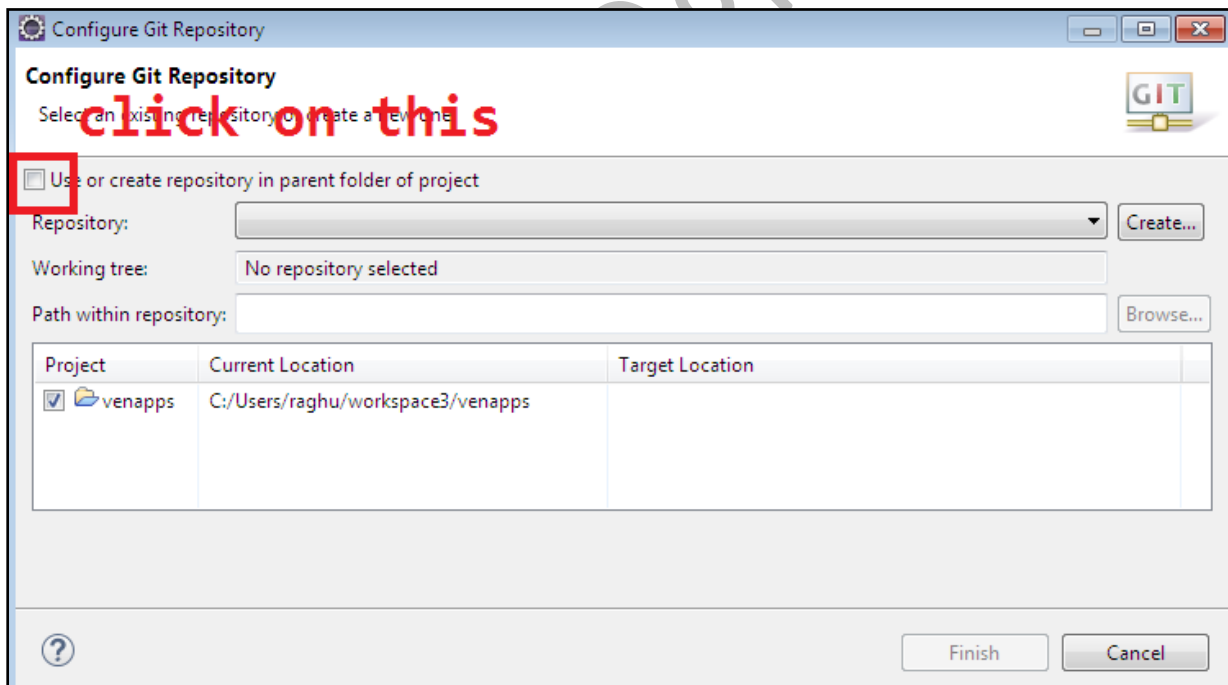
>right click on project > tem > share project



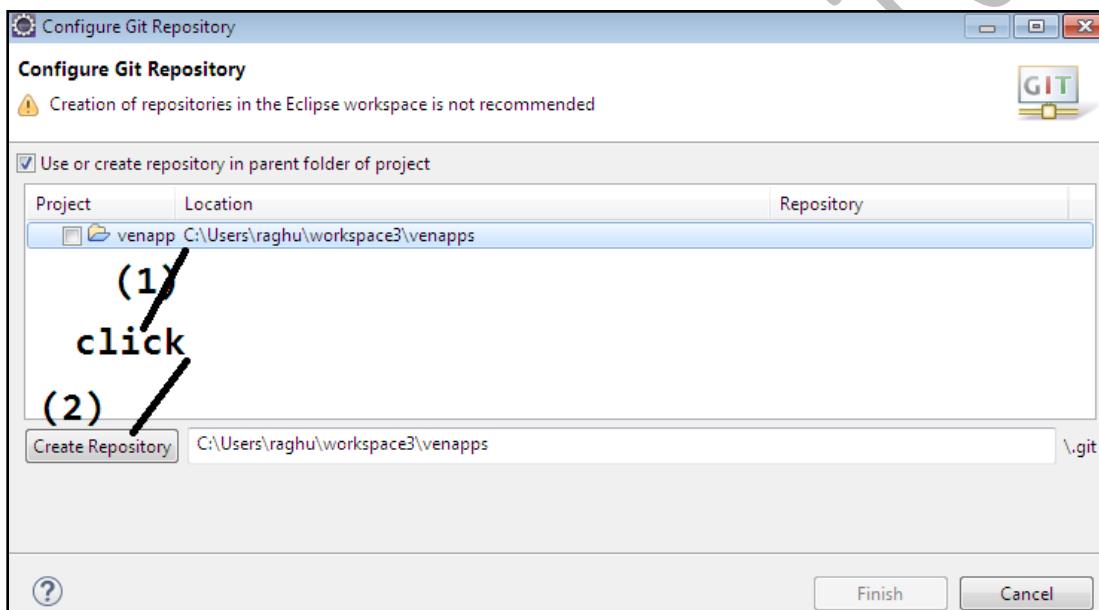
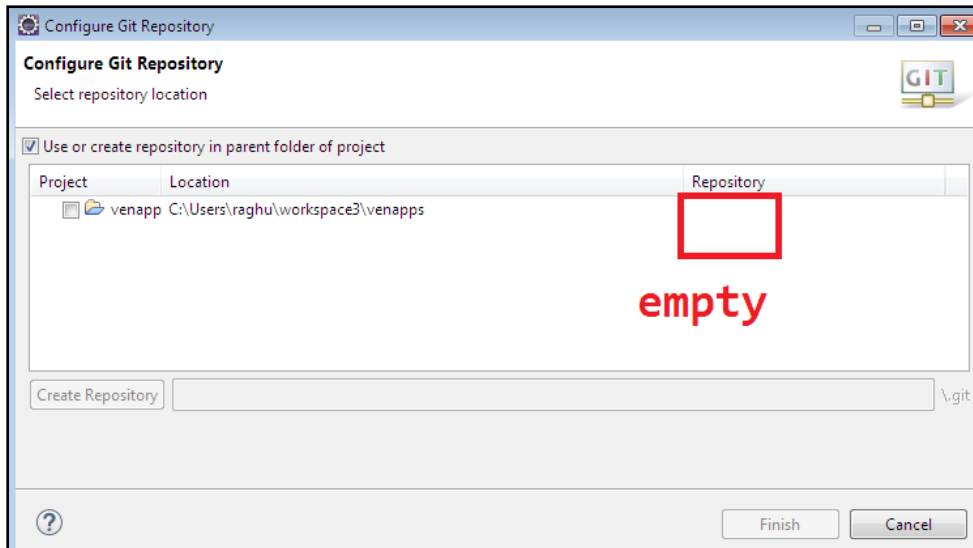
looks as below:



