Document Info

Purpose:

To configure GIT and Basic GIT Operation

Configure GIT Desktop and basic usages of GIT from Eclipse

Create a new file from and branch with pull request

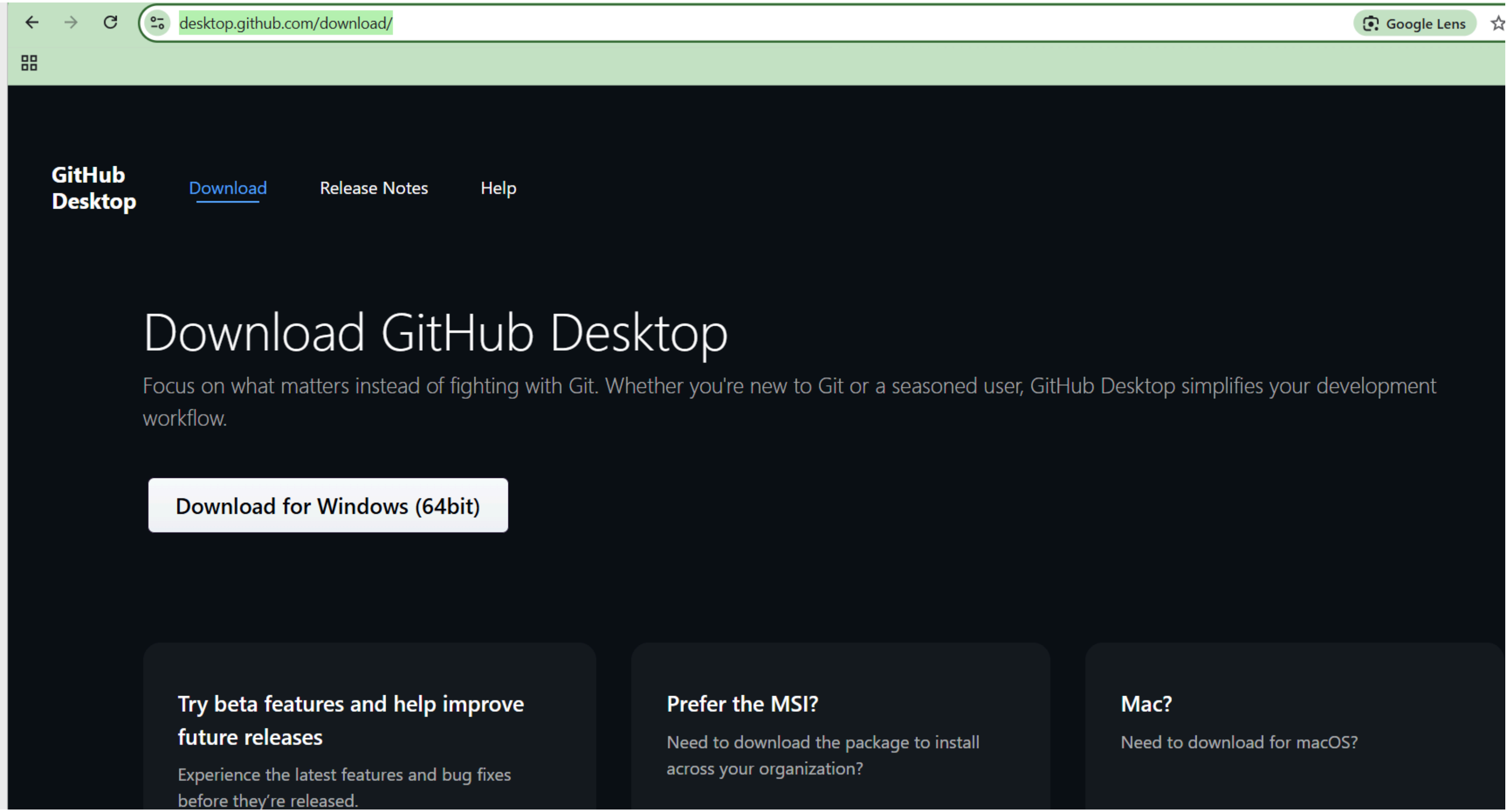
Merge the file with the main branch

Author: Poon

Date: 8 June 2025

Download and configure GitHub Desktop

https://desktop.github.com/download/



Download GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

Try beta features and help improve future releases

Experience the latest features and bug fixes before they're released.

Prefer the MSI?

Need to download the package to install across your organization?

Mac?

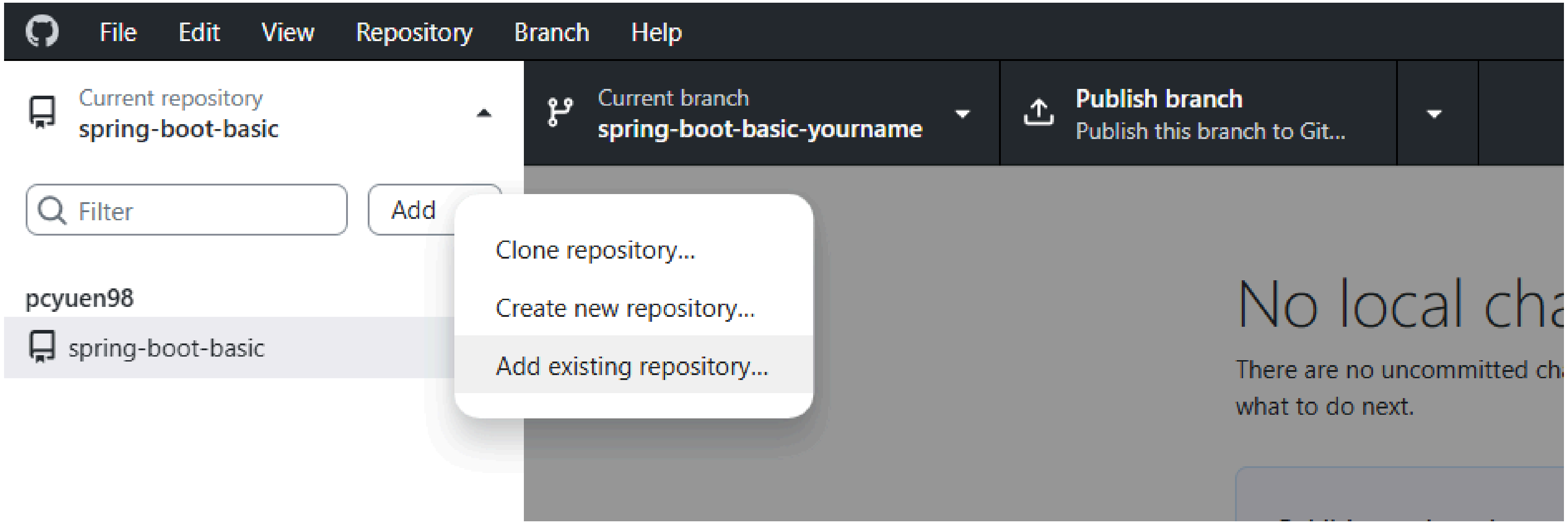
Need to download for macOS?

Clone a project

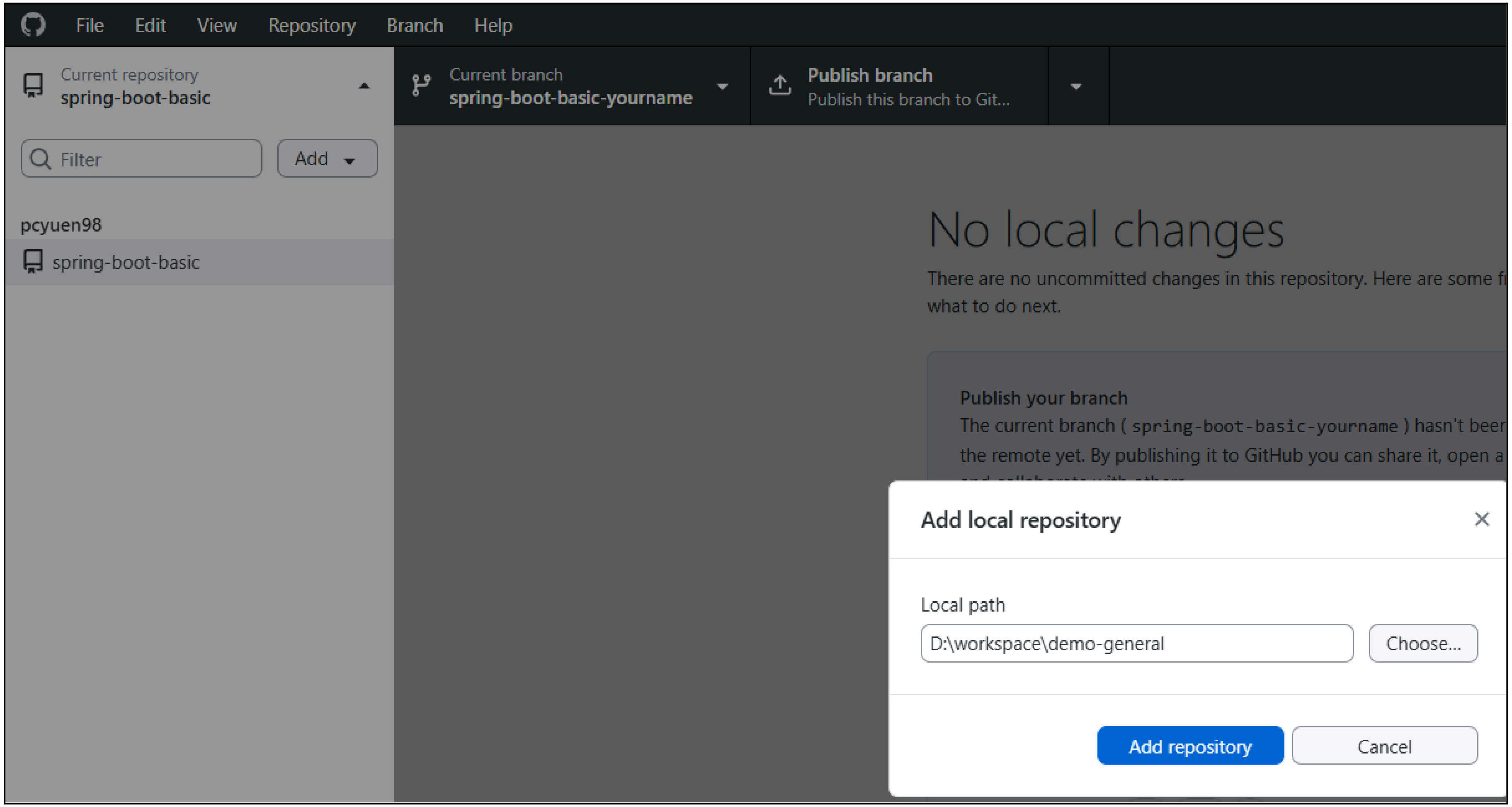
Clone this project as example as below and duplicate your own repo **<Your GitHub URL>/DevOps/Git_Demo_App**
https://github.com/pcyuen98/demo-general/

Maven build is not required as it’s a simple program.

From GitDesktop: Add the “your” cloned project



Select the directory



No local changes

There are no uncommitted changes in this repository. Here are some f
what to do next.

