

JAVAGURU AGILE WEB DEVELOPMENT

LESSON 1

ABOUT US

- ▶ Teaching more than **8 years**
- ▶ Over **1500 students** graduated
- ▶ Constantly **expanding list** of available courses
- ▶ **Assisting** students to get a job after graduation

HOMEWORK ASSIGNMENTS

- ▶ Required for course completion
 - ▶ All meeting (task) assignments
- ▶ Everything else
 - ▶ Optional, but highly recommended
- ▶ Upload to Git Repository

GETTING STARTED

▶ Toolbox

- ▶ **JDK 8:** <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- ▶ **IntelliJ IDEA Community Edition:** <https://www.jetbrains.com/idea/download/>
- ▶ **Git:** <https://git-scm.com/downloads>

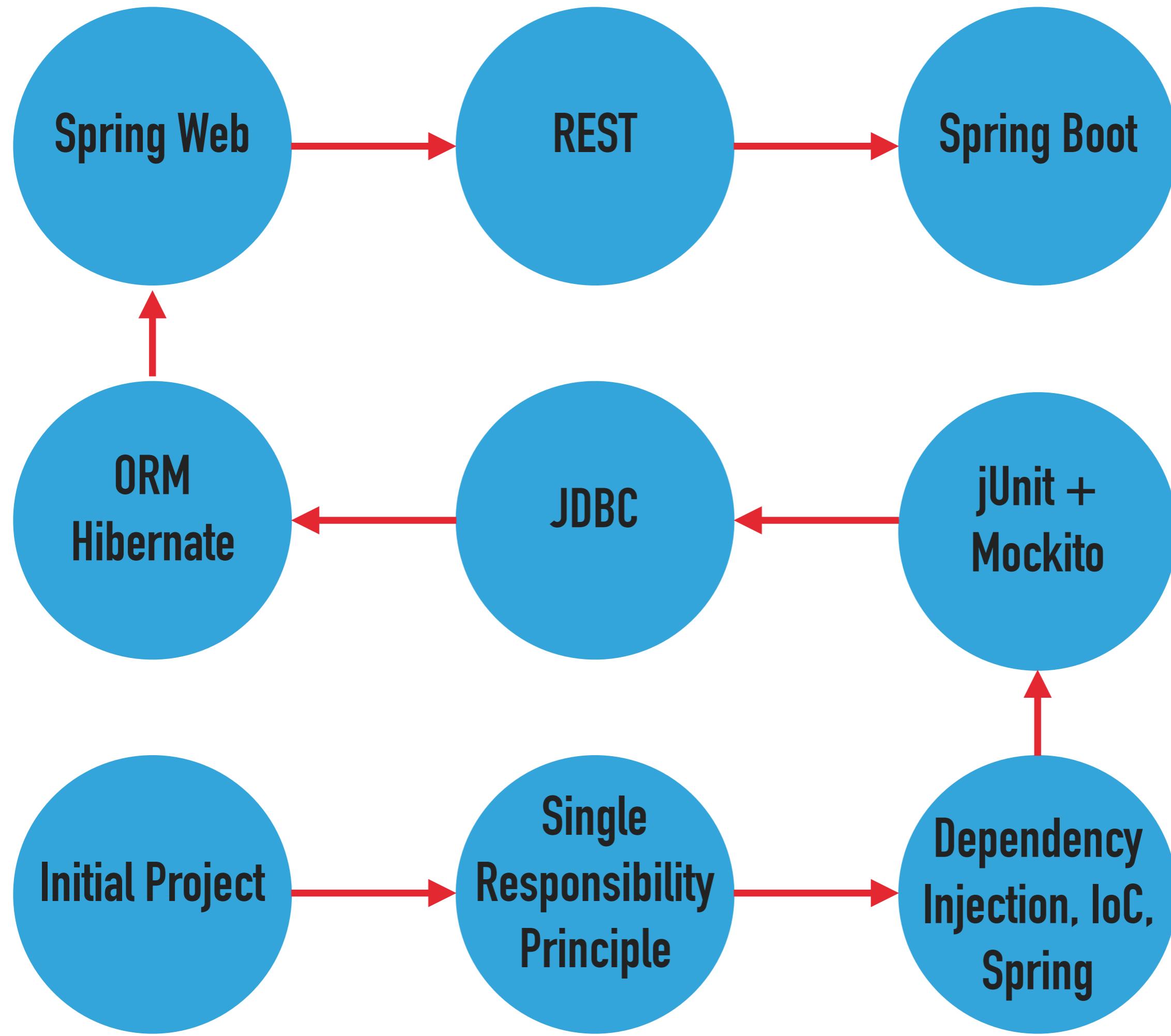
▶ Collaboration

- ▶ **GitHub:** <https://github.com/>

▶ Communication

- ▶ **Skype:** <https://www.skype.com/en/get-skype/>

Course structure



VERSION CONTROL

WHAT IS VERSION CONTROL

- ▶ **Version control systems** are a category of software tools that help a software team manage changes to source code over time.
- ▶ Version control software keeps track of every modification to the code in a special kind of database
- ▶ Version control helps teams solve these kinds of problems, tracking every individual change by each contributor and helping prevent concurrent work from conflicting

BENEFITS

- ▶ Developing software without using version control is risky, like not having backups. Version control can also enable developers to move faster and it allows software teams to preserve efficiency and agility as the team scales to include more developers.

HISTORY

- ▶ A complete long-term change history of every file. This means every change made by many individuals over the years.
- ▶ Changes include the creation and deletion of files as well as edits to their contents.
- ▶ Different VCS tools differ on how well they handle renaming and moving of files. This history should also include the author, date and written notes on the purpose of each change.

BRANCHING

- ▶ Branching and merging. Having team members work concurrently is a no-brainer, but even individuals working on their own can benefit from the ability to work on independent streams of changes.
- ▶ Creating a "branch" in VCS tools keeps multiple streams of work independent from each other while also providing the facility to merge that work back together, enabling developers to verify that the changes on each branch do not conflict
- ▶ Many software teams adopt a practice of branching for each feature or perhaps branching for each release, or both

TRACEABILITY

- ▶ Being able to trace each change made to the software and connect it to project management and bug tracking software such as [Jira](#), and being able to annotate each change with a message describing the purpose and intent of the change can help not only with root cause analysis and other forensics

- ▶ Having the annotated history of the code at your fingertips when you are reading the code, trying to understand what it is doing and why it is so designed can enable developers to make correct and harmonious changes that are in accord with the intended long-term design of the system



git

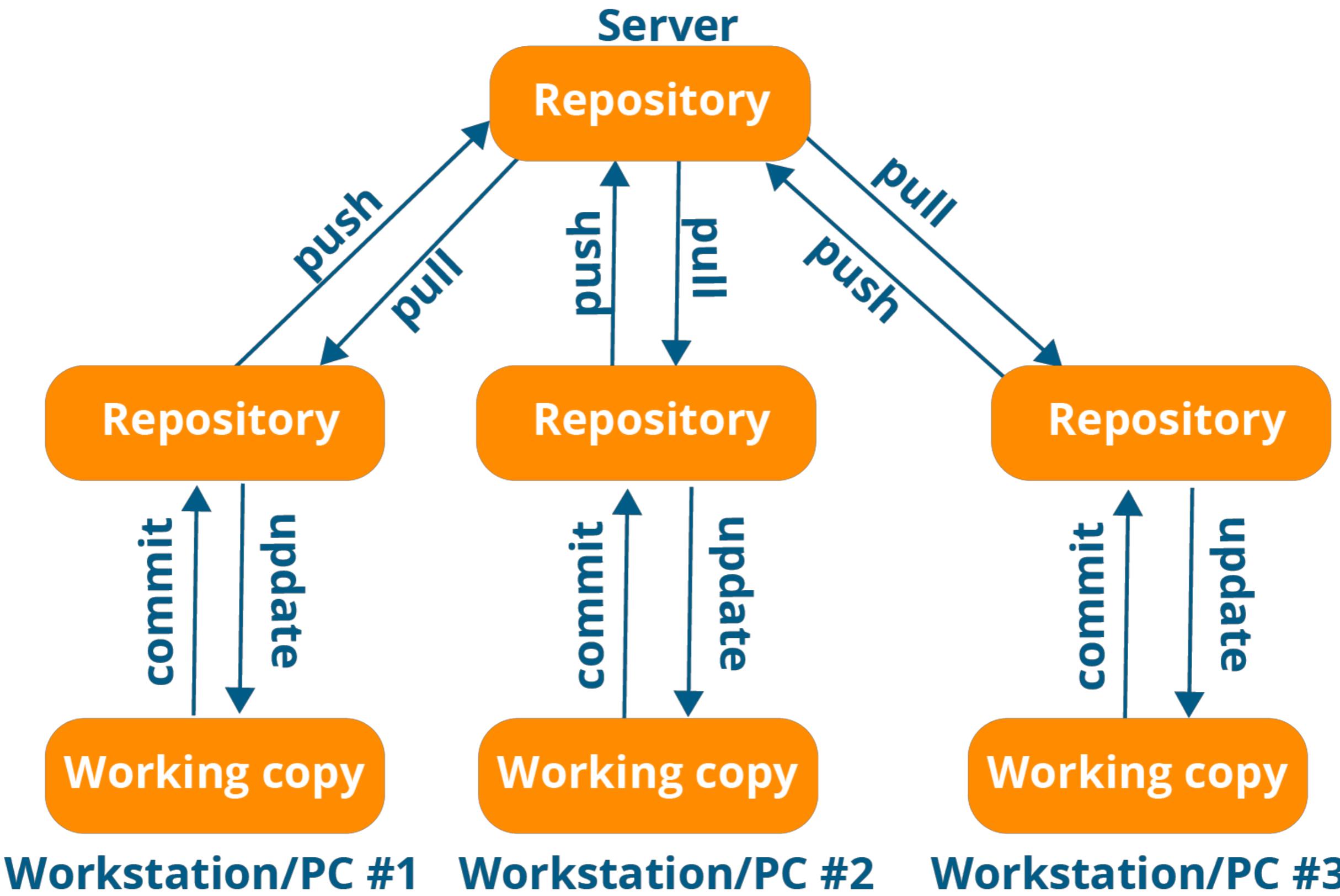
WHAT IS A GIT

- ▶ By far, the most widely used modern version control system in the world today is Git.
- ▶ Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel.
- ▶ A staggering number of software projects rely on Git for version control, including commercial projects as well as open source