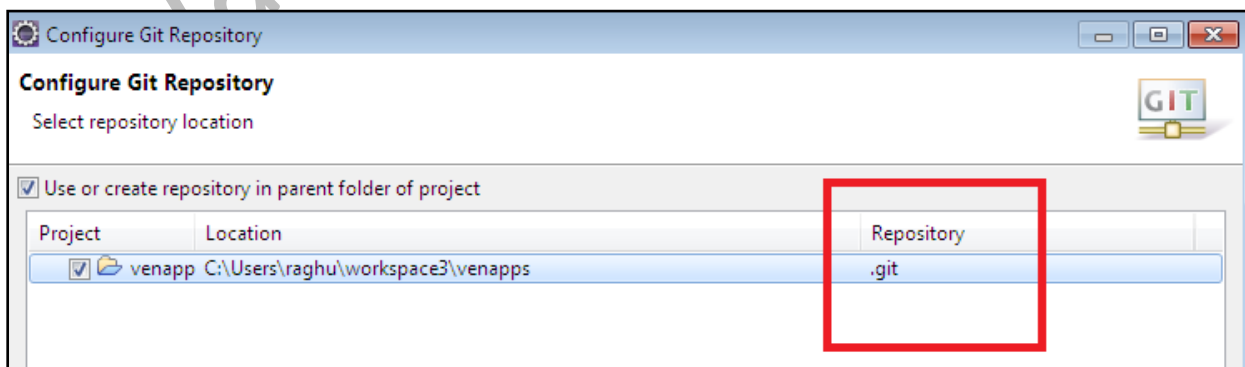
Click on check box:



looks as below:

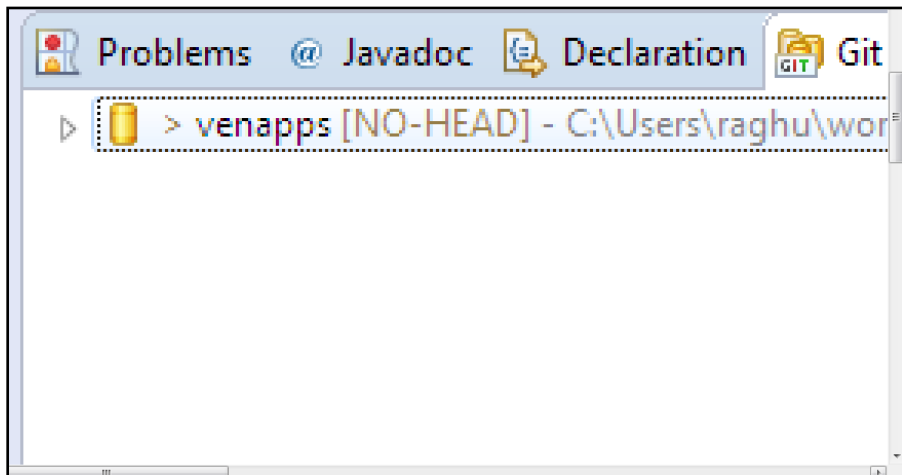


then observe repository name:

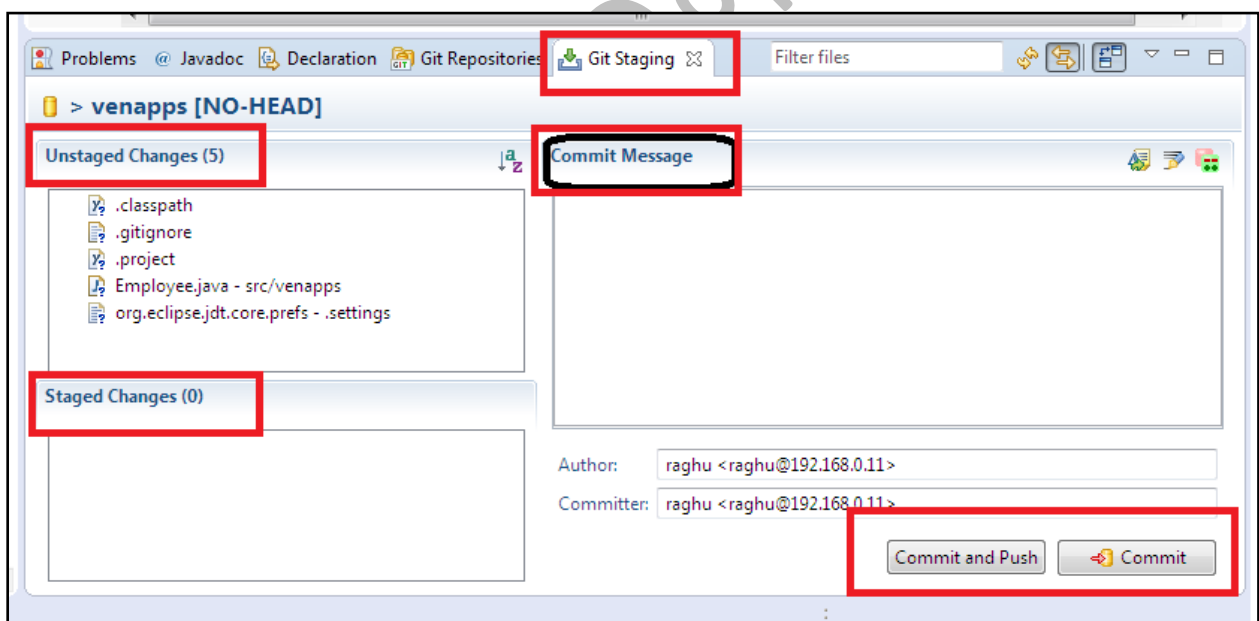


click on finish.

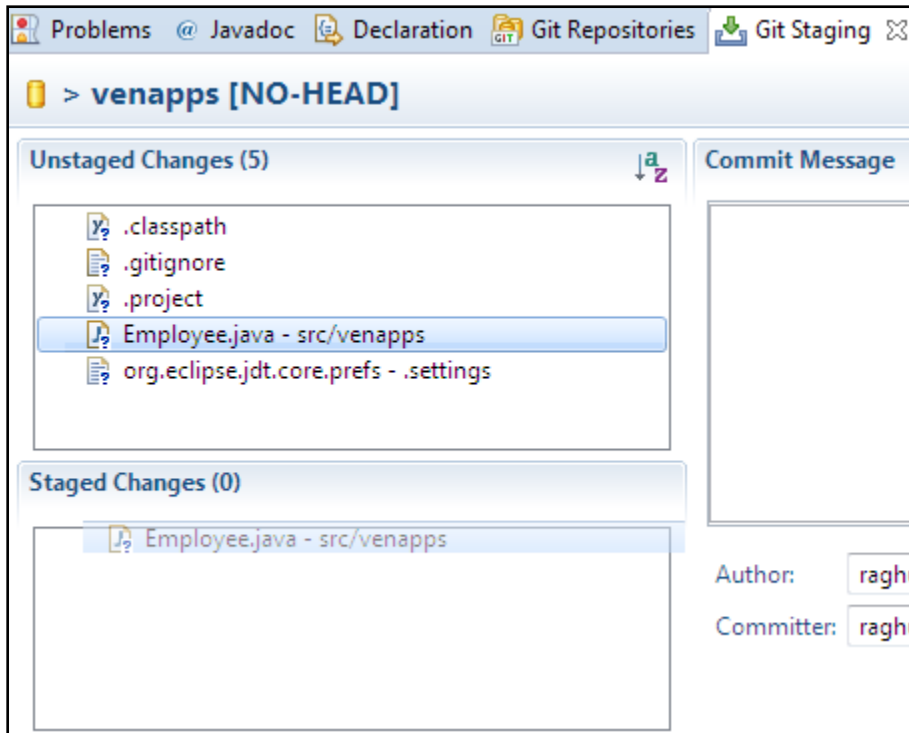
By this we created local and remote repositories.
 Now link local and remote, using first commit and push.
 Come to git repositories