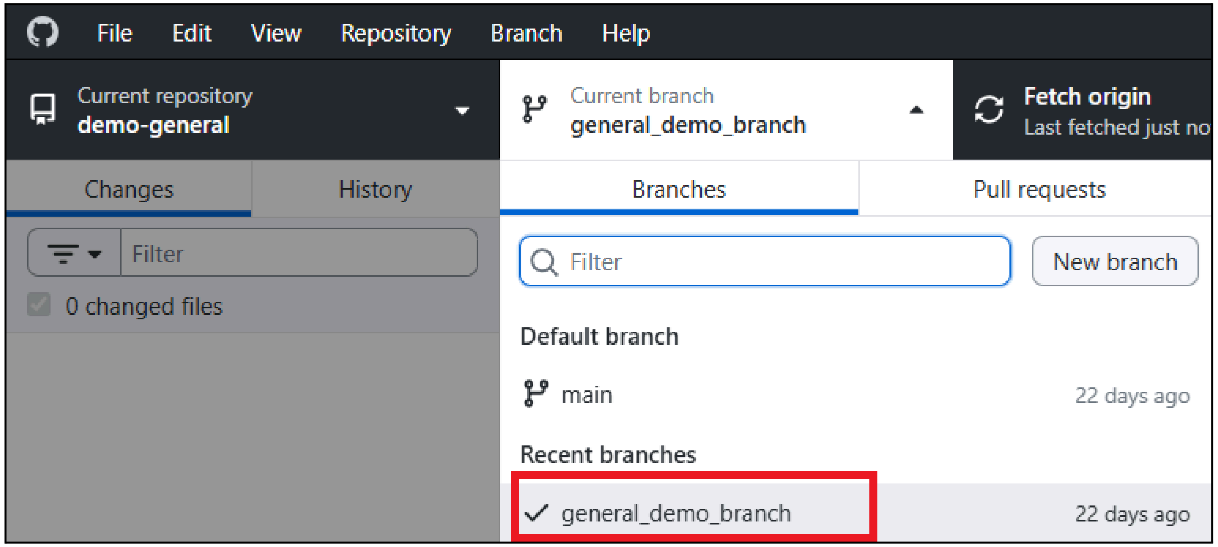
Publish your branch

The current branch (`spring-boot-basic-yourname`) hasn't beer
the remote yet. By publishing it to GitHub you can share it, open a

Add local repository

Local path: D:\workspace\demo-general

Ensure a branch is exist



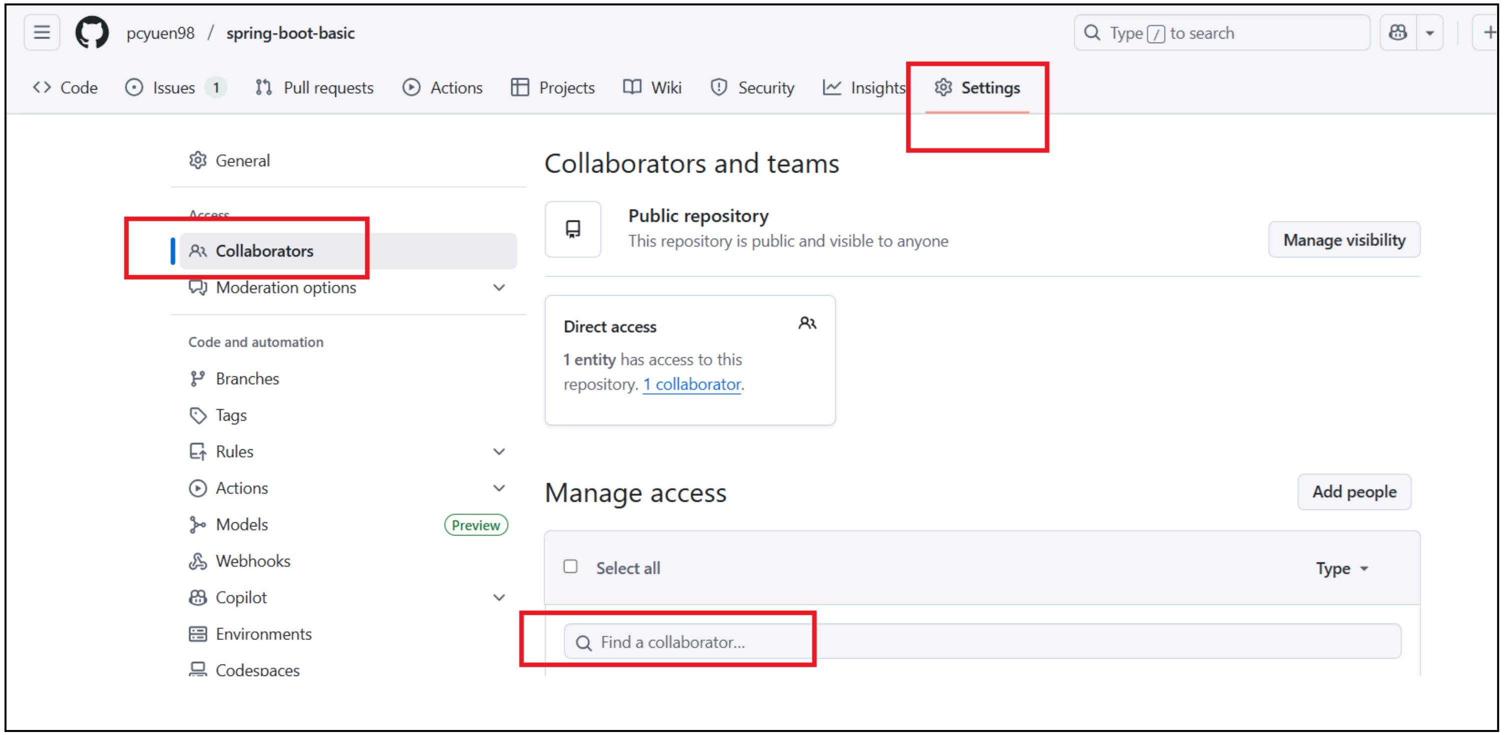
Create a new file push, pull request and merge

Section 1 - GIT Commit and Push - Create a file then push to GitHub

Create a new file from branch and commit

Ensure using your own repo - <GitHub URL>/DevOps/Git_Demo_App

On Git Hub: Add your peer for pull request/approval. Add manage access below as their GitHub username or Git Email



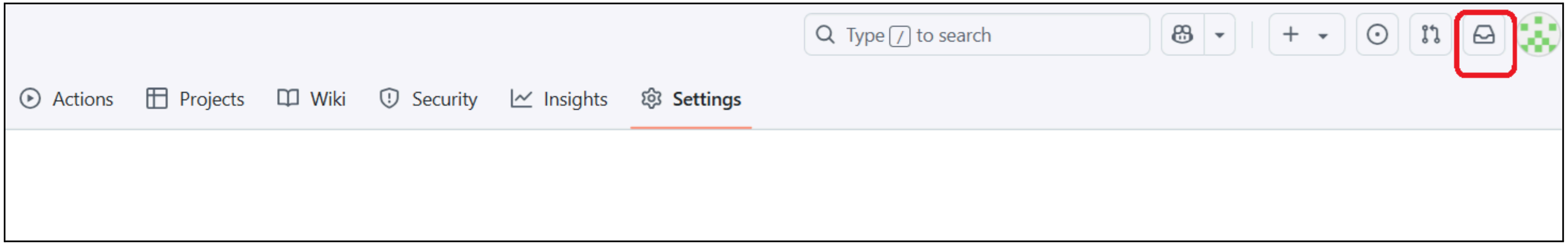
Collaborators and teams

Public repository
This repository is public and visible to anyone

Direct access
1 entity has access to this repository. 1 collaborator.

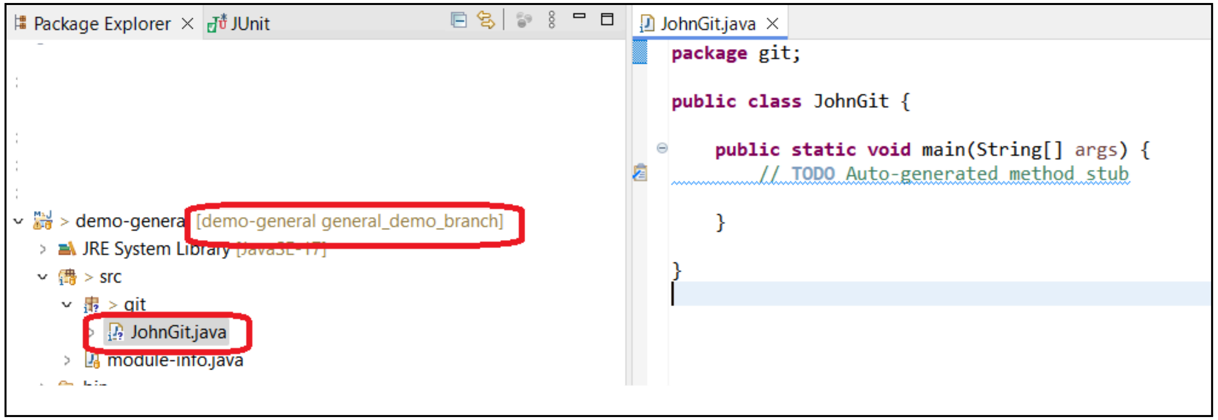
Manage access

On your peer Git Hub:
Your peer is supposed to receive a notification. Click on it and accept the invitation.



On Eclipse: Create and commit your first java file under the package of the git folder.

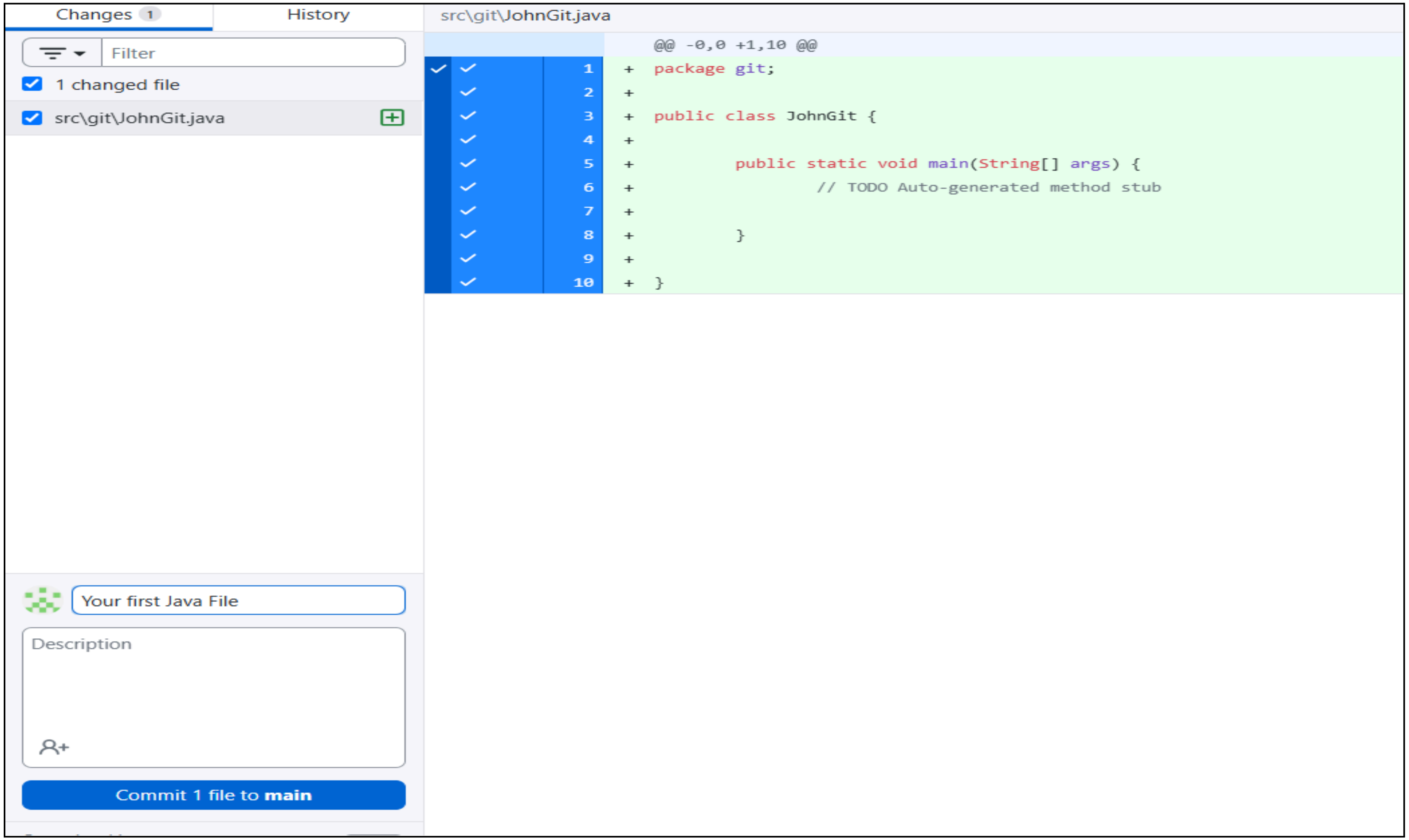
Note: Use your <name> as sample below. Ensure you are working on the “Branch” and not “Main”


Package Explorer
JUnit
demo-genera [demo-general general_demo_branch]
JRE System Library [JavaSE-17]
src
git
JohnGit.java
module-info.java
JohnGit.java

```
package git;

public class JohnGit {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

}
```