GIT IS A QUALITY OPEN SOURCE PROJECT

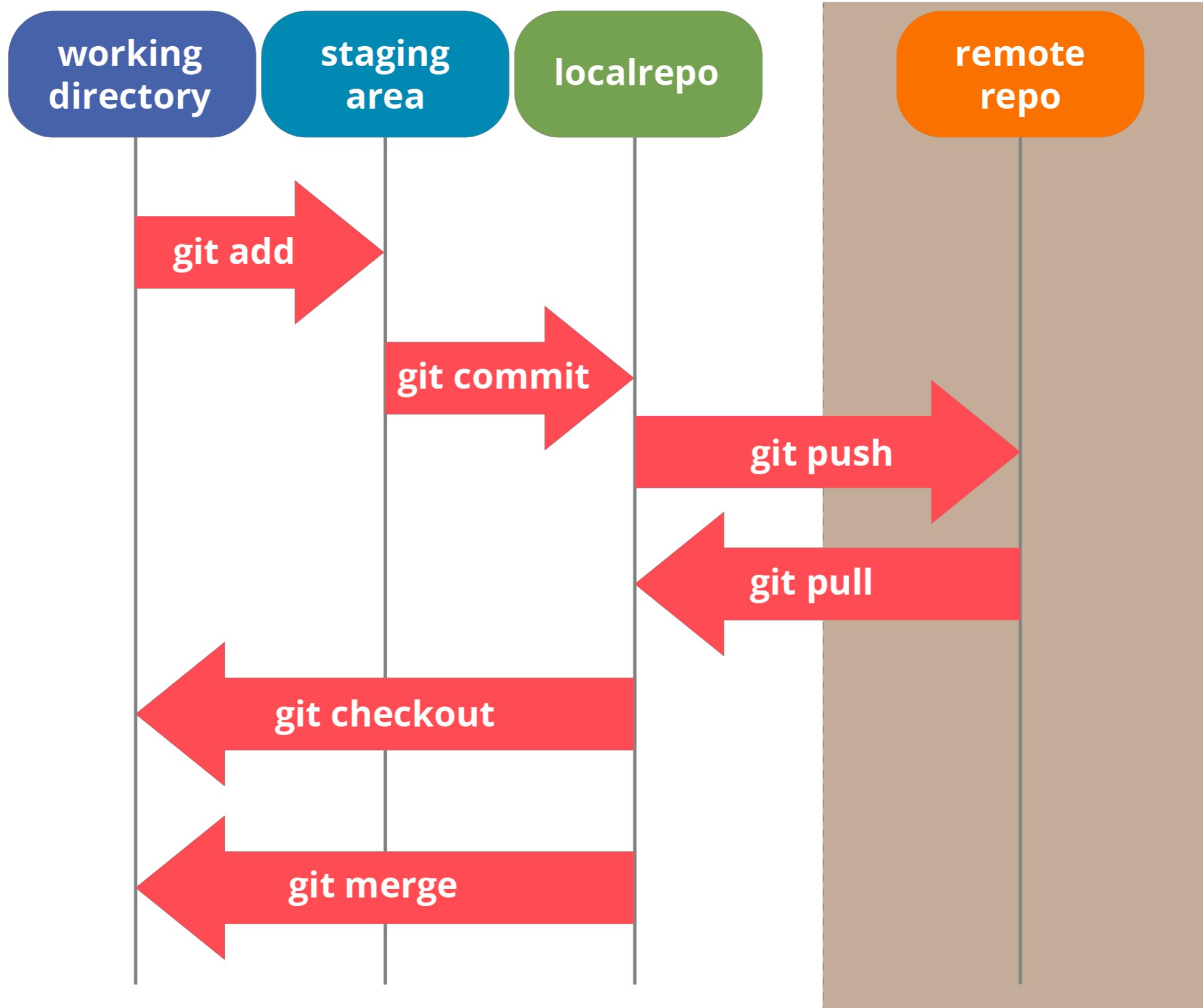
- ▶ Git is a very well supported open source project with over a decade of solid stewardship. The project maintainers have shown balanced judgment and a mature approach to meeting the long term needs of its users with regular releases that improve usability and functionality. The quality of the open source software is easily scrutinized and countless businesses rely heavily on that quality.
- ▶ Git enjoys great community support and a vast user base. Documentation is excellent and plentiful, including books, tutorials and dedicated web sites. There are also podcasts and video tutorials