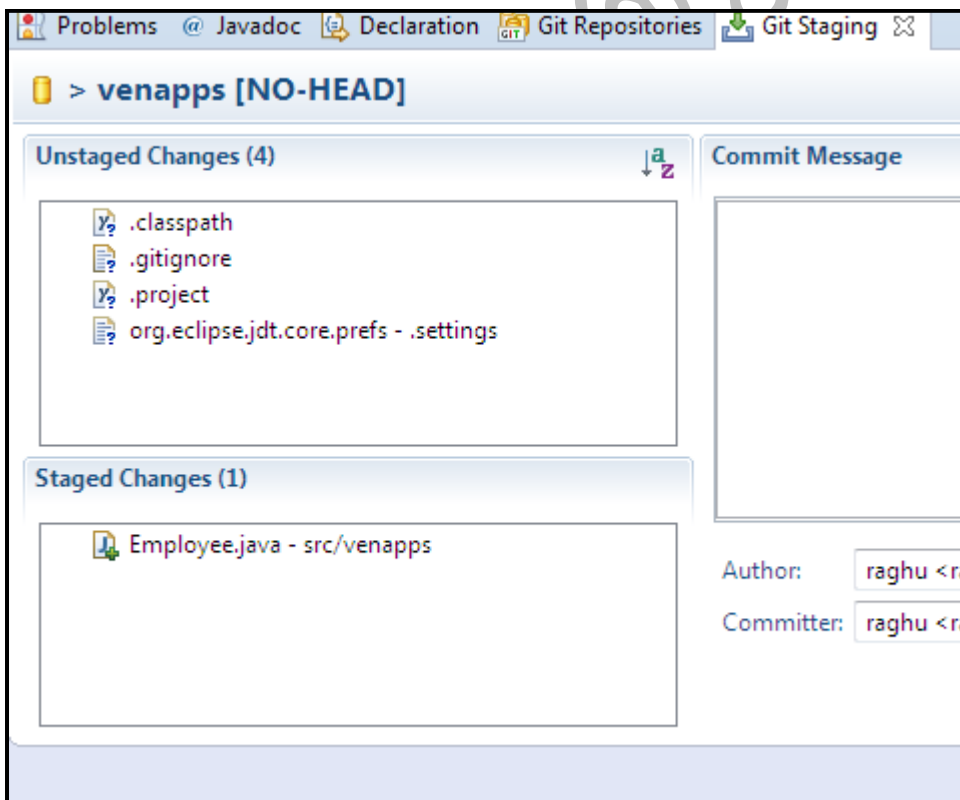
In staging are:



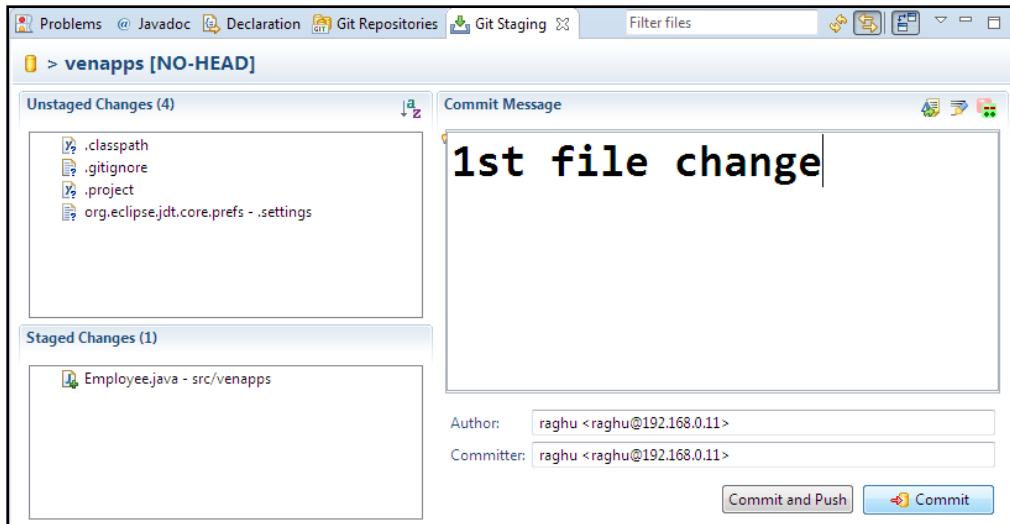
- 1) Unstaged changes : files which are not ready to send to local
 - 2) staged changes: file which needs to move to local/remote
 - 3) commit message: what is the purpose specify here as message for sending of this code.
 - 4) click commit/ commit & push to send the code.
-
- *) Click on a file in unstage drag and drop into stage.
 Click and Drag:



After drop

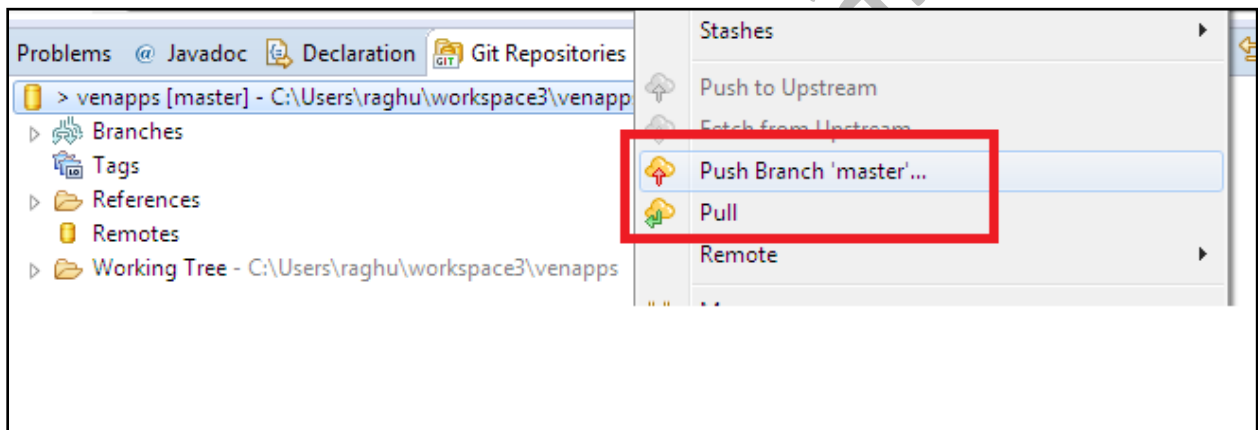


Enter commit message



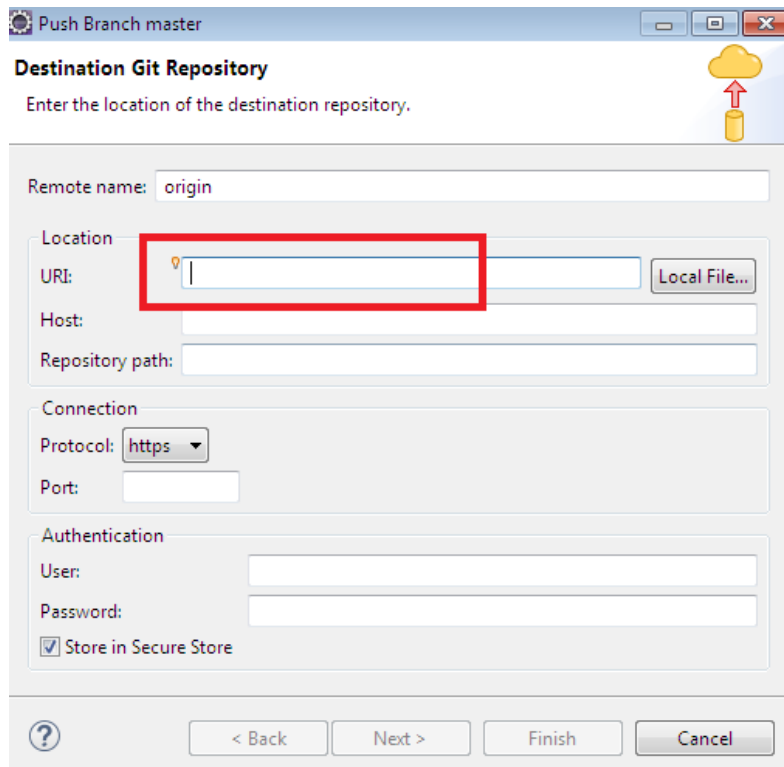
And click on commit.

In GitRepository , right click on projectName>choose Push Branch master



After Push option one screen looks below, there enter URI
(Repository link)

Ex: <https://github.com/javabyraghu/venapps.git>



Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name: origin

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

☒ Store in Secure Store

< Back Next > Finish Cancel



Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name: origin

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

☒ Store in Secure Store

(1)

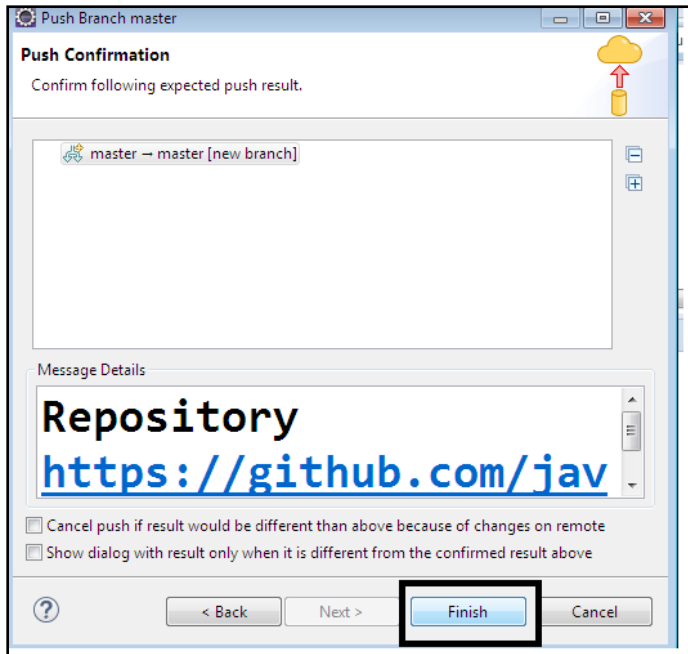
(2)

(3)