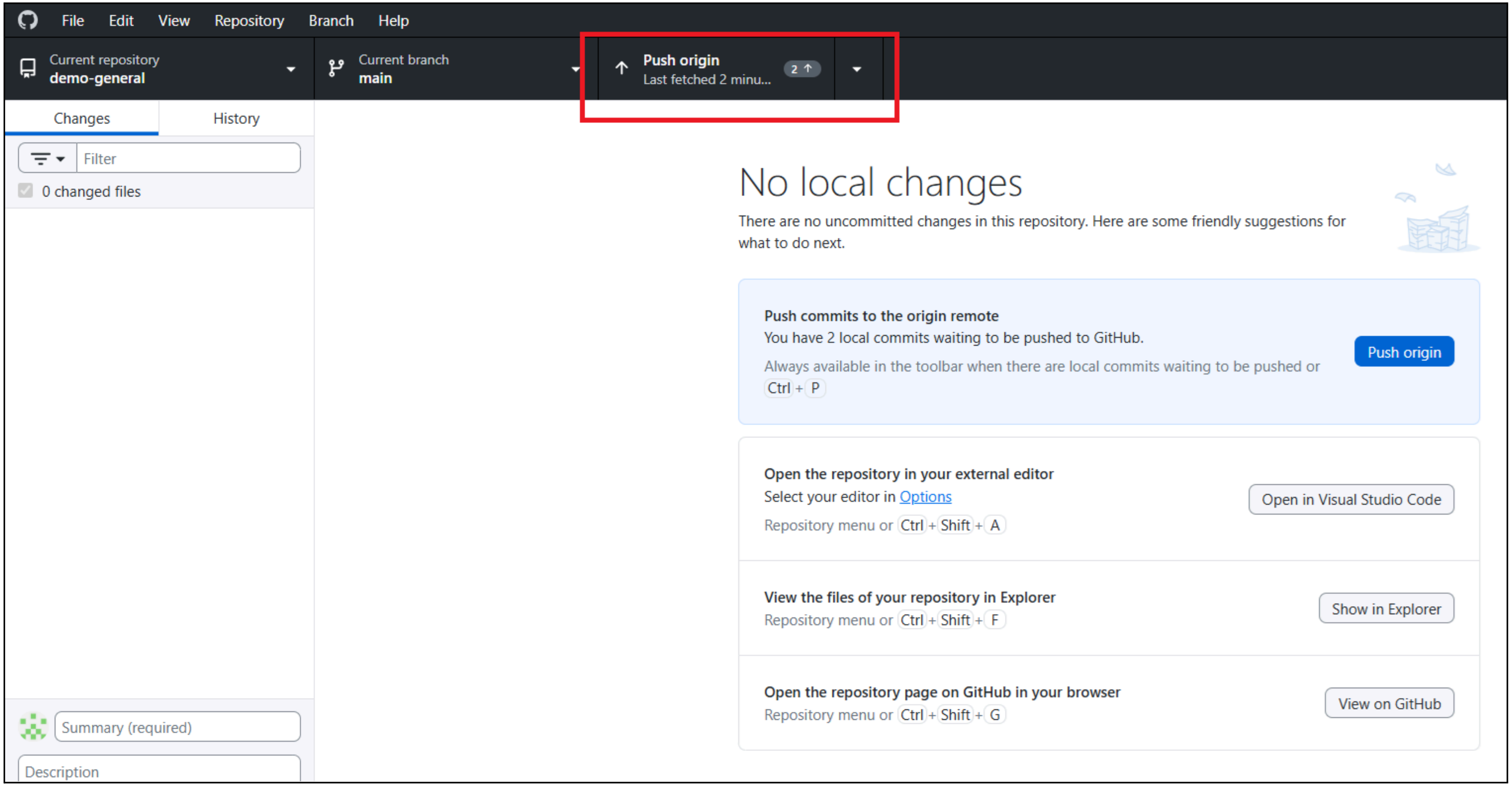
On Git Desktop: Create and commit your first java file under package of git folder

Note: Use your <name> follow by Git as sample below



```
@@ -0,0 +1,10 @@
+ package git;
+
+ public class JohnGit {
+
+         public static void main(String[] args) {
+                 // TODO Auto-generated method stub
+
+         }
+
+ }
```

On Git Desktop: Click push origin



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote
You have 2 local commits waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or Ctrl + P

Push origin

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl + Shift + A

Open in Visual Studio Code

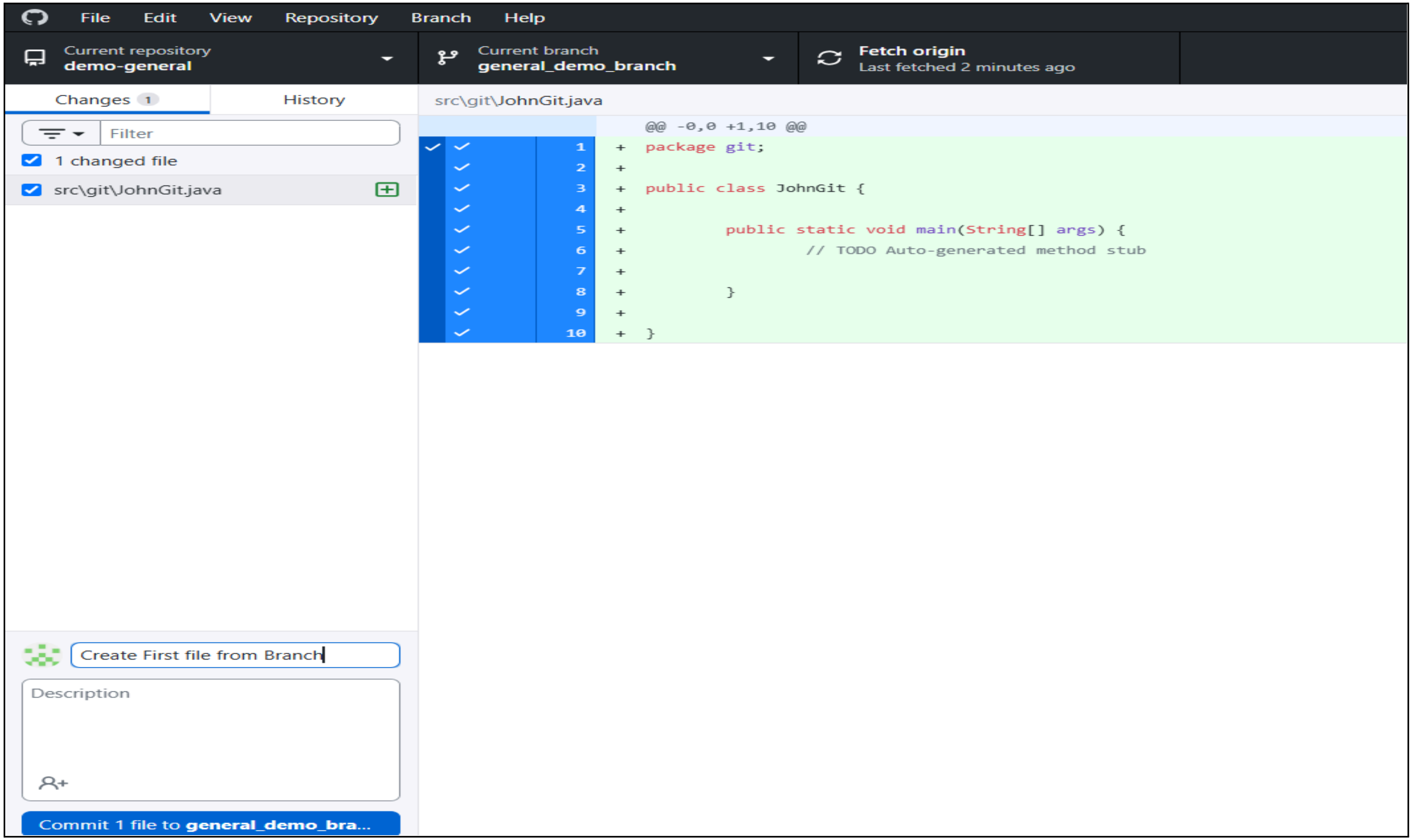
View the files of your repository in Explorer
Repository menu or Ctrl + Shift + F

Show in Explorer

Open the repository page on GitHub in your browser
Repository menu or Ctrl + Shift + G

View on GitHub

From GIT Desktop: Click Commit first file creation from Branch



Current repository: demo-general

Current branch: general_demo_branch

☑ 1 changed file

☑ src\git\JohnGit.java

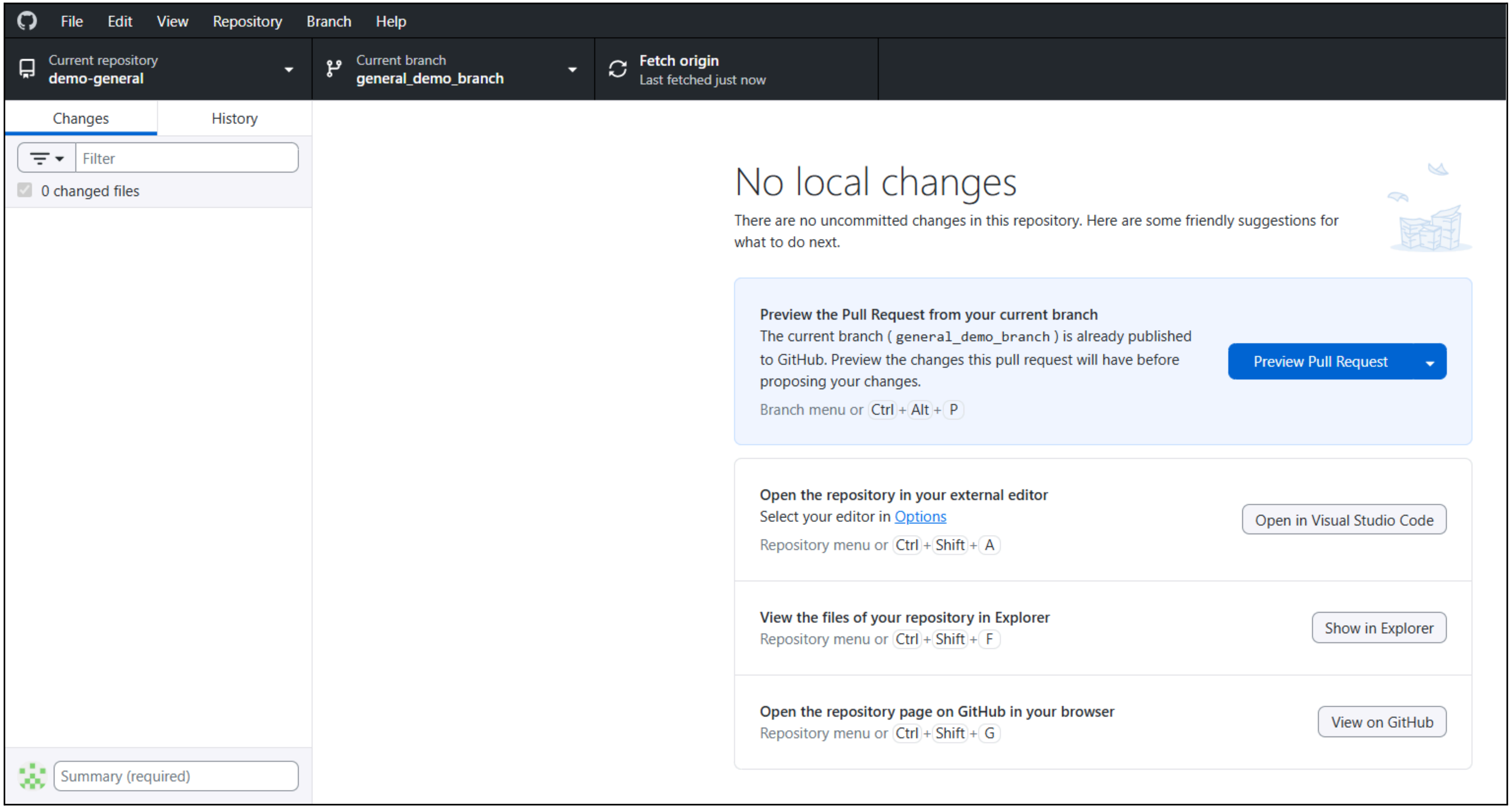
src\git\JohnGit.java

```
@@ -0,0 +1,10 @@
+ package git;
+
+ public class JohnGit {
+
+         public static void main(String[] args) {
+                 // TODO Auto-generated method stub
+
+         }
+
+ }
```