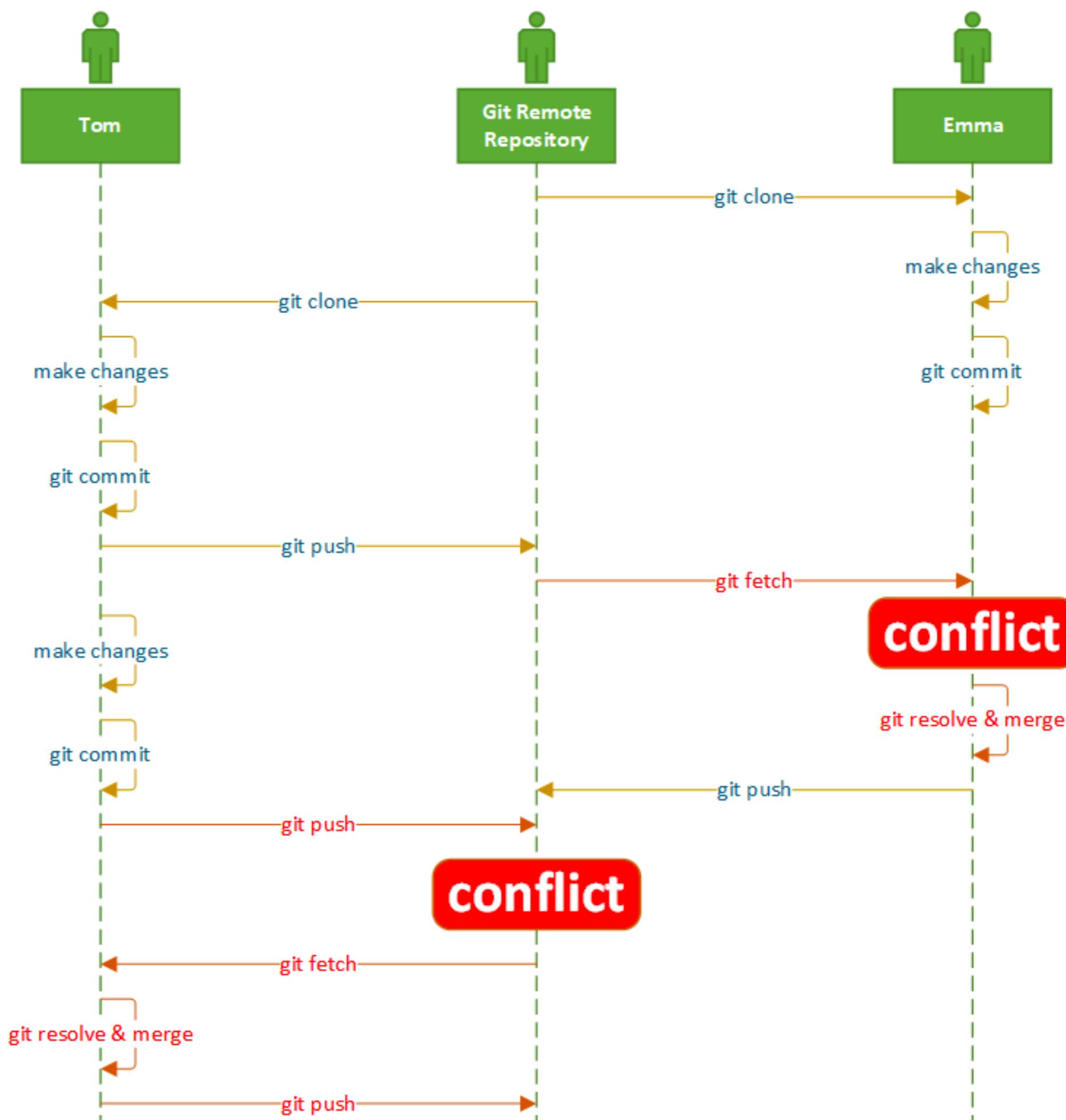
Distributed version control system



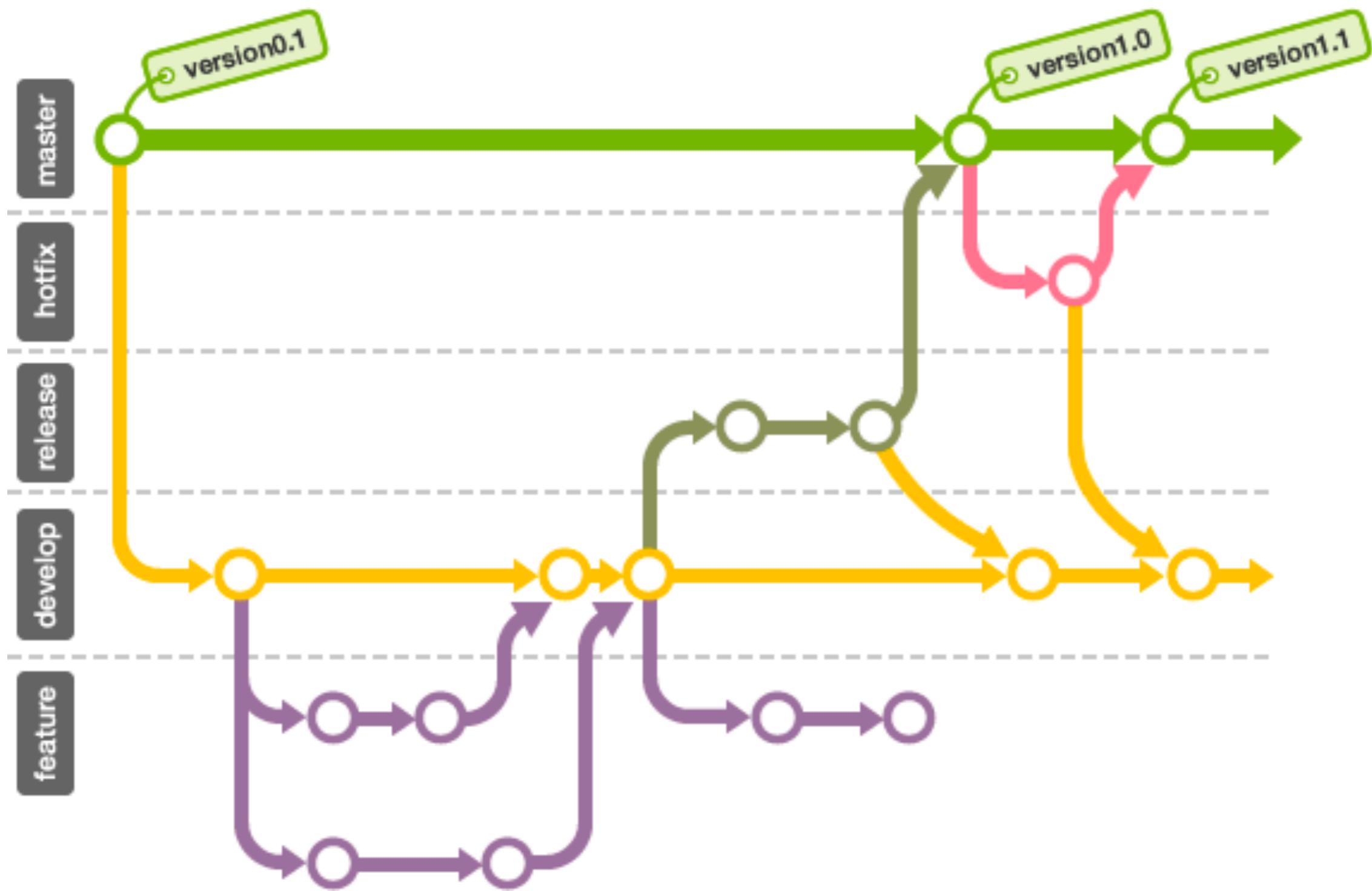
Local

Remote





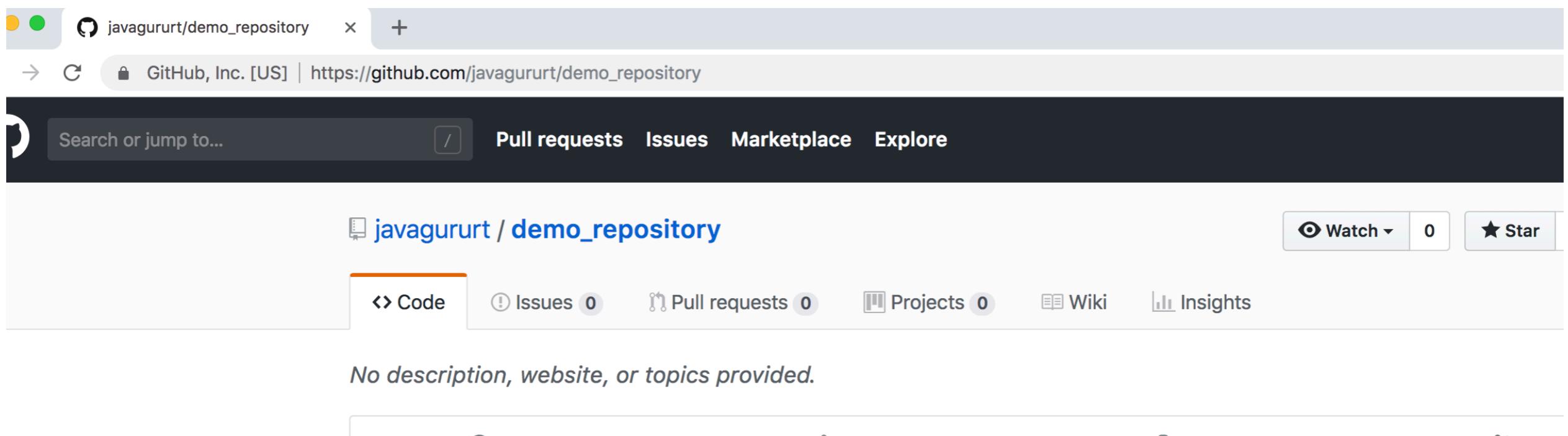
BRANCHING



CLONE FROM GIT
REPOSITORY

CLONE FROM GITHUB

- ▶ Step 1. Get repository link
- ▶ Step 2. Come to repository in your browser.



FIRST PROJECT

LOOKS LIKE SOMETHING LIKE THIS

The screenshot shows a GitHub repository page for 'javagururt / demo_repository'. The repository has no description, website, or topics provided. It contains 1 commit, 1 branch, 0 releases, and 1 contributor. The latest commit was made by 'ruslanstufetulovs' 2 minutes ago. The README.md file contains the text 'demo_repository'.

javagururt / **demo_repository**

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

Watch 0 Star 0 Fork 0

No description, website, or topics provided.

1 commit 1 branch 0 releases 1 contributor

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

ruslanstufetulovs first commit Latest commit caf41a6 2 minutes ago

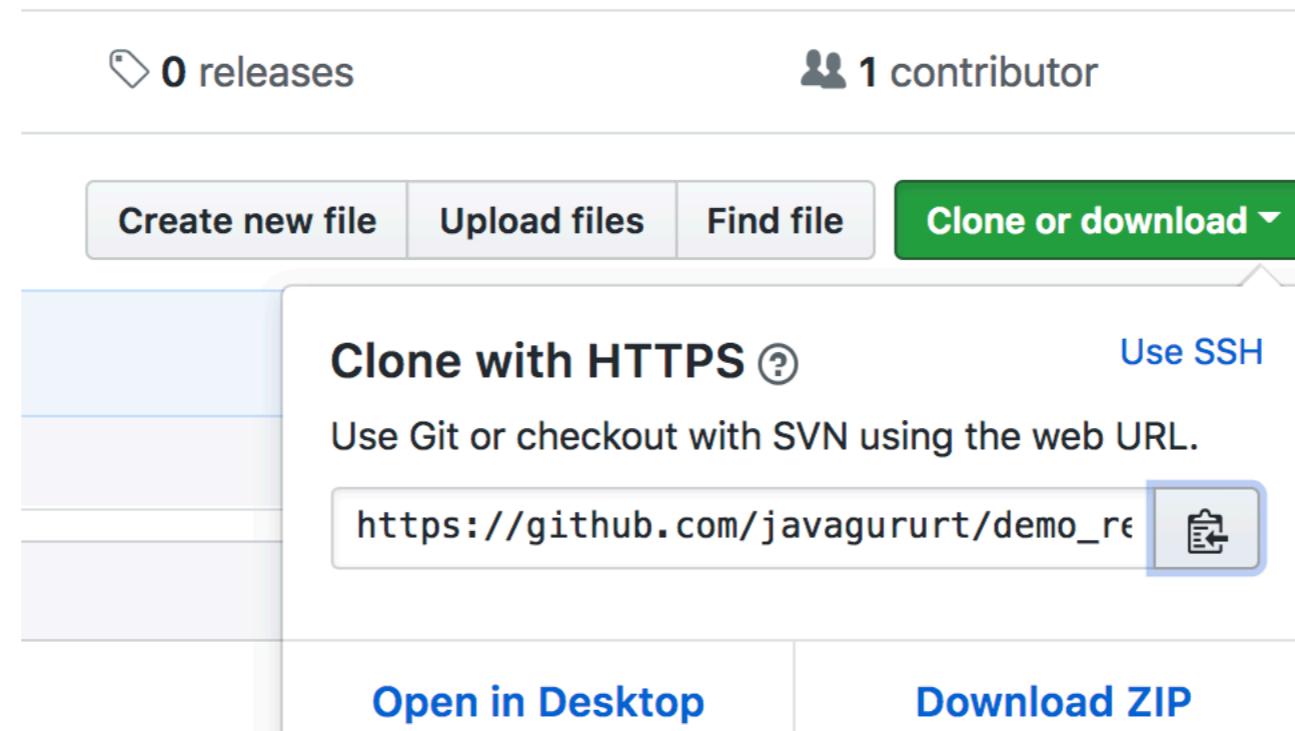
README.md first commit 2 minutes ago

README.md

demo_repository

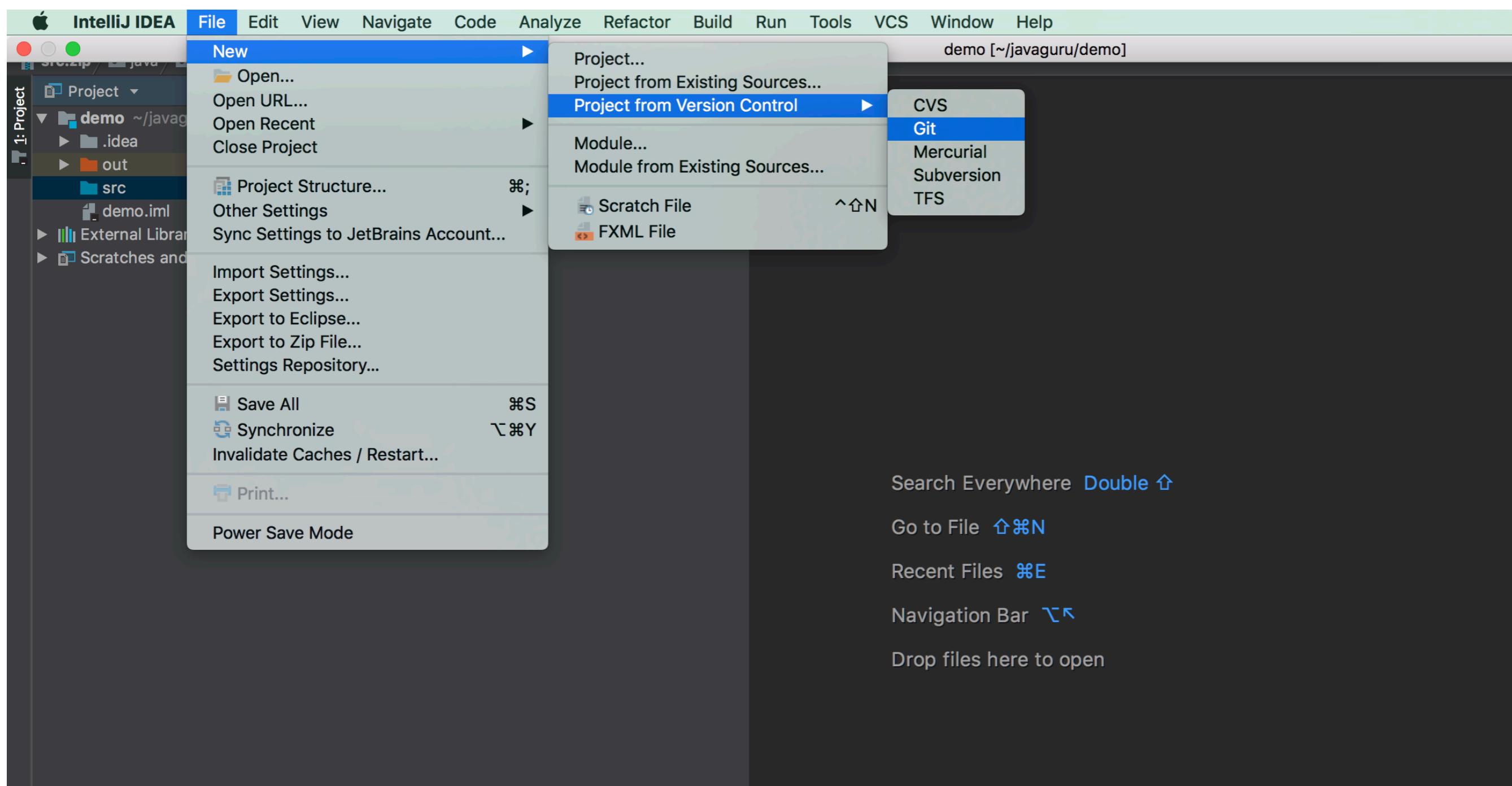
GETTING A REPOSITORY LINK

- ▶ Step 3. Press green button “Clone or Download” and copy link (*.git)