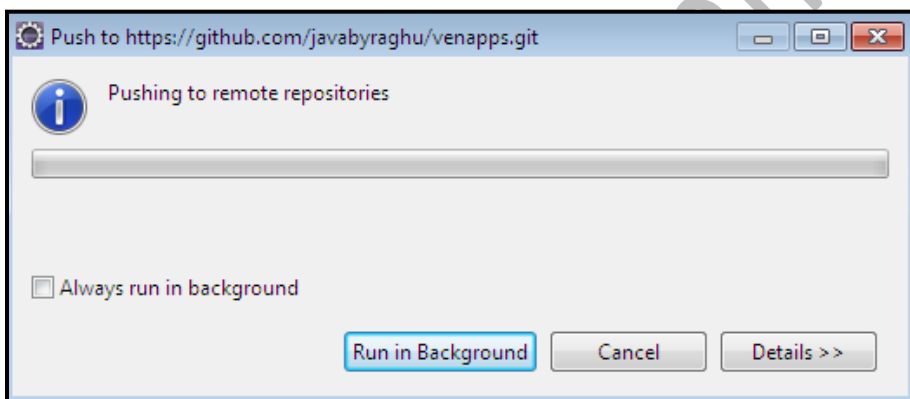
enter username and pwd

< Back Next > Finish Cancel

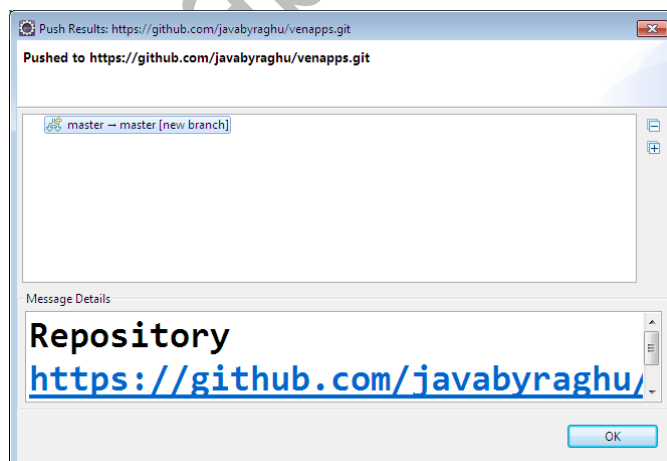
Click on next>next>finish



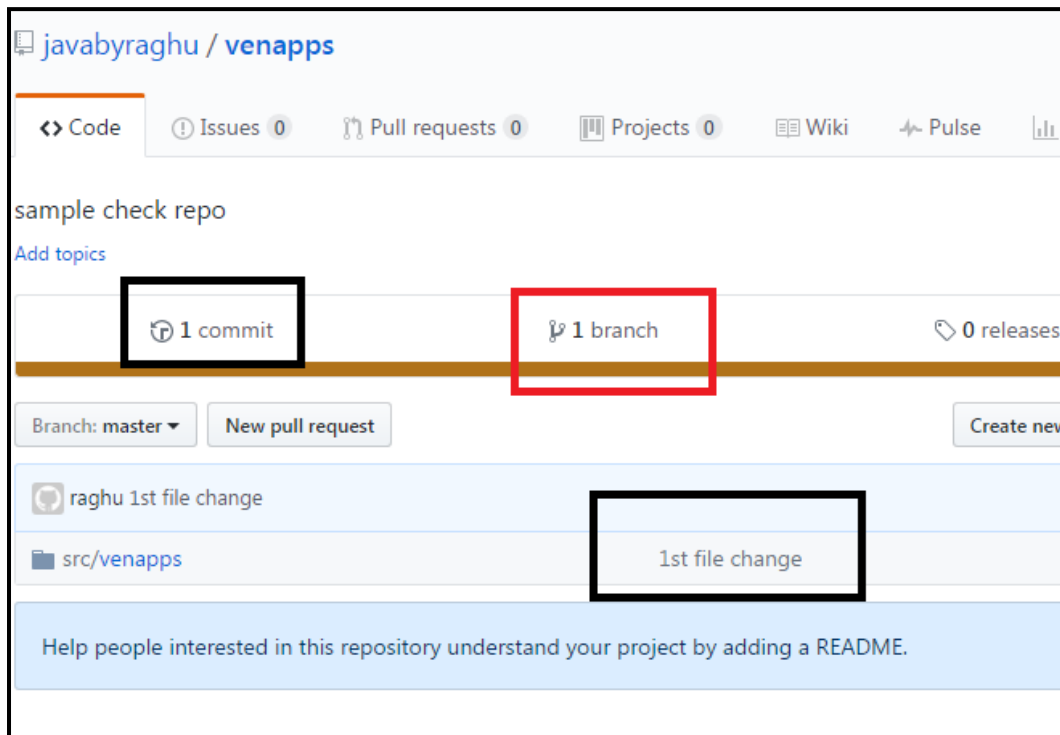
Shows as below:



On success:



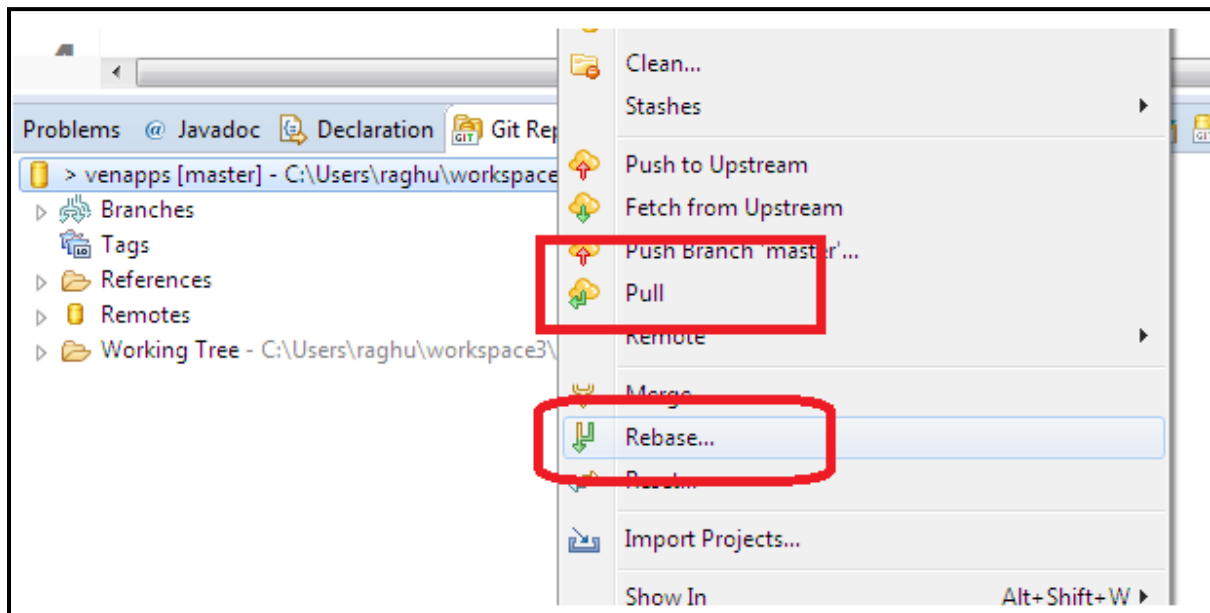
Come back to browser and refresh:



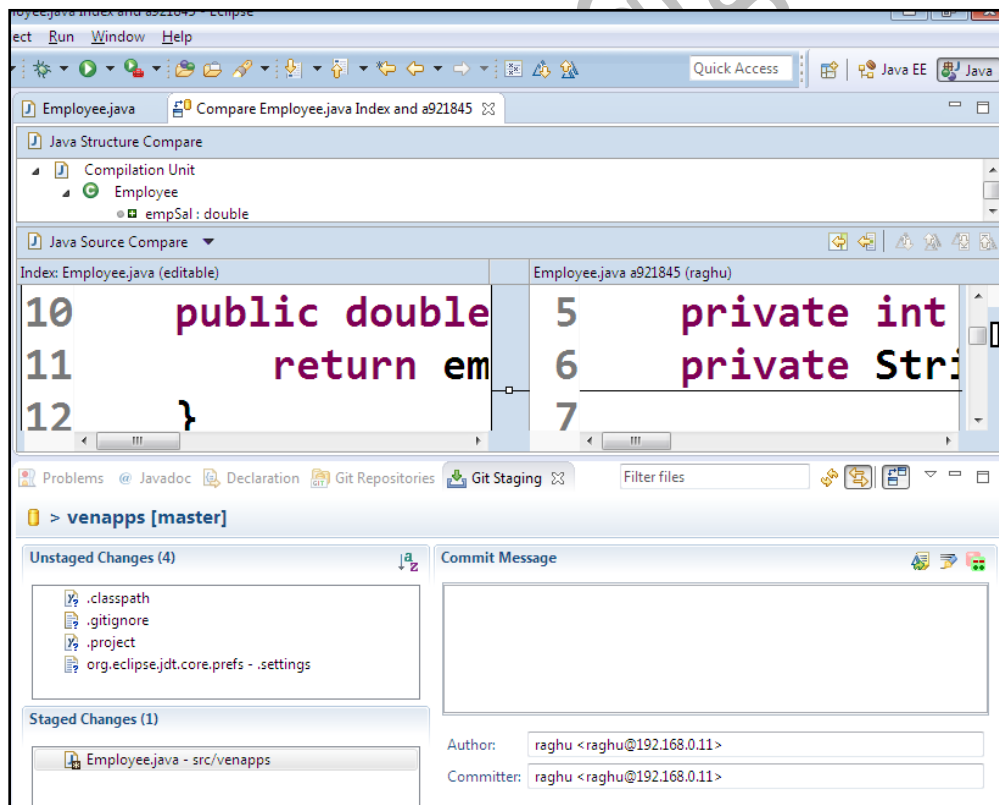
Click on src/venapp>Employee.java



Right click on project: pull and then rebase for other update:

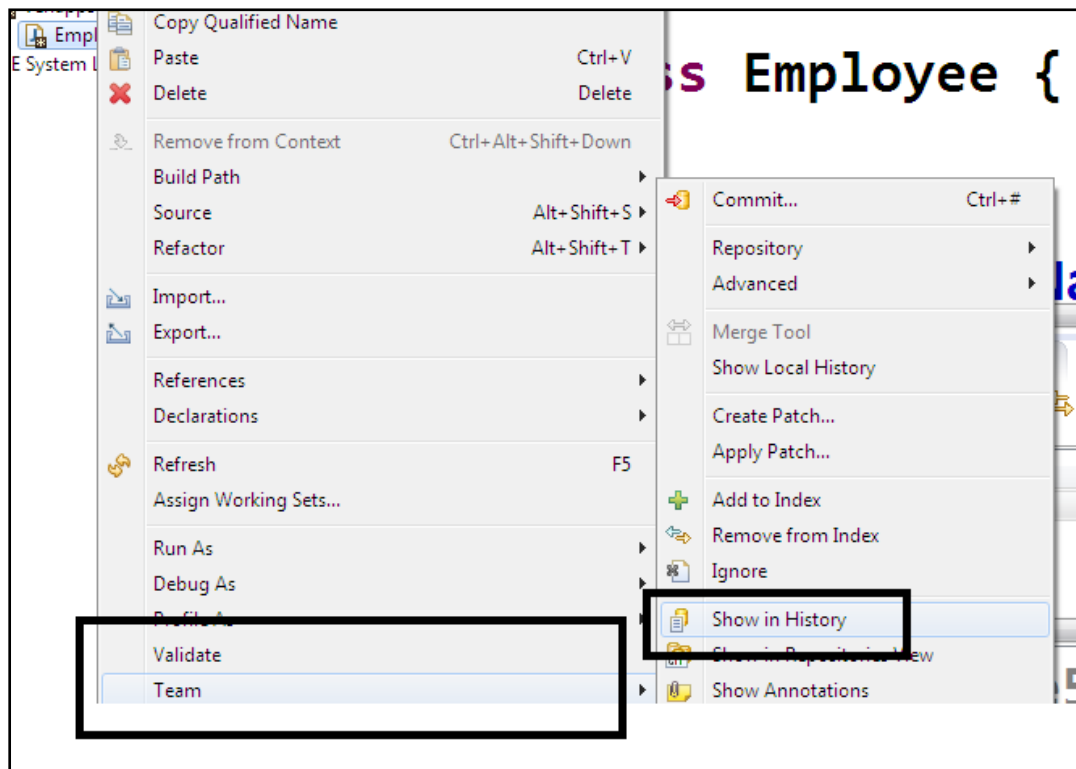


After Adding file to staging ,double click on that to compare with Old one (previous one)looks as :

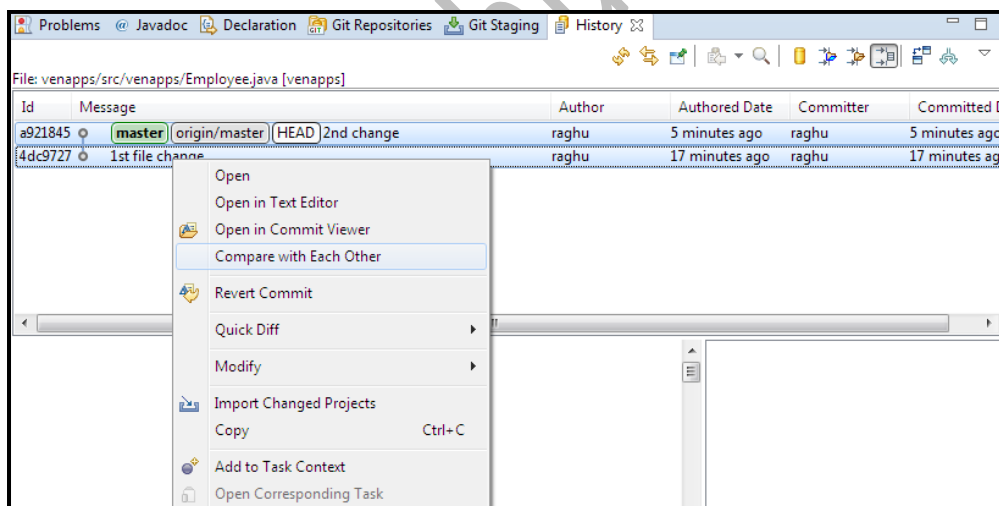


To see history:

Right click on File (ex: Employee.java)> Team>Show in History



View history:



**** GIT Merge Conflicts ****

If any file code is modified(added/removed) then file version will be changed.

ex: Employee.java = Version#1

1. class Employee

2. {

3.

4. }

Employee.java = Version#2

1. class Employee

2. {

3.+ int empId;

4.+ String empName; - }

5.+ }

V#2 = V#1 [+3,-1].

Git manages version management automatically Here programmer do not need to remember any version number.

** On git pull & rebase, local repository and remote repository versions will become equal.

1. Repositories
2. right click on project
3. "pull" option
4. again right click
5. "rebase" option.

Merge conflict example:

Step#1: Dev#1 has done git pull & rebase

Step#2: Dev#2 has done git pull & rebase

Step#3: Dev#1 modified Employee.java

(that is changed to Version#2) and
did add/commit/push.

Now git has Employee.java Version#2

Step#4: Dev#2 Still working with old code
of Employee.java (Version#1).

File modified and did add/commit/
push, then git shows
Git - Version Conflict: Dev#2
Employee.java(version#1) not
matched with Git Repo Employee.java
(Version#2). Please update before
commit/push.

To Resolve merge conflict steps are.

- project -> right click -> reset
- project -> right click -> pull & rebase
- project -> right click -> merge

Now code will be updated to V#2,
write your code, then >>> add/commit/push.

face book group Id: <https://www.facebook.com/groups/thejavatemple/>
email :
javabyraghu@gmail.com