Create First file from Branch

Section 2 - Pull Request - Get Approval

On GIT Desktop: Click Preview Pull Request



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

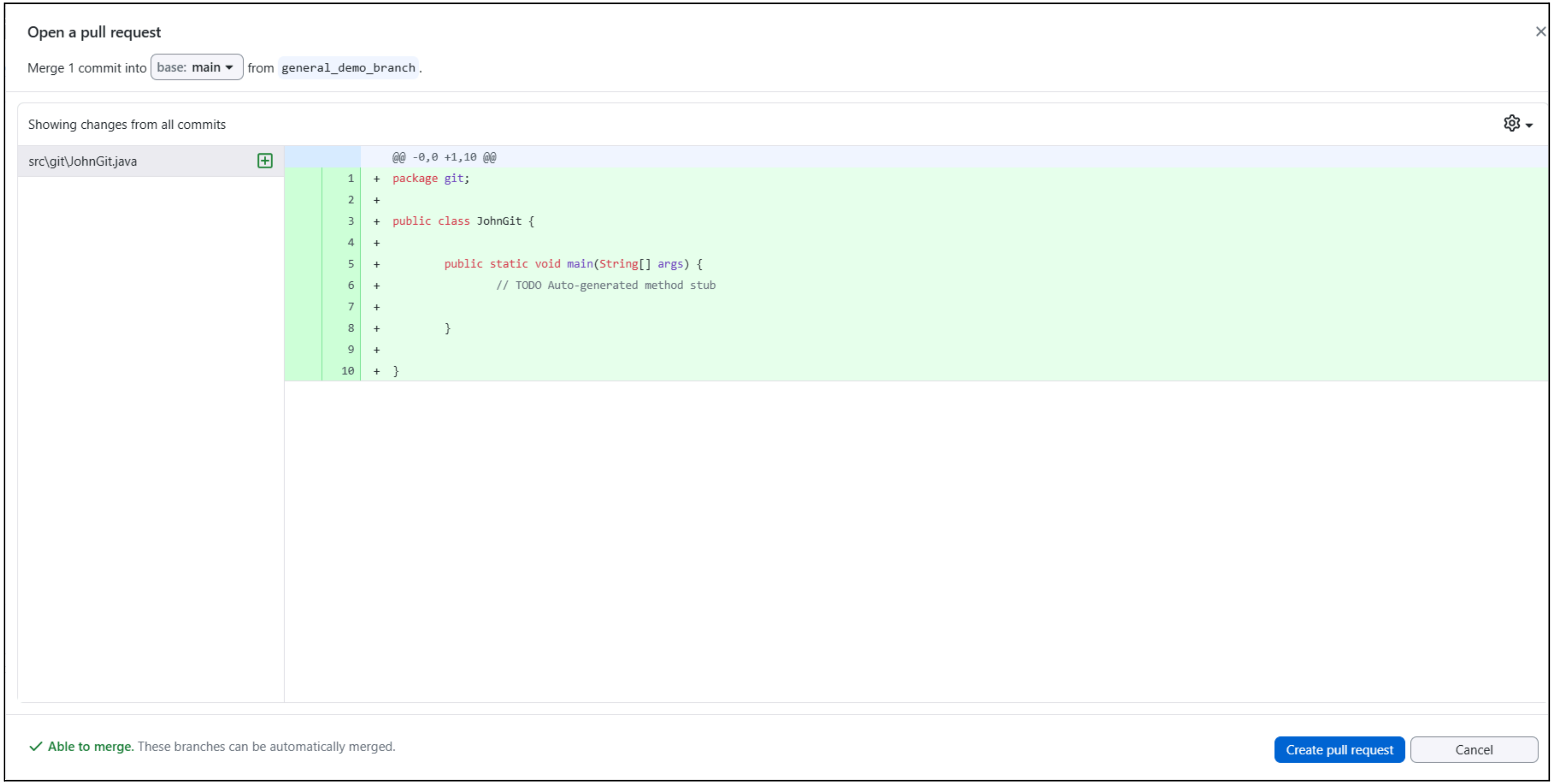
Preview the Pull Request from your current branch
The current branch (`general_demo_branch`) is already published to GitHub. Preview the changes this pull request will have before proposing your changes.
Branch menu or Ctrl + Alt + P

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl + Shift + A

View the files of your repository in Explorer
Repository menu or Ctrl + Shift + F

Open the repository page on GitHub in your browser
Repository menu or Ctrl + Shift + G

On GIT Desktop: Click Create Pull request



Open a pull request

Merge 1 commit into base: main from `general_demo_branch`.

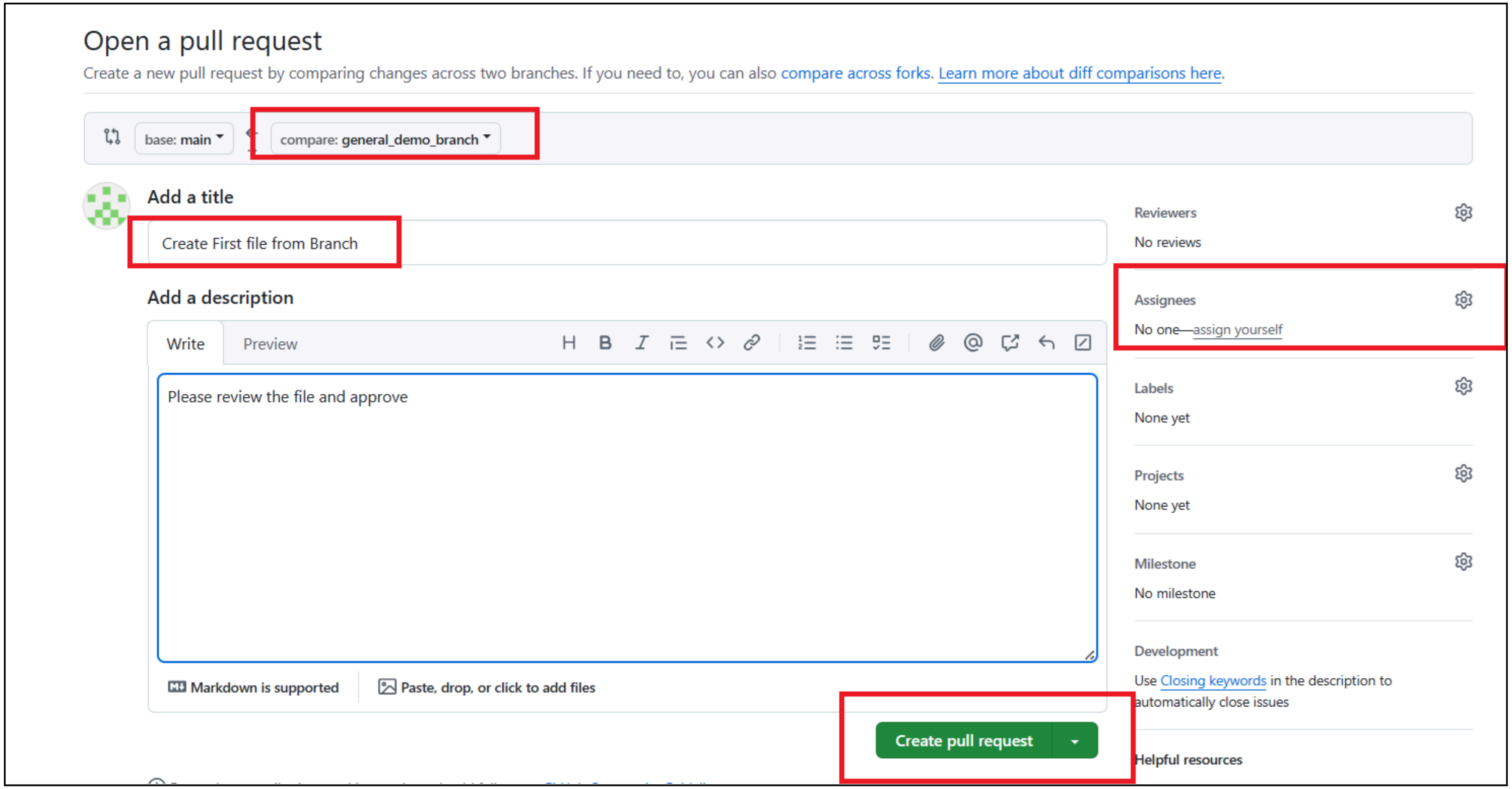
Showing changes from all commits

src\git\JohnGit.java

```
@@ -0,0 +1,10 @@
+ package git;
+
+ public class JohnGit {
+
+         public static void main(String[] args) {
+                 // TODO Auto-generated method stub
+
+         }
+
+ }
```

✓ Able to merge. These branches can be automatically merged.

From GITHUB:Assign the pull request to your peer for approval



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff comparisons here.

base: main ▾ compare: general_demo_branch ▾

Add a title

Create First file from Branch

Add a description

Write Preview

Please review the file and approve

Markdown is supported Paste, drop, or click to add files

Create pull request ▾

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

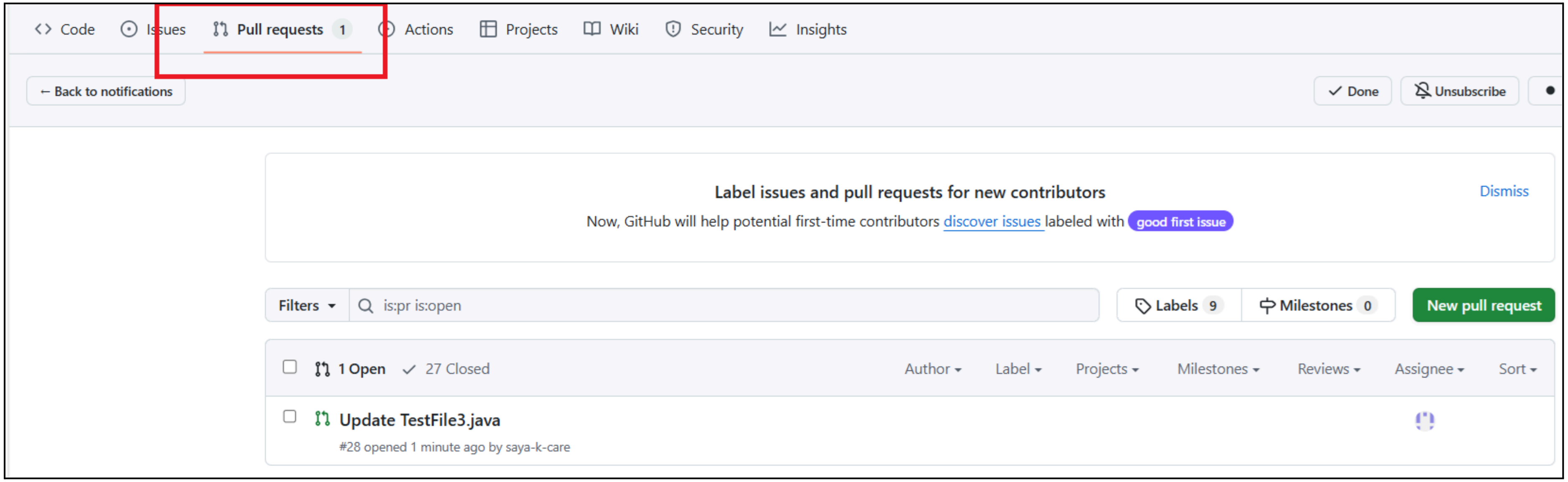
Projects
None yet

Milestone
No milestone

Development
Use Closing keywords in the description to automatically close issues

Helpful resources

From GITHUB:Your assignee should receive an notification about pull request



Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

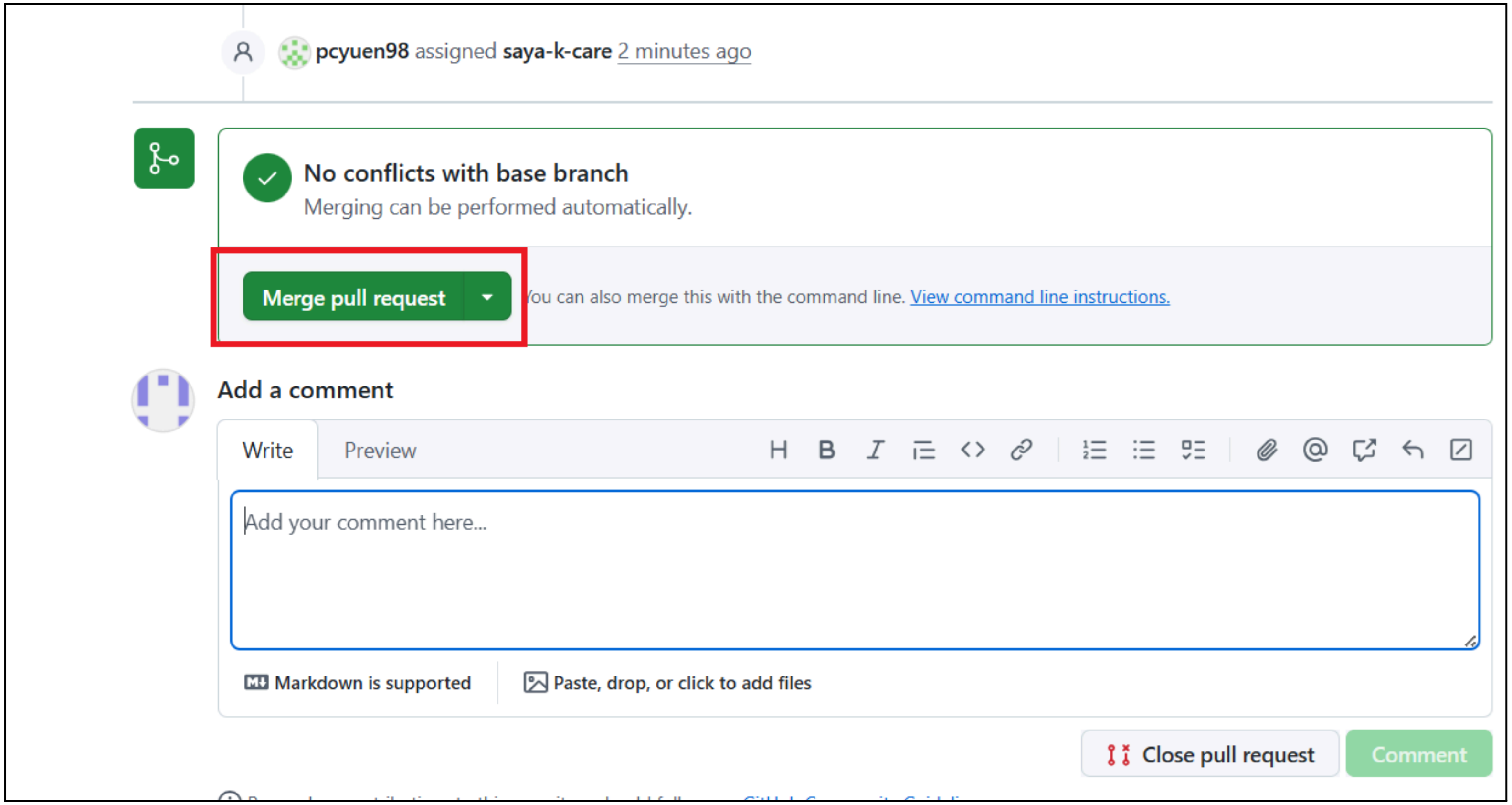
is:pr is:open

□ 1 Open ✓ 27 Closed

□ Update TestFile3.java

#28 opened 1 minute ago by saya-k-care

From GITHUB: Click Merge Pull request then close the pull request for approval or add a comment then click close pull request



pcyuen98 assigned saya-k-care 2 minutes ago

No conflicts with base branch
Merging can be performed automatically.