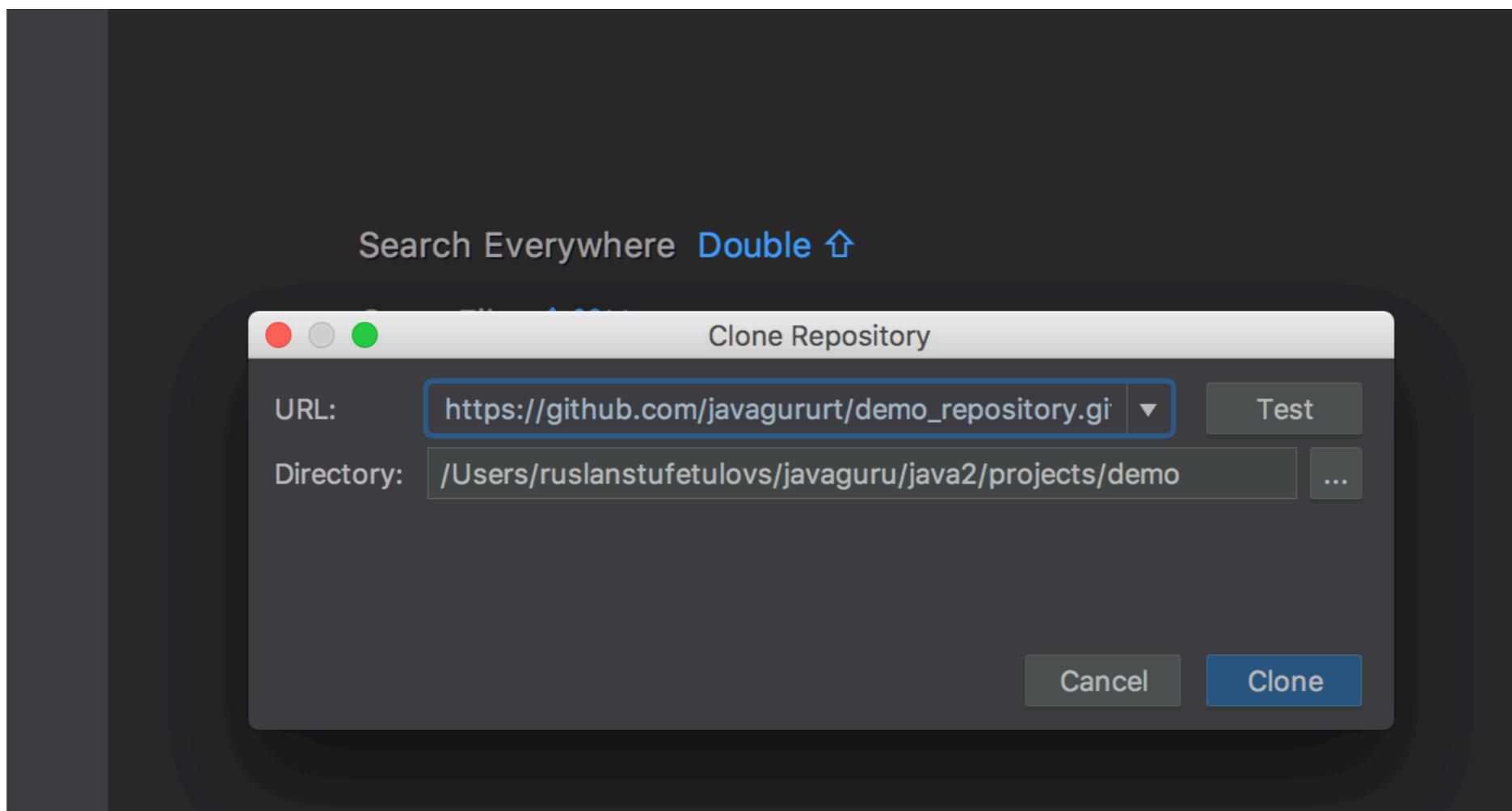
CLONE FROM GITHUB

▶ Step 4. Checkout project from repository



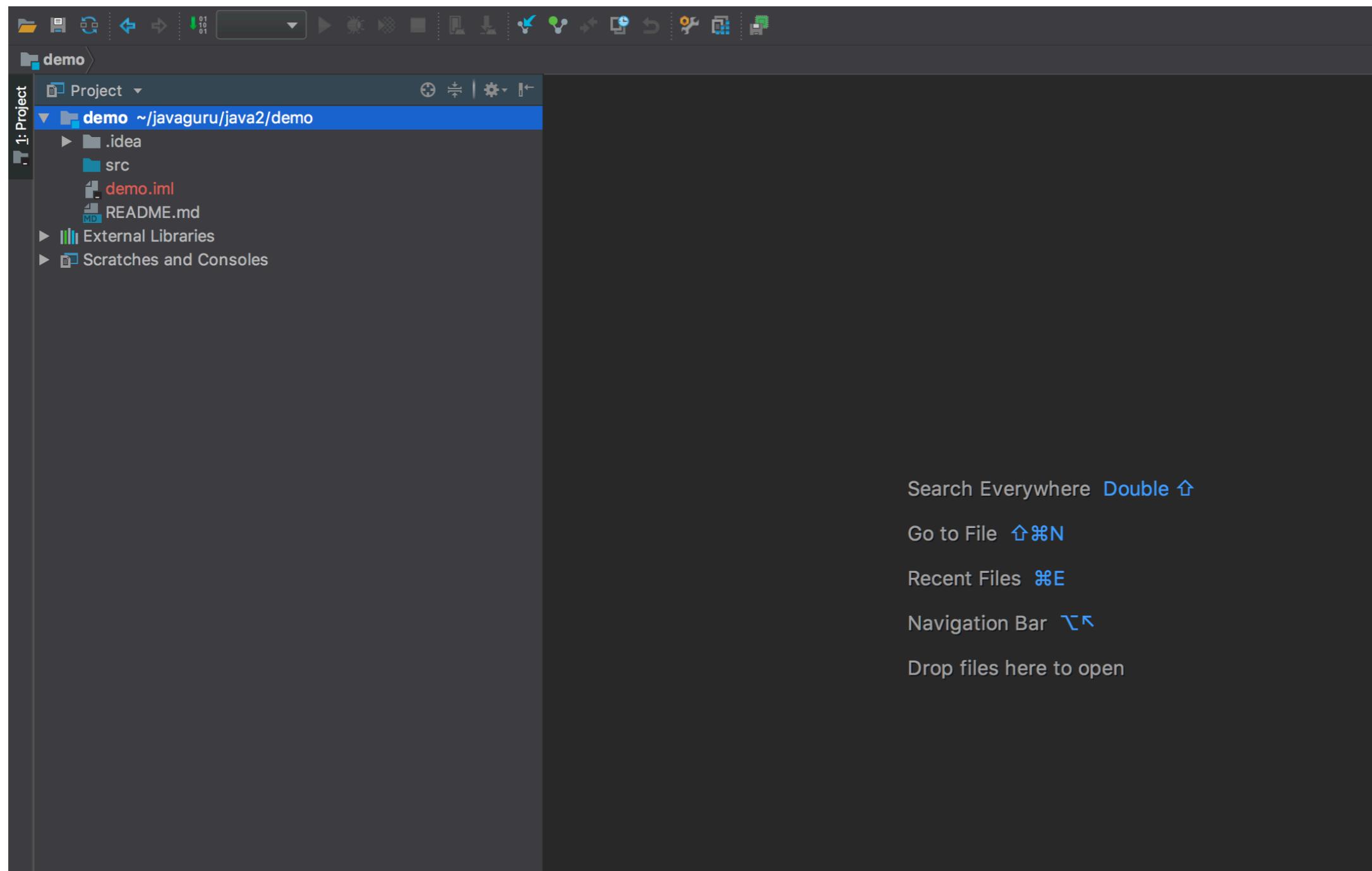
CLONE

- ▶ Step 5. Enter copied url into “URL” field and choose project directory. Then press “Clone”



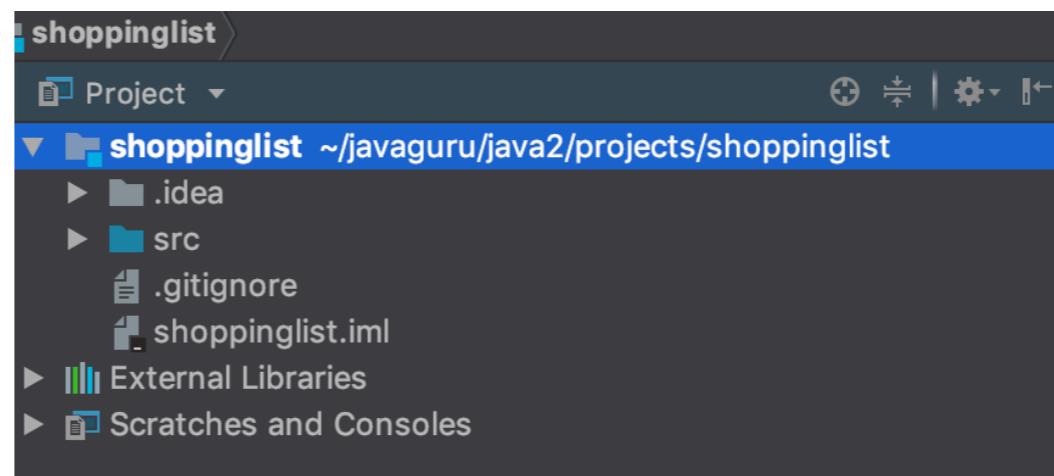
FIRST PROJECT

WELL DONE!



COMMIT TO GIT
REPOSITORY

► Step 1. Copy .gitignore to your project



► Step 2. Go to terminal and enter “git init” to initialize git repository

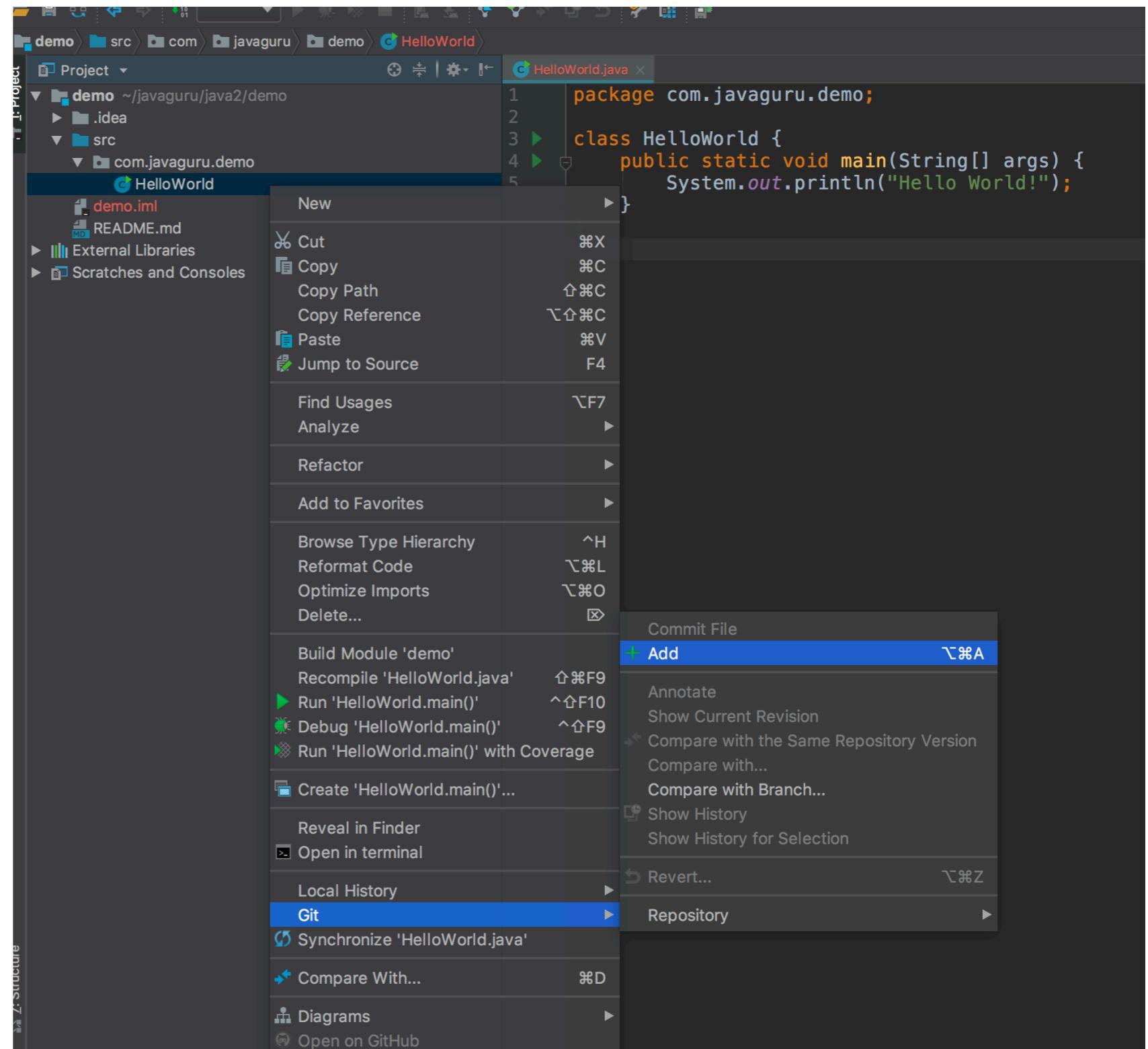
```
Ruslanss-MacBook-Pro:todoList ruslanstufetulovs$ git init
Initialized empty Git repository in /Users/ruslanstufetulovs/javaguru/todoList/.git/
Ruslanss-MacBook-Pro:todoList ruslanstufetulovs$
```

The screenshot shows a macOS Terminal window. The title bar says 'Terminal'. The main pane contains the following text:

- A green plus sign followed by the text 'Ruslanss-MacBook-Pro:todoList ruslanstufetulovs\$ git init'
- A red X followed by the text 'Initialized empty Git repository in /Users/ruslanstufetulovs/javaguru/todoList/.git/'
- Another green plus sign followed by the text 'Ruslanss-MacBook-Pro:todoList ruslanstufetulovs\$'

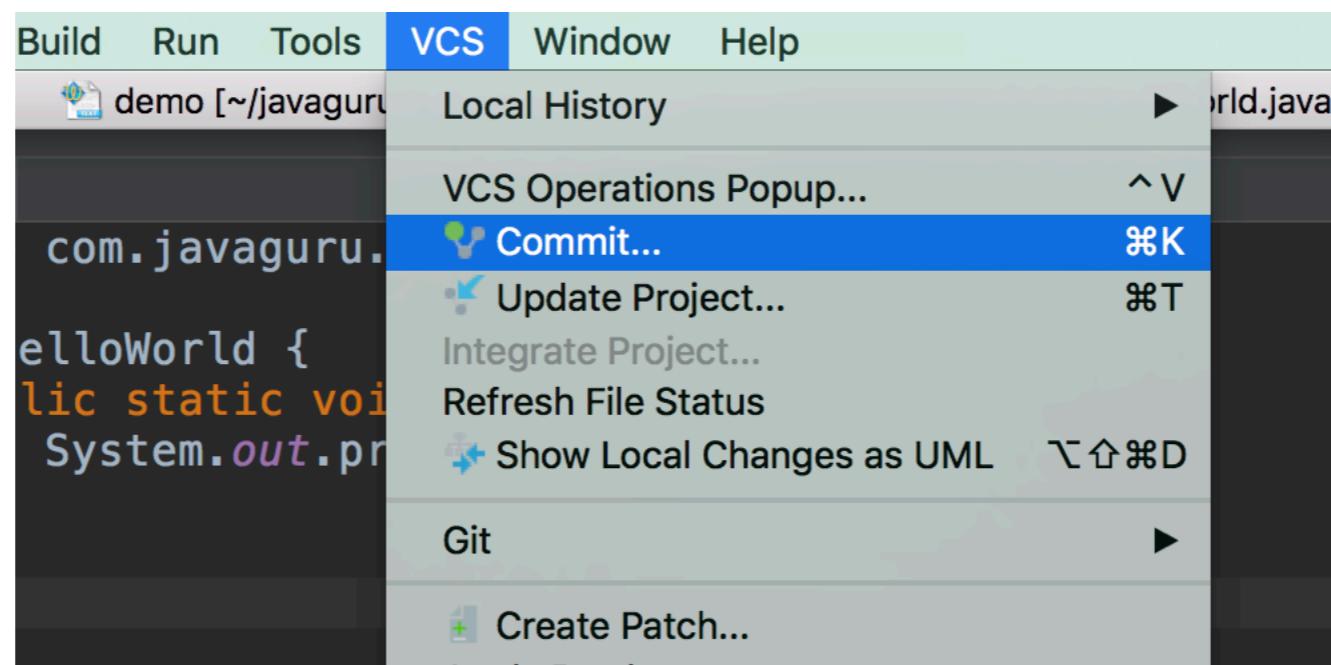
COMMIT TO GIT REPOSITORY

- ▶ Step 3. Select “unversioned” file
- ▶ Step 4. Use right-click on file and select “Git”
- ▶ Step 5. After that select “Add” in git menu



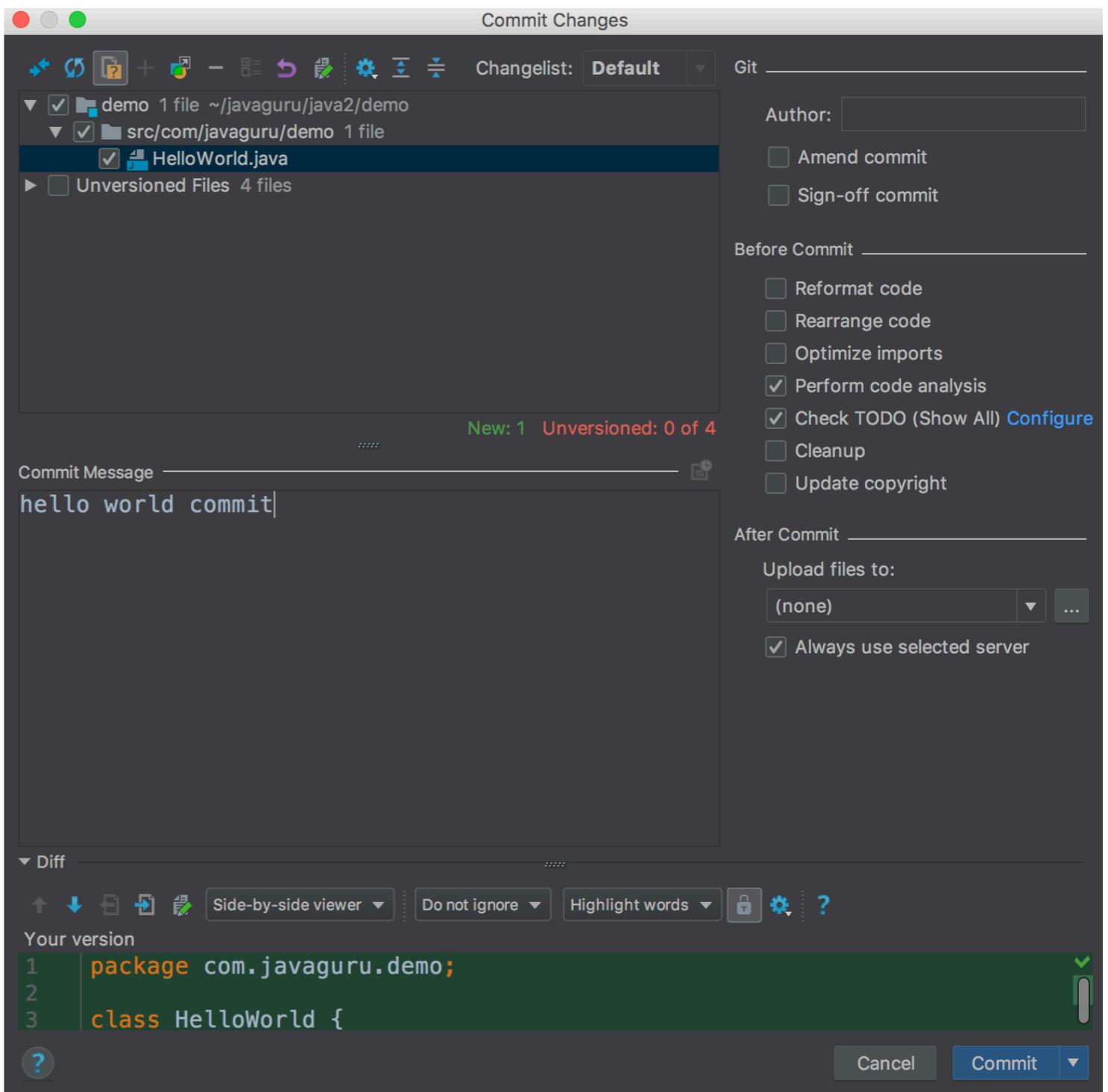
COMMIT TO GIT REPOSITORY

- ▶ Step 6. Go to the top panel and select “VCS”
- ▶ Step 7. Then select “Commit...”