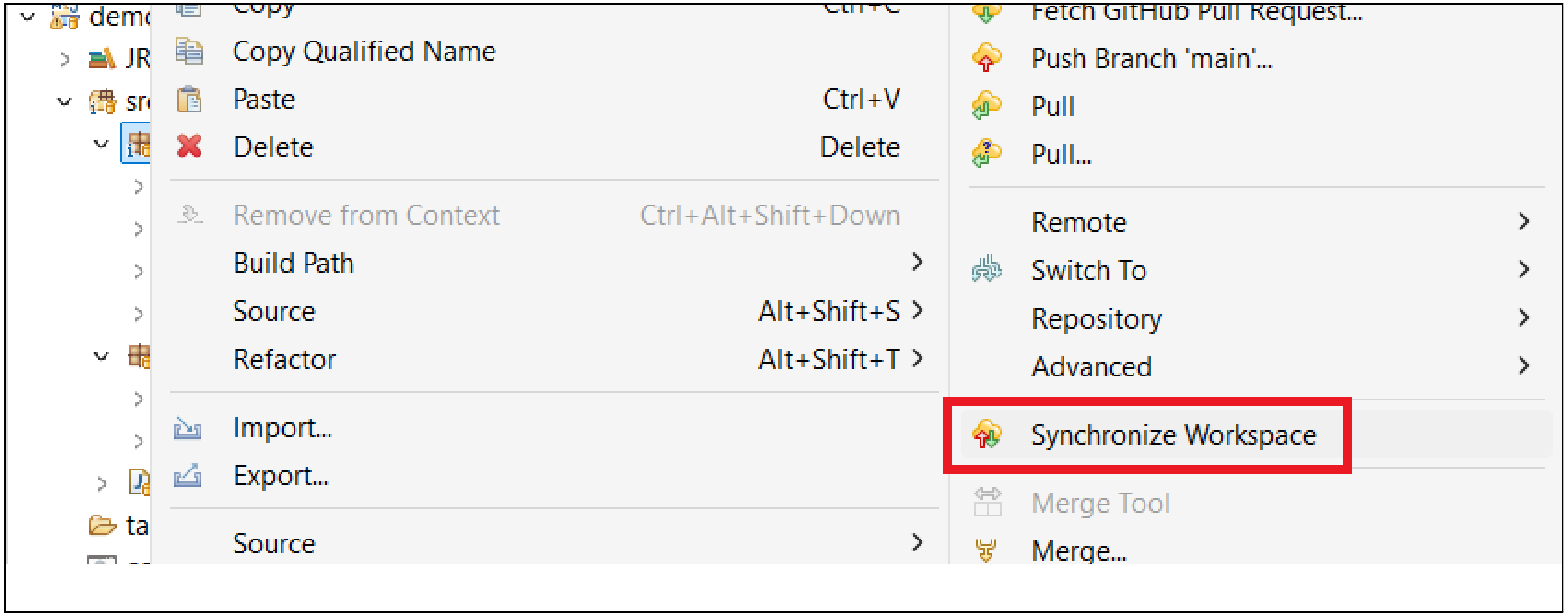
You can also merge this with the command line. View command line instructions.

Add a comment

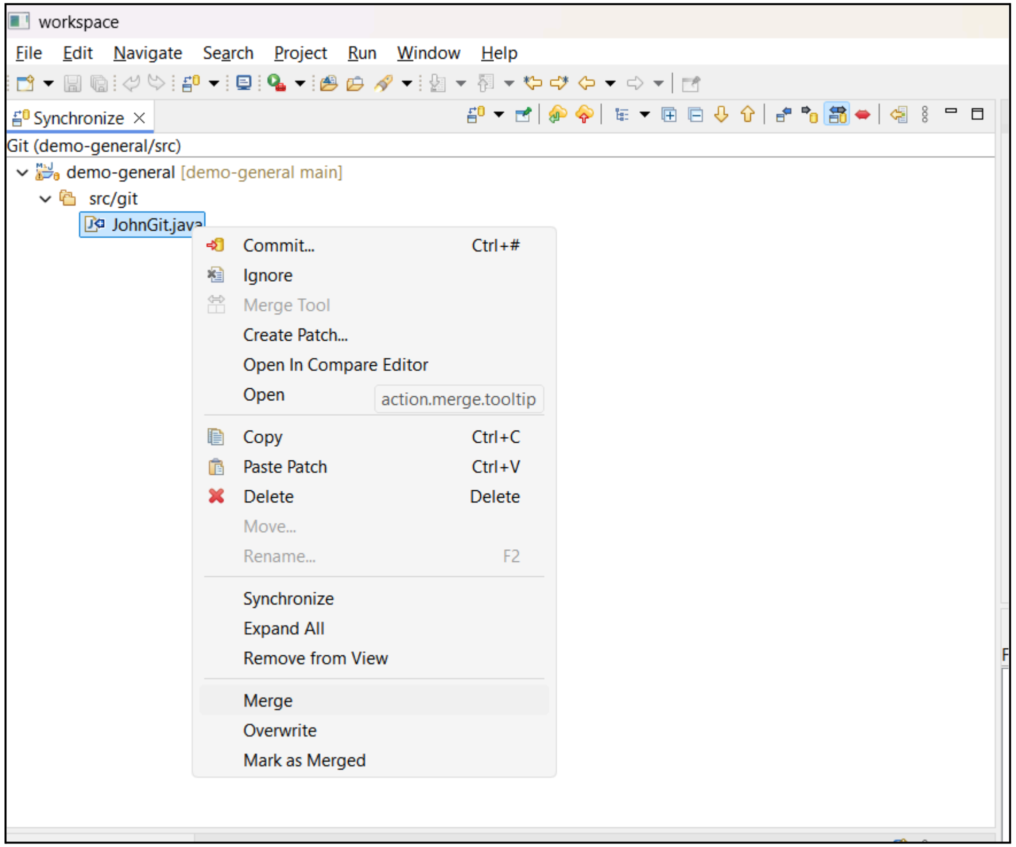
Section 3 - GIT Merge - Check and Merge the approved File

From Eclipse:
Switch to main branch and check for changes

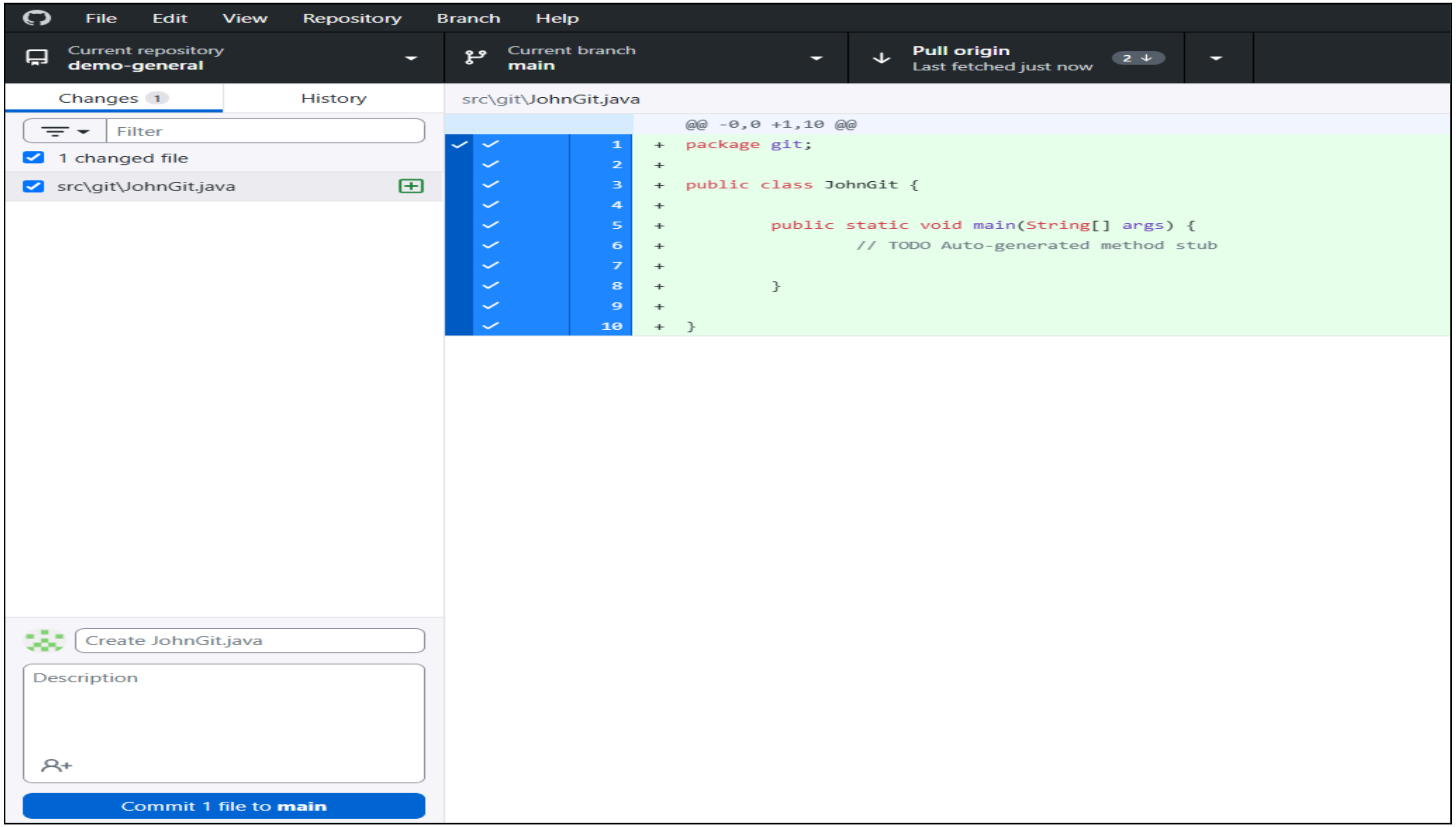
Click on project properties → team → synchonise workspace
Click on team synchronisation and check the changes



Click merge


workspace
File Edit Navigate Search Project Run Window Help
Synchronize
Git (demo-general/src)
demo-general [demo-general main]
src/git
JohnGit.java
Commit... Ctrl+#
Ignore
Merge Tool
Create Patch...
Open In Compare Editor
Open
action.merge.tooltip
Copy Ctrl+C
Paste Patch Ctrl+V
Delete Delete
Move...
Rename... F2
Synchronize
Expand All
Remove from View
Merge
Overwrite
Mark as Merged

From GIT Desktop:Click Commit



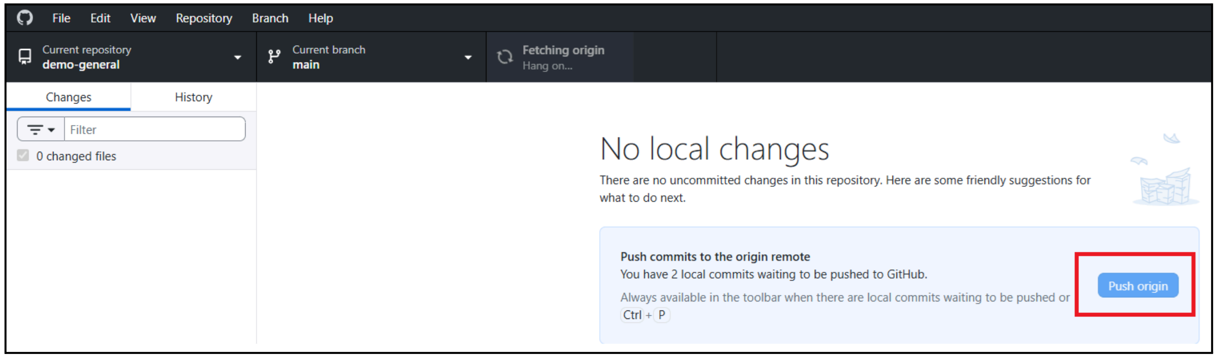
src\git\JohnGit.java

```
@@ -0,0 +1,10 @@
+ package git;
+
+ public class JohnGit {
+
+         public static void main(String[] args) {
+                 // TODO Auto-generated method stub
+
+         }
+
+ }
```