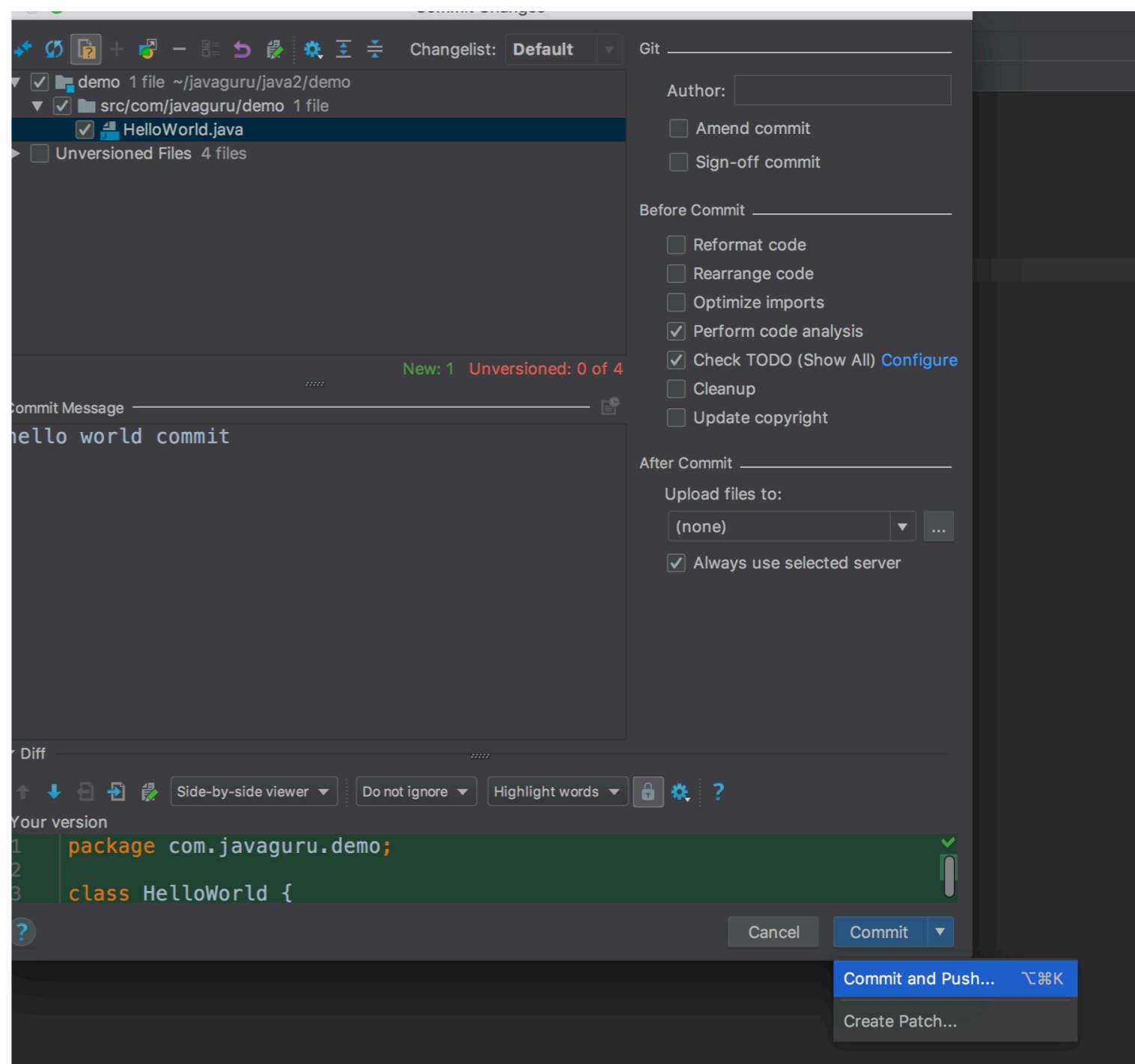
COMMIT TO GIT REPOSITORY

- ▶ Step 8. Enter commit message. Message must describe the purpose of commit.



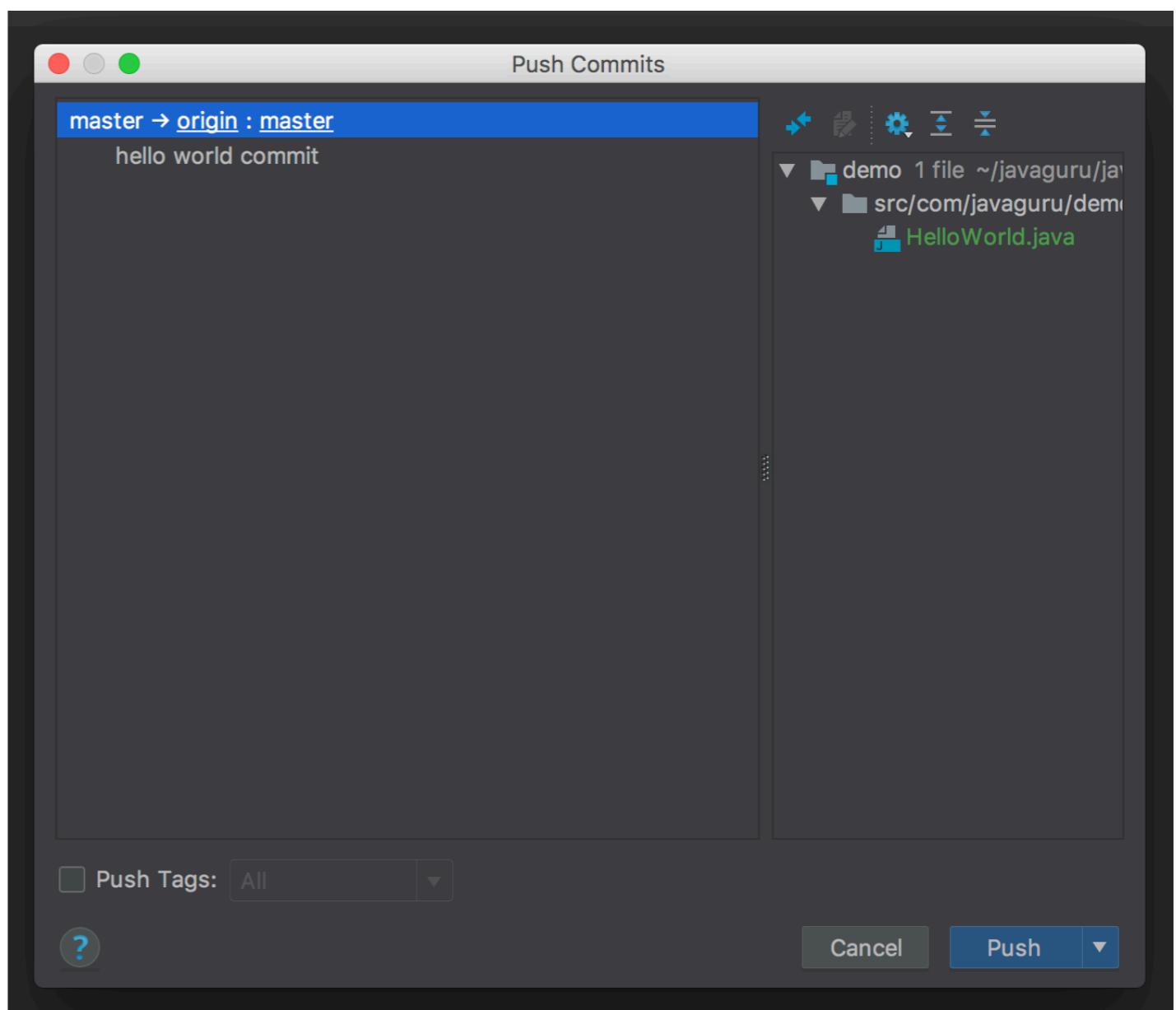
COMMIT TO GIT REPOSITORY

- ▶ Step 9. Press on “dropdown” triangle and choice “Commit and Push”...



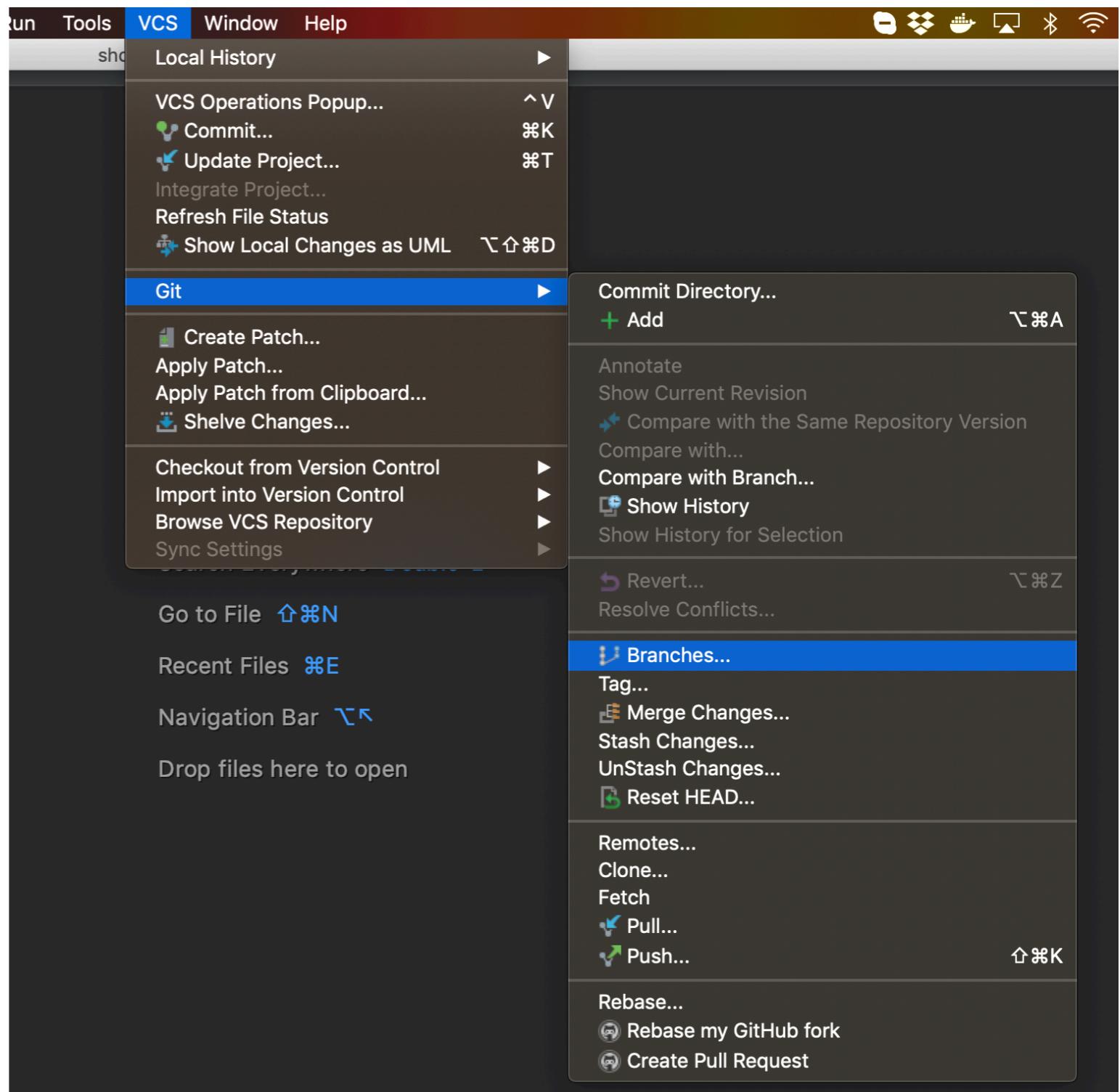
COMMIT TO GIT REPOSITORY

- ▶ Step 10. Check that your commit will be pushed in correct branch and press “Push”.



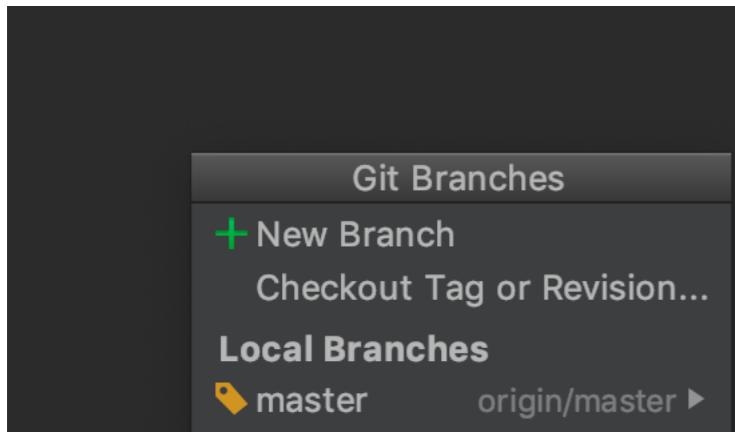
HOW TO CREATE NEW BRANCH

Step 1. Go to top panel VCS

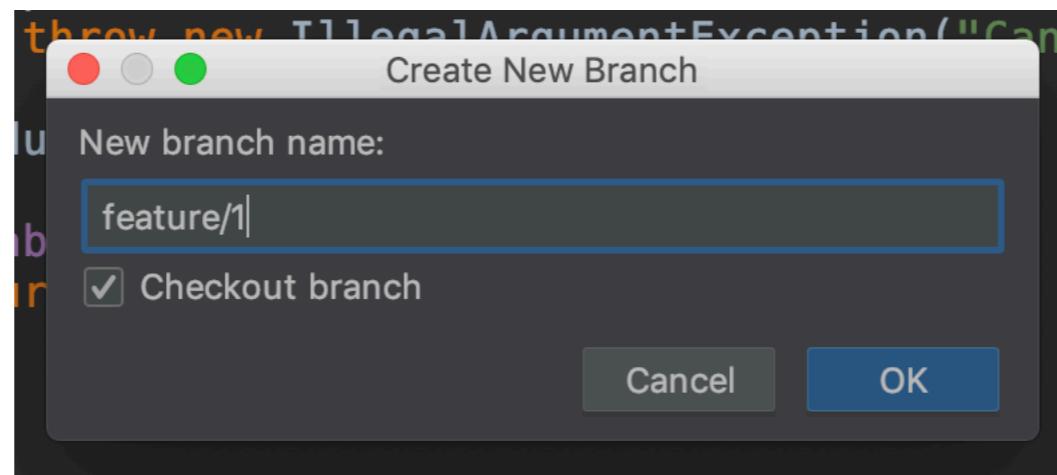


Step 2. Select Git and then Branches...