Create JohnGit.java

Description

From GIT Desktop: Click pull and push origin to complete the process



No local changes

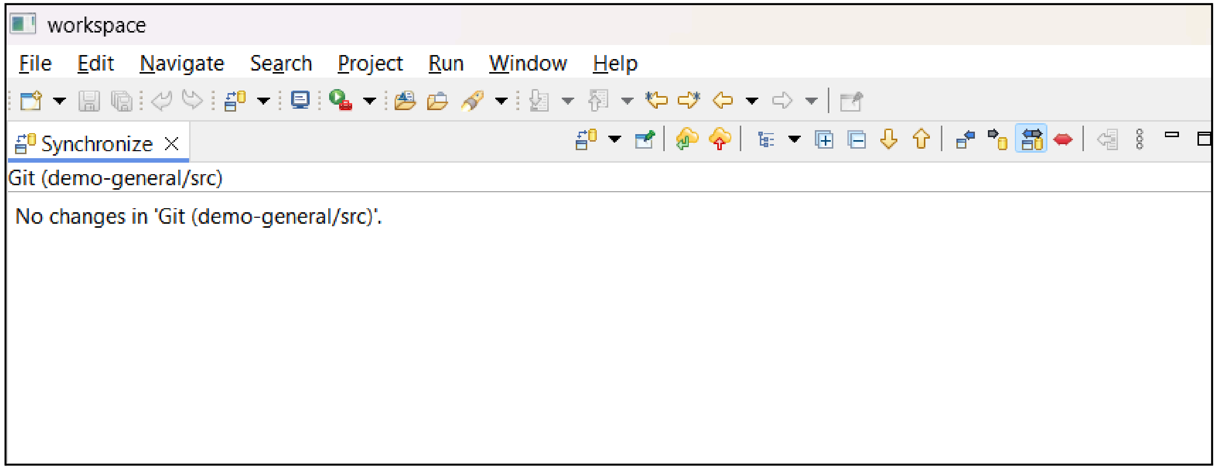
There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote

You have 2 local commits waiting to be pushed to GitHub.

Always available in the toolbar when there are local commits waiting to be pushed or Ctrl + P

From Eclipse: Check the team Synchronisation



Git (demo-general/src)

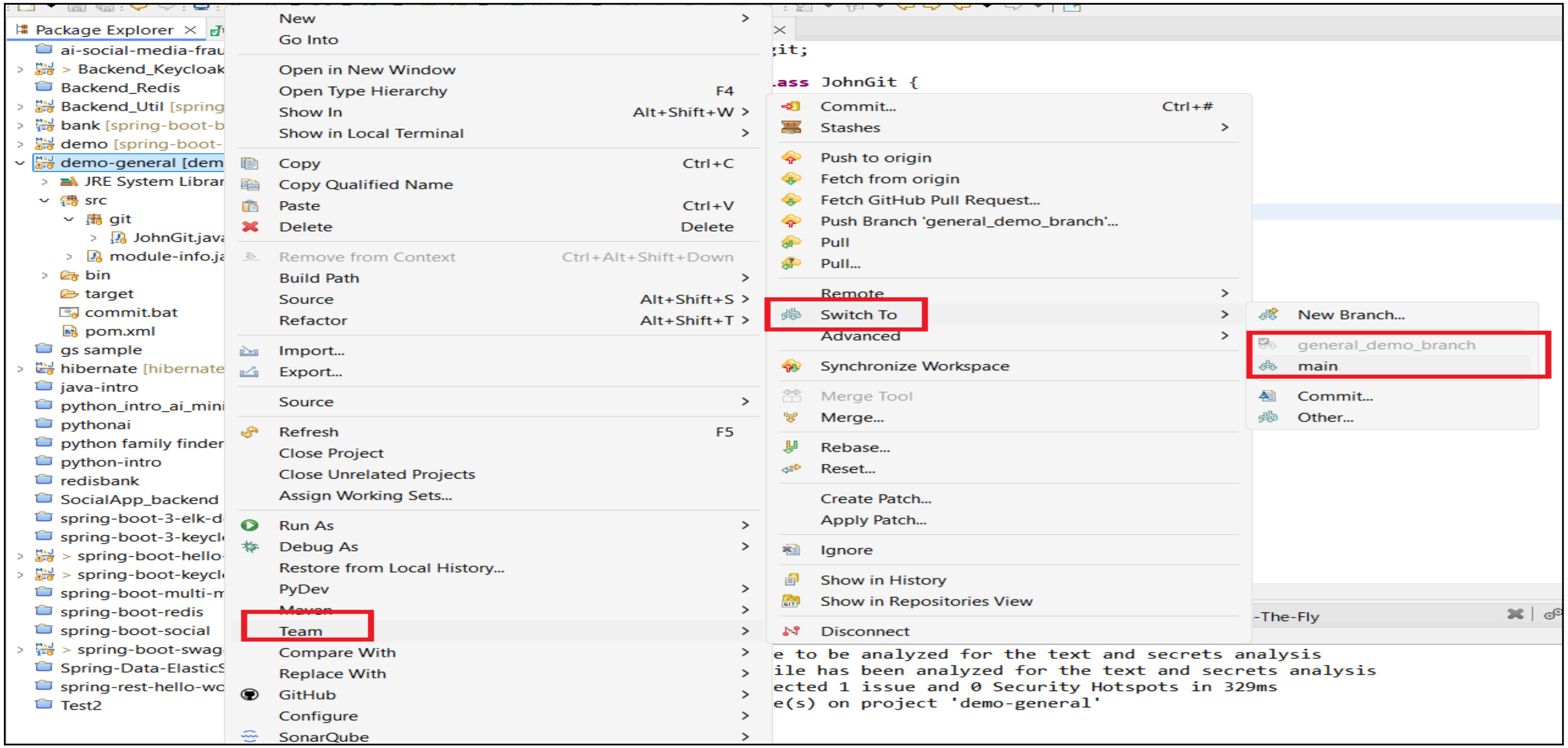
No changes in 'Git (demo-general/src)'.

On your GitHub Browser: Double check changes been updated

Other useful commands and techniques

How to Switch to your branch and sync with the main branch

From Eclipse:



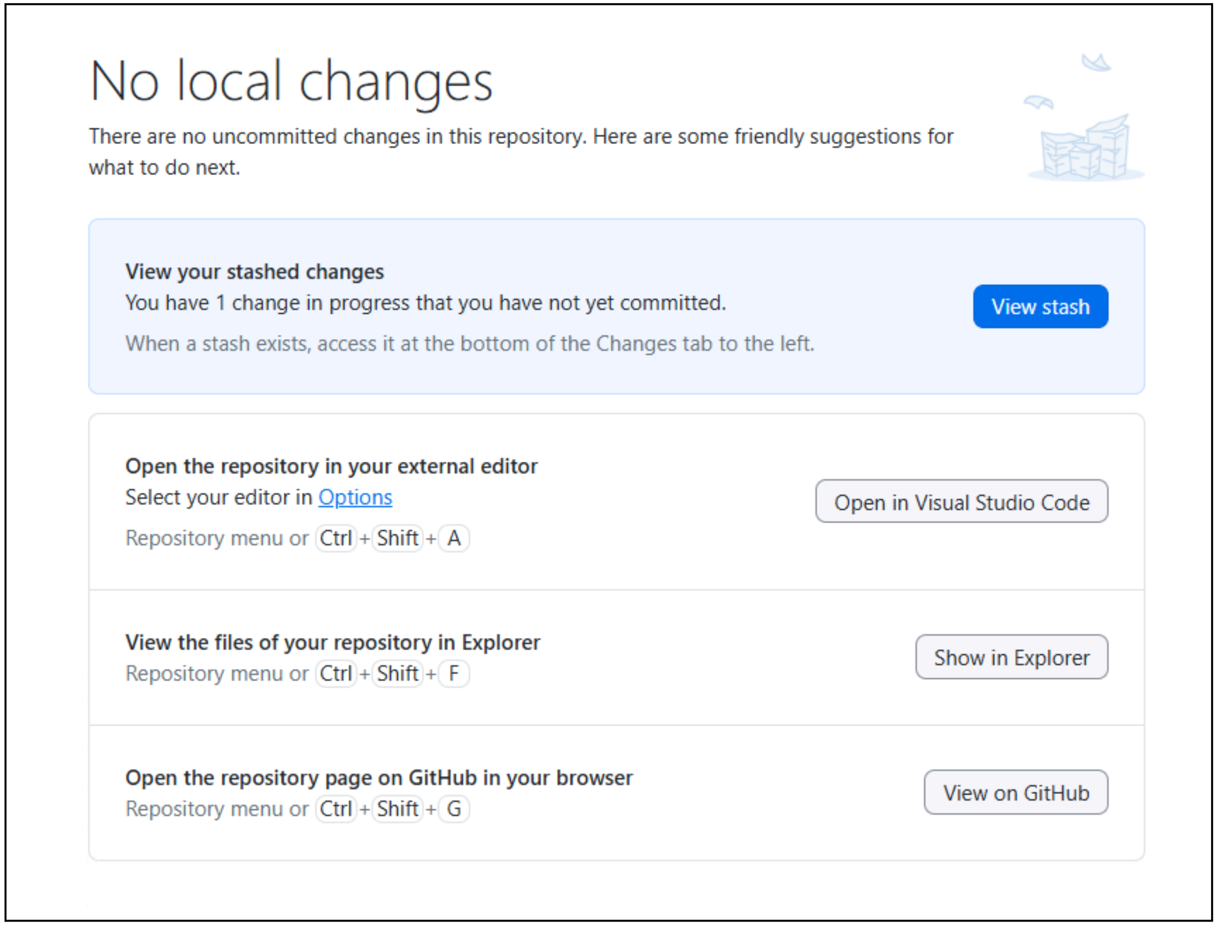
```
;it;

.ass JohnGit {
```

```
e to be analyzed for the text and secrets analysis
ile has been analyzed for the text and secrets analysis
ected 1 issue and 0 Security Hotspots in 329ms
e(s) on project 'demo-general'
```

,

When Missing commit items during switching branch to main or vice versa

Click on stash



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

View your stashed changes

You have 1 change in progress that you have not yet committed.

When a stash exists, access it at the bottom of the Changes tab to the left.

Open the repository in your external editor

Select your editor in Options

Repository menu or Ctrl + Shift + A

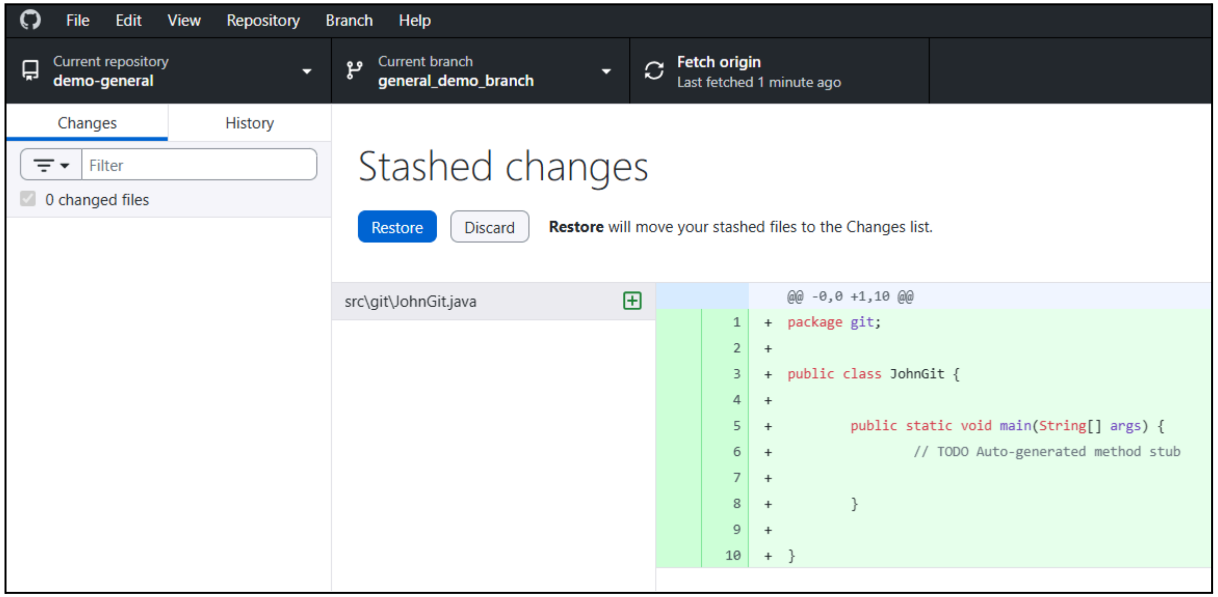
View the files of your repository in Explorer

Repository menu or Ctrl + Shift + F

Open the repository page on GitHub in your browser

Repository menu or Ctrl + Shift + G

Click restore



Stashed changes

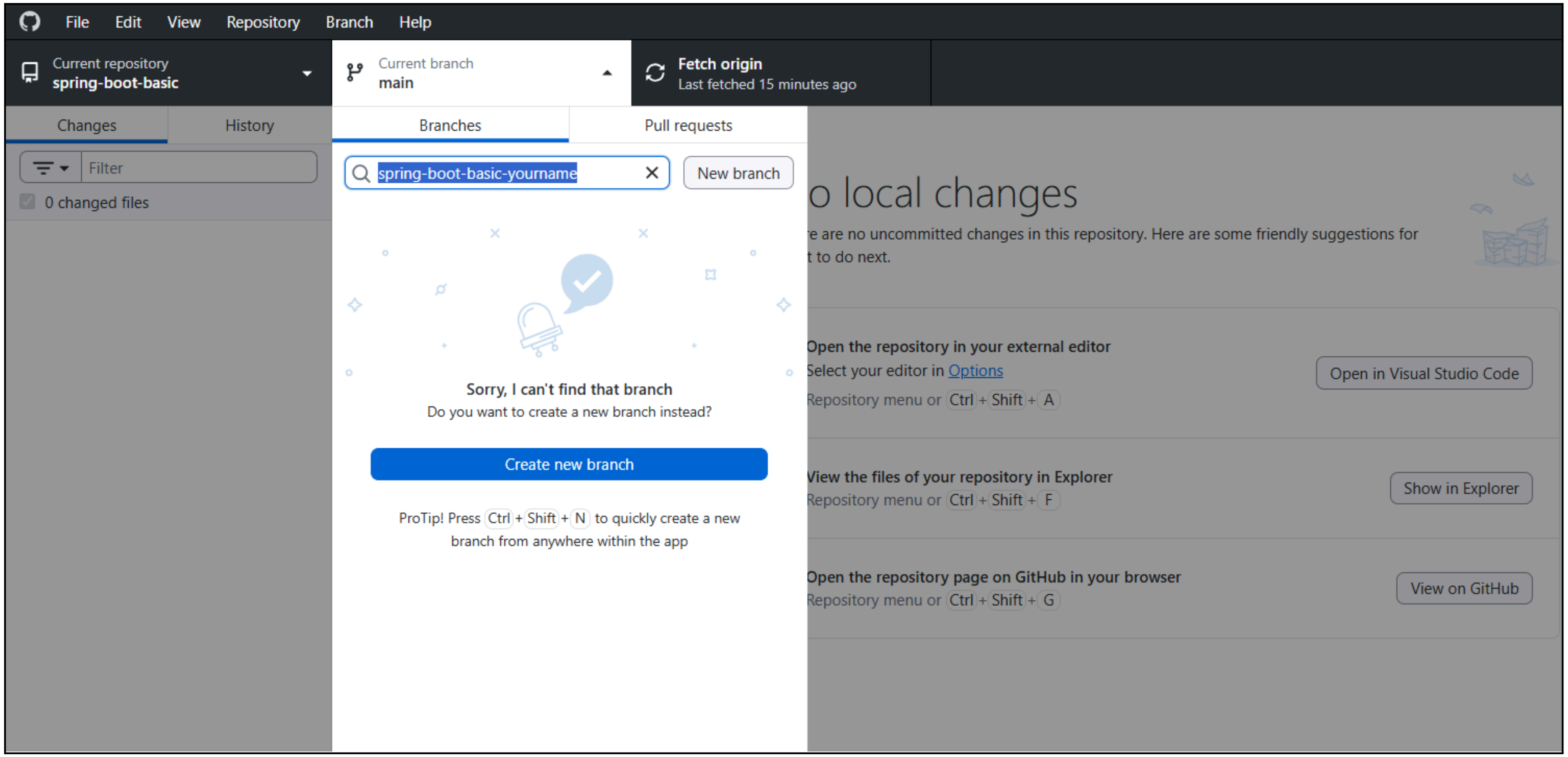
Restore will move your stashed files to the Changes list.

src\git\JohnGit.java

```
@@ -0,0 +1,10 @@
+ package git;
+
+ public class JohnGit {
+
+         public static void main(String[] args) {
+                 // TODO Auto-generated method stub
+
+         }
+
+ }
```

Create a new Branch

Create a new Branch



spring-boot-basic-yourname

Sorry, I can't find that branch

Do you want to create a new branch instead?

ProTip! Press Ctrl + Shift + N to quickly create a new branch from anywhere within the app

o local changes

e are no uncommitted changes in this repository. Here are some friendly suggestions for t to do next.

Open the repository in your external editor

Select your editor in Options

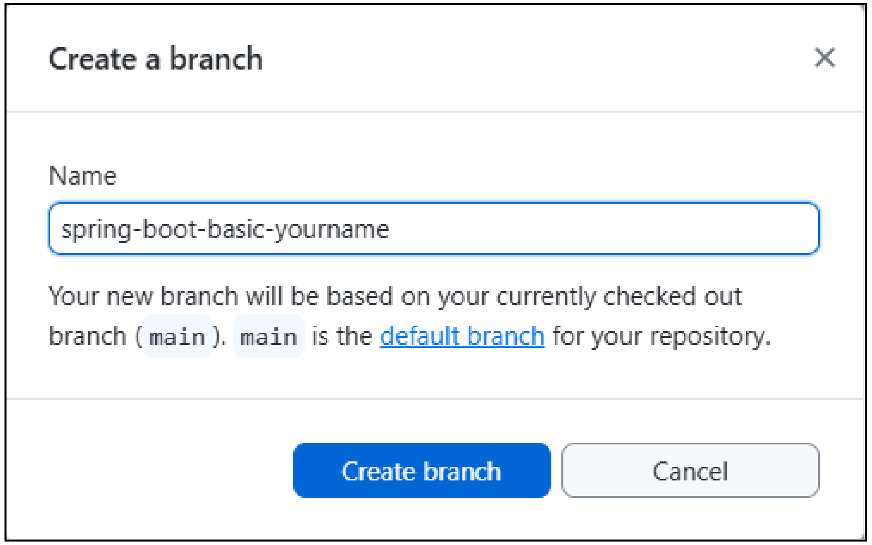
Repository menu or Ctrl + Shift + A

View the files of your repository in Explorer

Repository menu or Ctrl + Shift + F

Open the repository page on GitHub in your browser

Repository menu or Ctrl + Shift + G



Create a branch

Name

spring-boot-basic-yourname

Your new branch will be based on your currently checked out branch (`main`). `main` is the default branch for your repository.

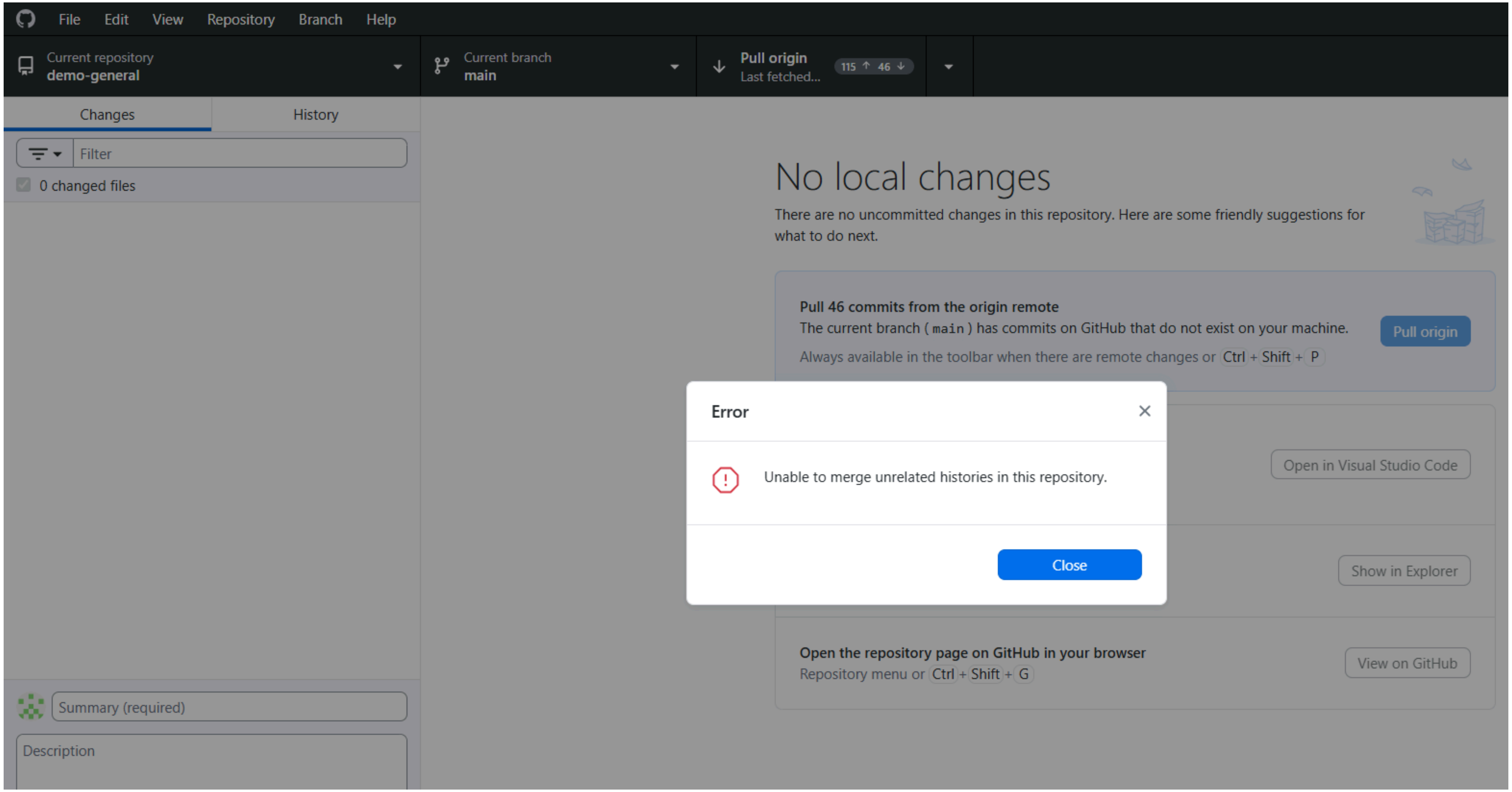
GIT reset

git reset --hard
Use when: You want to undo local changes and go back to your last commit.

git reset --hard @{u}
Use when: You want to **revert your local branch to match the remote**, removing local commits and changes.

Specify the branch if hitting error
git reset --hard origin/main

Error Unable to merge unrelated histories in this repository



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Pull 46 commits from the origin remote

The current branch (`main`) has commits on GitHub that do not exist on your machine.

Always available in the toolbar when there are remote changes or Ctrl + Shift + P

Open the repository page on GitHub in your browser

Repository menu or Ctrl + Shift + G

Error

Unable to merge unrelated histories in this repository.

Close

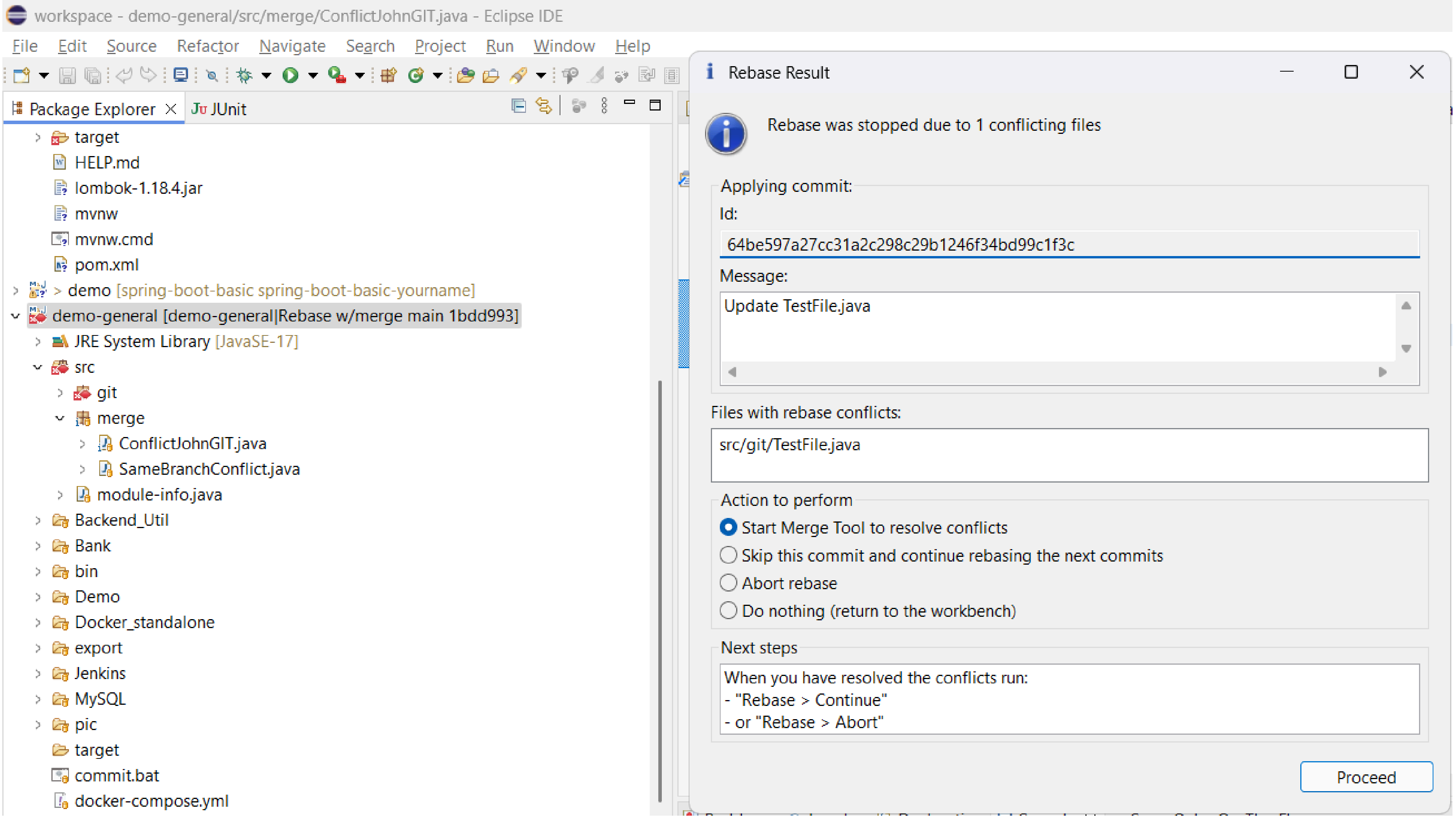
Backup local repo to avoid damages of the merging

Option 1: Delete the branch and recreate

Option 2: Do a merging and fix the conflict
Do a Git hard Reset then run the command below

```
D:\workspace\demo-general>git pull origin main --allow-unrelated-histories
From https://github.com/pcyuen98/demo-general
 * branch            main       -> FETCH_HEAD
Auto-merging pom.xml
CONFLICT (add/add): Merge conflict in pom.xml
Automatic merge failed; fix conflicts and then commit the result.
```

Doing a force rebase from Branch to sync with main



workspace - demo-general/src/merge/ConflictJohnGIT.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
target
HELP.md
lombok-1.18.4.jar
mvnw
mvnw.cmd
pom.xml
> demo [spring-boot-basic spring-boot-basic-yourname]
demo-general [demo-general|Rebase w/merge main 1bdd993]
JRE System Library [JavaSE-17]
src
git
merge
ConflictJohnGIT.java
SameBranchConflict.java
module-info.java
Backend_Util
Bank
bin
Demo
Docker_standalone
export
Jenkins
MySQL
pic
target
commit.bat
docker-compose.yml
Rebase Result
Proceed

Rebase was stopped due to 1 conflicting files

Applying commit:

Id: 64be597a27cc31a2c298c29b1246f34bd99c1f3c

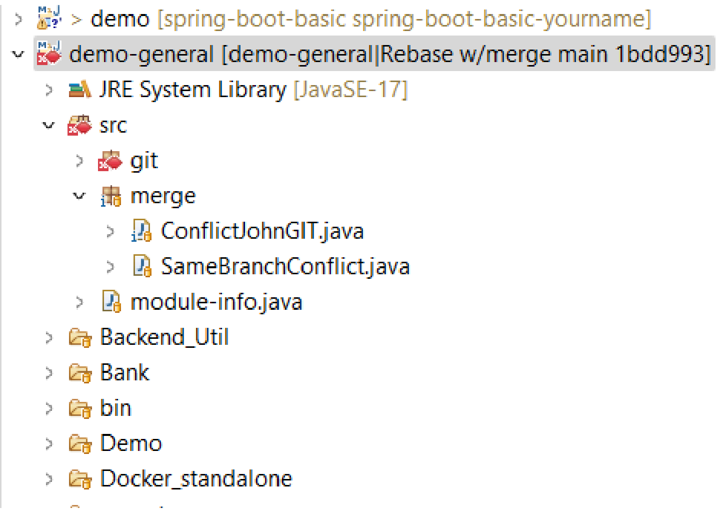
Message: Update TestFile.java

Files with rebase conflicts:
src/git/TestFile.java

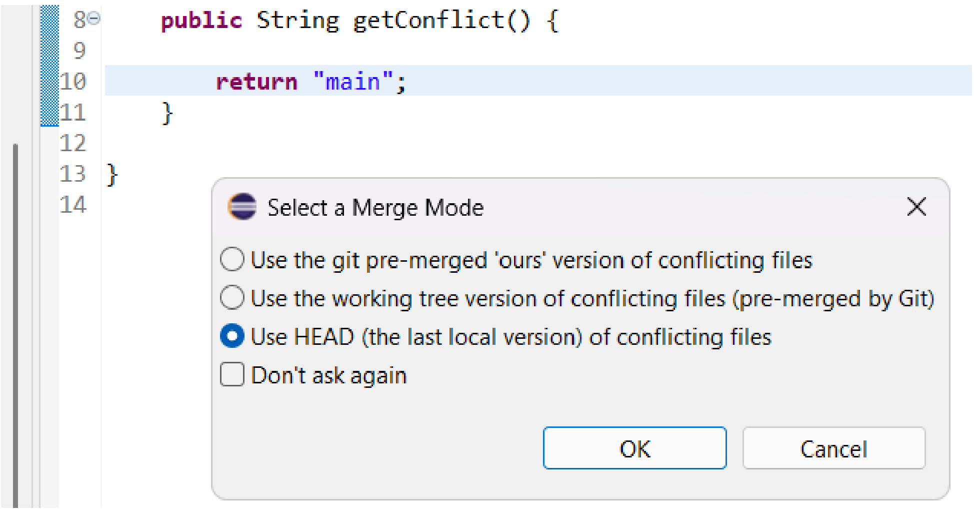
Action to perform
- ● Start Merge Tool to resolve conflicts
- ○ Skip this commit and continue rebasing the next commits
- ○ Abort rebase
- ○ Do nothing (return to the workbench)

Next steps

When you have resolved the conflicts run:
- "Rebase > Continue"
- or "Rebase > Abort"



- > demo [spring-boot-basic spring-boot-basic-yourname]
- demo-general [demo-general|Rebase w/merge main 1bdd993]
  - JRE System Library [JavaSE-17]
  - src
    - git
    - merge
      - ConflictJohnGIT.java
      - SameBranchConflict.java
    - module-info.java
  - Backend_Util
  - Bank
  - bin
  - Demo
  - Docker_standalone



```
    public String getConflict() {

        return "main";
    }

}
```

Select a Merge Mode

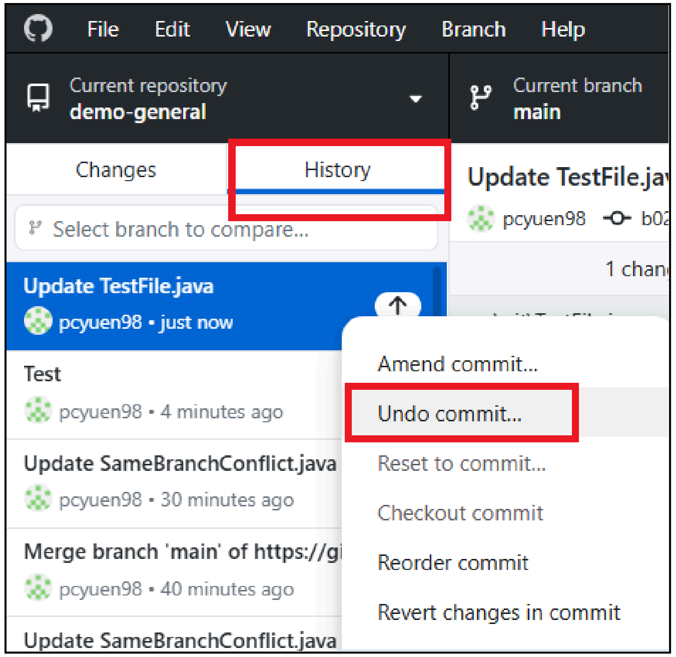
○ Use the git pre-merged 'ours' version of conflicting files

○ Use the working tree version of conflicting files (pre-merged by Git)

● Use HEAD (the last local version) of conflicting files

□ Don't ask again

How to Undo the commit



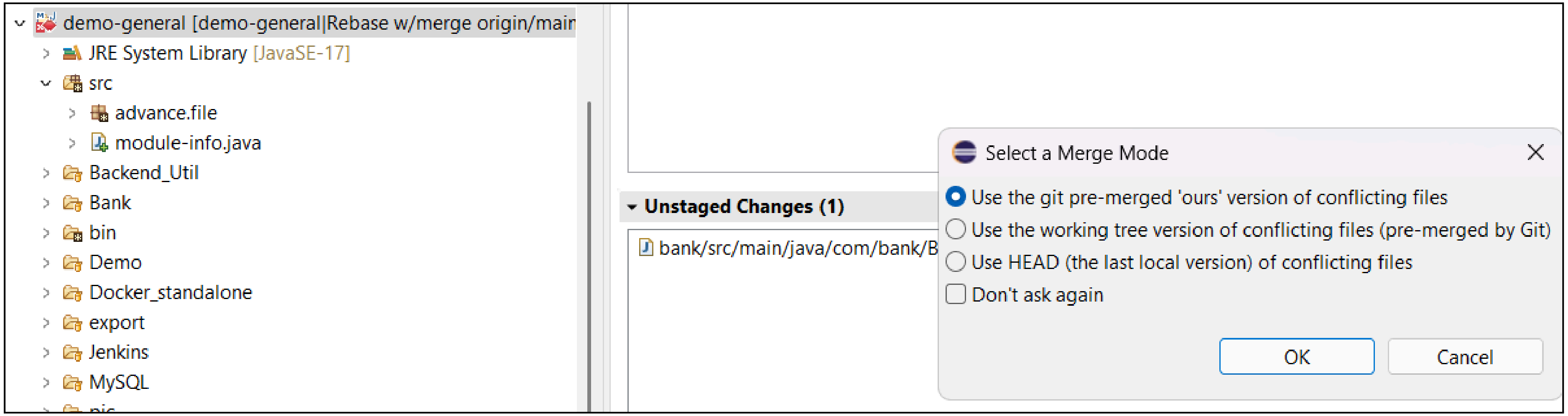
Git pull main branch

Common Error: You are not currently on a branch.
Please specify which branch you want to merge with.

Run command below:

```
git pull origin main
```

Merging Error when rebase or reset



● Use the git pre-merged 'ours' version of conflicting files
○ Use the working tree version of conflicting files (pre-merged by Git)
○ Use HEAD (the last local version) of conflicting files
□ Don't ask again

Better to backup the whole repo locally and hard reset the branch to avoid damages to the local files and conflicts

Run Jenkins in the terminal manually

```
java -jar /root/.jenkins/workspace/build-bank-demo/Bank/target/spring-boot-bank-example-0.0.1-SNAPSHOT.jar
--spring.config.location=file:///root/.jenkins/workspace/application-prod.properties  --server.port=9001
```

Run Maven build in the terminal manually

```
mvn clean install -Dspring.config.location=file:///root/.jenkins/workspace/application-prod.properties
```

Check and Monitor Docker Resources

1. docker stats – Live container resource usage

```
docker stats
```

This displays real-time resource usage for running containers:

```
Container       CPU %           MEM USAGE / LIMIT MEM %          NET I/O          BLOCK I/O     PIDS
```

```
docker stats <container_name_or_id>
```

2. docker inspect – Detailed resource limits per container

```
docker inspect <container_name_or_id>
"Memory": 536870912,
"CpuShares": 512,
"NanoCpus": 1000000000
```

3. docker container top – Show running processes in a container

```
docker container top <container_name_or_id>
```

4. docker system df – Show disk usage (images, volumes, etc.)

```
docker system df
```