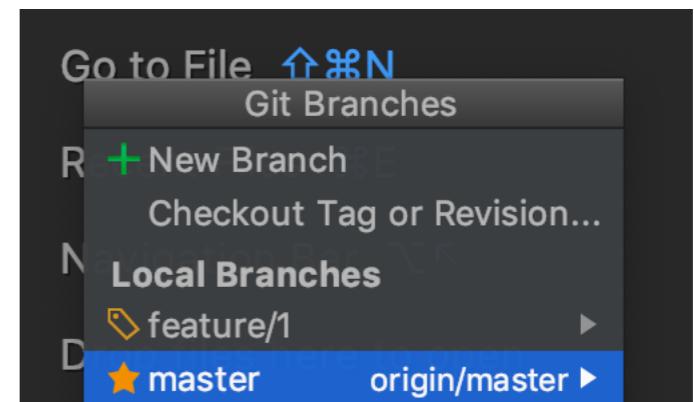
Step 3. Select “New branch” from menu



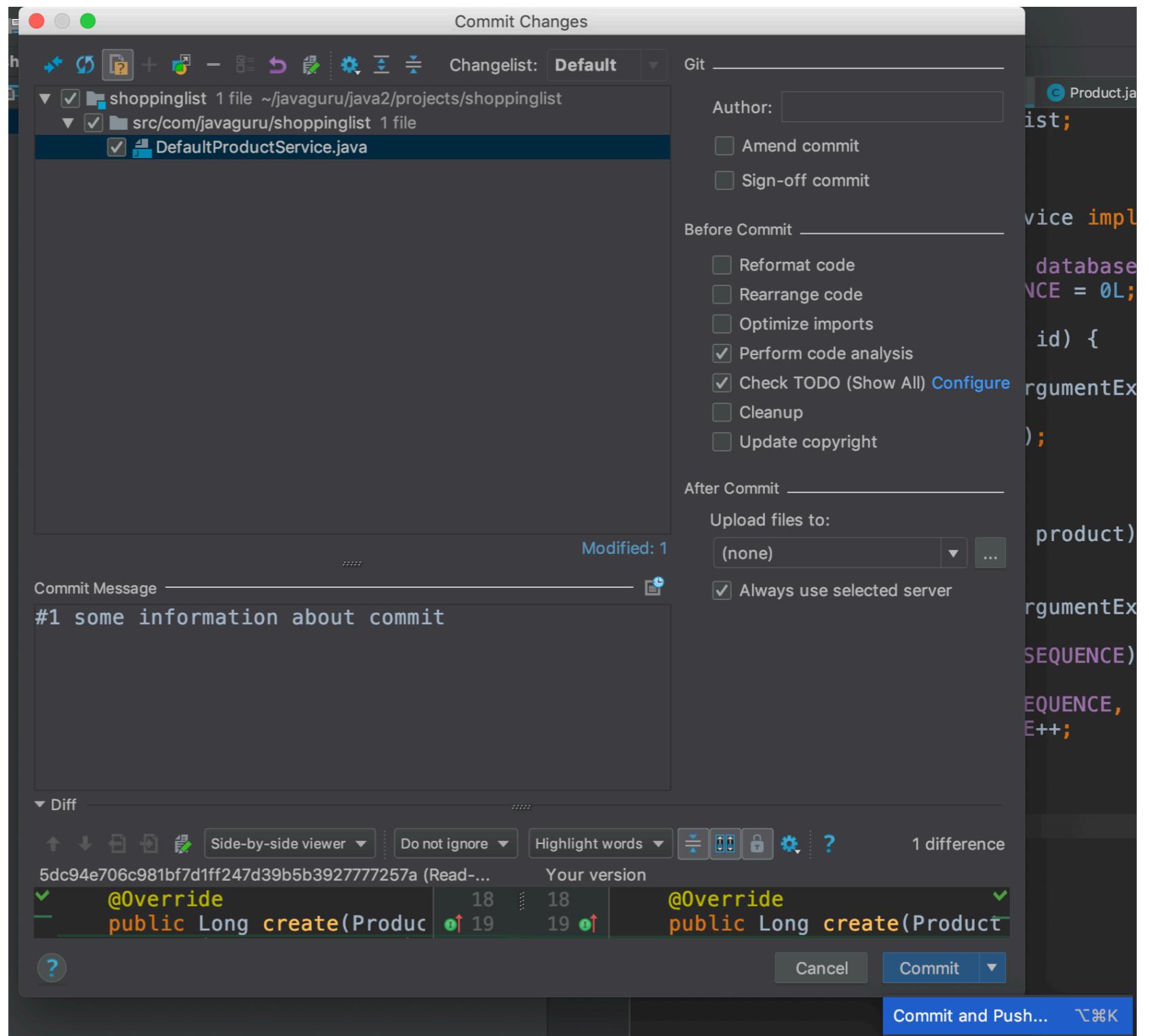
Step 4. Enter new branch name



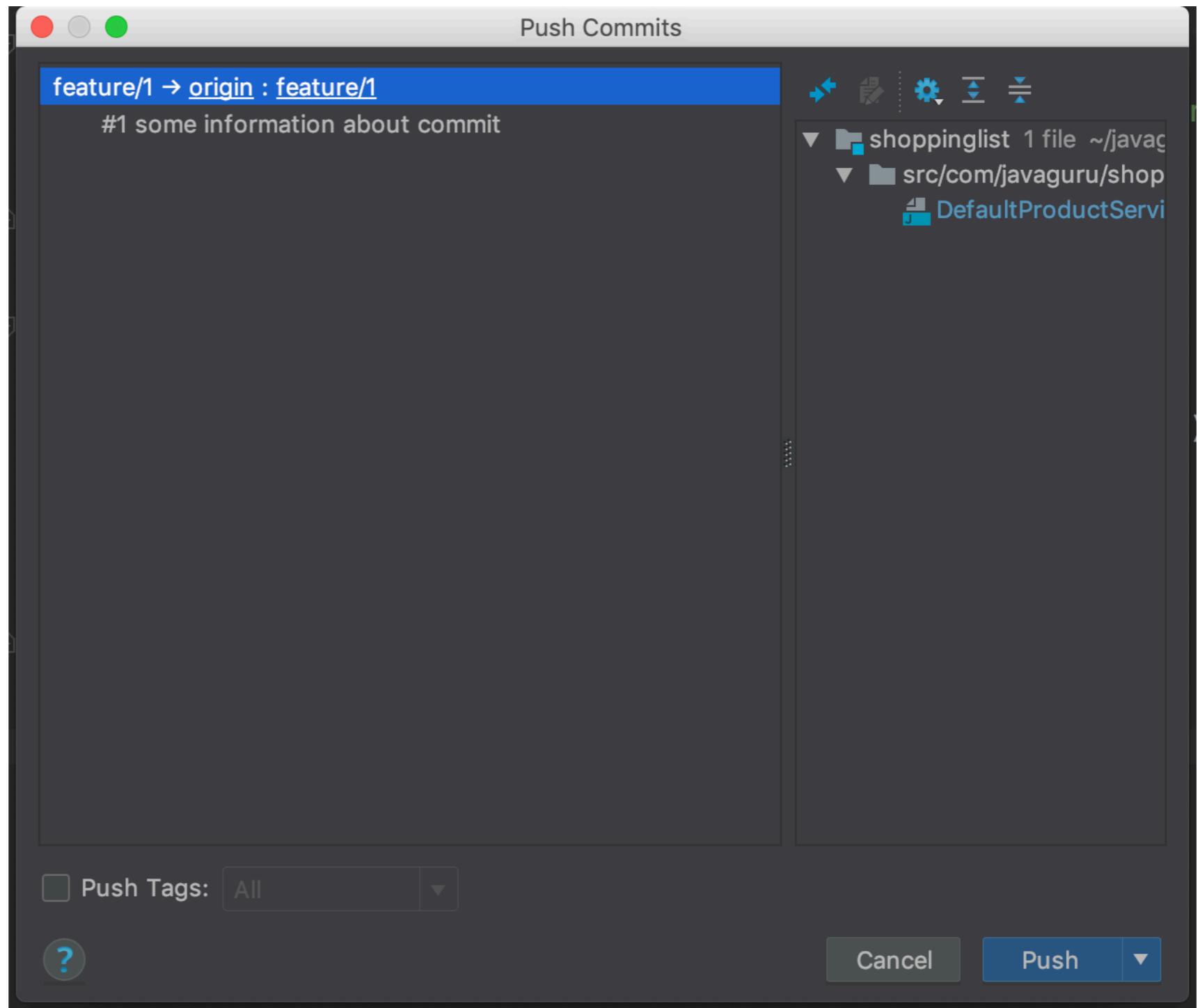
Step 5. Check that your branch created



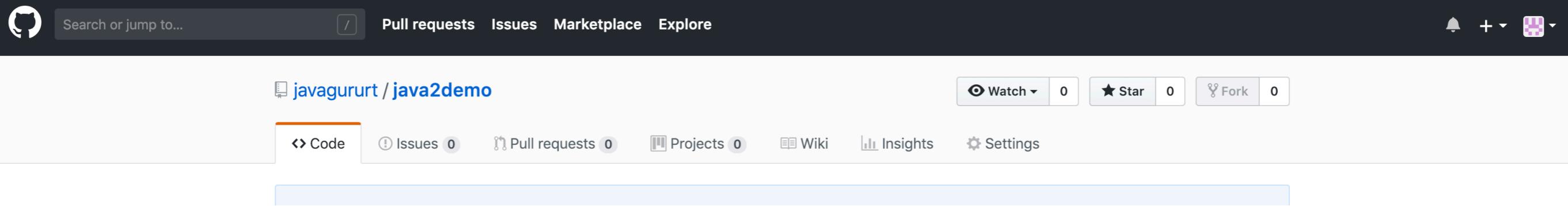
Step 6. After some changes in your branch you need to commit them



Step 7. Check that you push in correct branch



**ADD
COLLABORATOR**



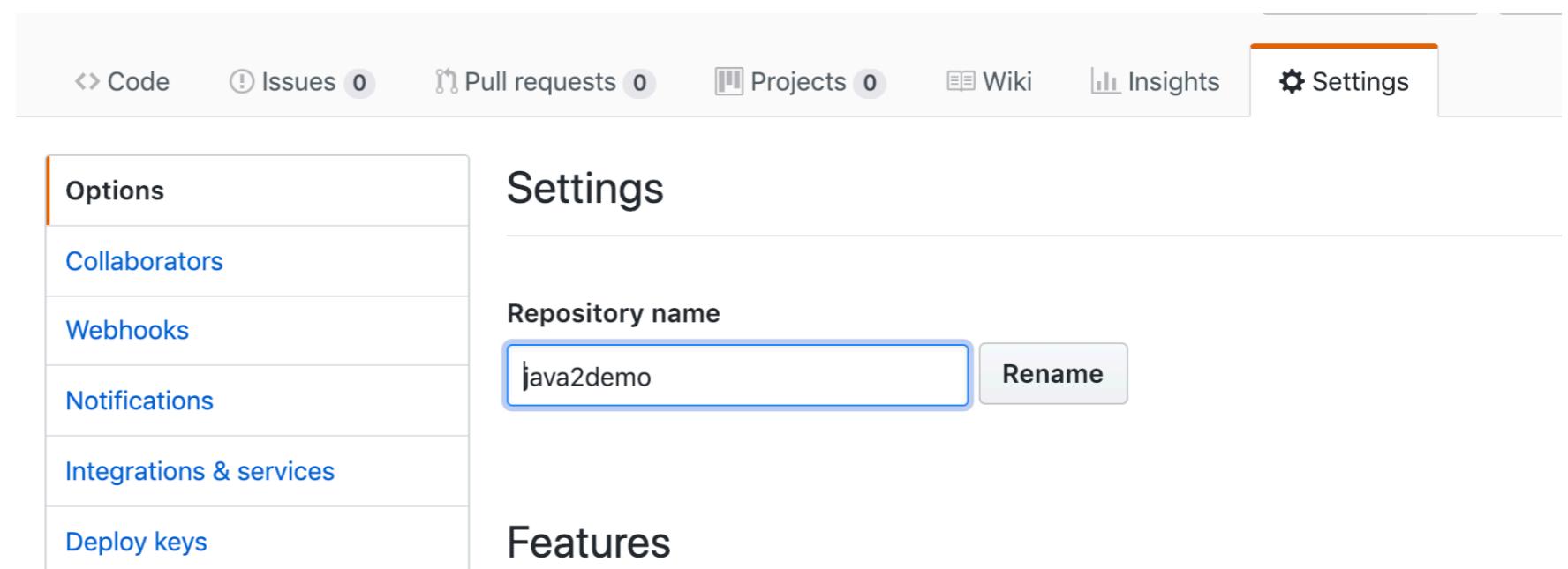
Search or jump to... / Pull requests Issues Marketplace Explore

javagururt / java2demo Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Step 1. Go to your github repository

Step 2. Go to “Settings” and select “Collaborators”



Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Options

- Collaborators
- Webhooks
- Notifications
- Integrations & services
- Deploy keys

Settings

Repository name

java2demo Rename

Features

Step 3. Add collaborator

Collaborators	Push access to the repository
This repository doesn't have any collaborators yet. Use the form below to add a collaborator.	
Search by username, full name or email address You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.	
<input type="text" value="javagururt"/>	<button>Add collaborator</button>
 javagururt	

Step 4. Wait until collaborator will accept your invite

Collaborators	Push access to the repository
 Awaiting javagururt's response	<button>Copy invite link ▾</button> <button>Cancel invite</button>
Search by username, full name or email address You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.	
<input type="text"/>	<button>Add collaborator</button>

Done

Collaborators

Push access to the repository



javagururt



Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator

MERGE/PULL REQUESTS

Step 1. Press “New pull request”

A screenshot of a GitHub repository page. At the top, there are statistics: 5 commits, 5 branches, 0 releases, and 1 contributor. Below these are buttons for creating files, uploading files, finding files, and cloning or downloading the repository. A red arrow points from the text "Step 1. Press ‘New pull request’" to the "New pull request" button, which is highlighted with a red circle. The main area shows a list of commits. The first commit is by user javagururt, merging a pull request from feature/1. The second commit is in the file src/com/javaguru/shoppinglist, fixing a product id sequence. The third commit is in the file .gitignore, marking it as the first commit. At the bottom, there is a call to action to add a README file.

5 commits 5 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download ▾

javagururt Merge pull request [#2](#) from javagururt/feature/1 ... Latest commit d6851a0 an hour ago

src/com/javaguru/shoppinglist product id sequence fix an hour ago

.gitignore first commit 12 days ago

Help people interested in this repository understand your project by adding a README. Add a README

[Code](#)[Issues 0](#)[Pull requests 1](#)[Projects 0](#)[Wiki](#)[Insights](#)[Settings](#)

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

The screenshot shows the GitHub 'Comparing changes' interface. At the top, there are two dropdown menus: 'base: master' and 'compare: master'. A red oval highlights these two fields. Below them is a modal window titled 'Choose a head branch'. Inside the modal, there is a text input field with placeholder text 'Branch, tag, commit, or history marker...' and a blue button labeled 'feature/1'. A second red oval highlights the 'feature/1' button. In the bottom right corner of the main interface, there is a card for 'EXAMPLE COMPARISONS' showing a comparison between 'feature/1' and 'master' made 26 seconds ago.

Step 2. Press "Compare"

Step 3. Choose a branch that will be merged into base (master in this case)

[Code](#)[Issues 0](#)[Pull requests 1](#)[Projects 0](#)[Wiki](#)[Insights](#)[Settings](#)

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](#).



base: master ▾



compare: feature/1 ▾

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



product id sequence fix

Write

Preview

AA B i “ < > ⌂

≡ 1/3 ≡

@ ⬤ ↵

Leave a comment

Step 5. Choose people who will be able to review your changes

Accepted collaborator invitation needed*

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Create pull request

Milestone
No milestone

Assign up to 10 people to this pull request

Filter people

javagururt

ruslanstufetulovs

1 commit

1 file changed

0 commit comments

1 contributor



Commits on Jan 29, 2019



ruslanstufetulovs

product id sequence fix

cc181b7

Step 4. Press "Assignees"

Reviewers
Suggestions
ruslanstufetulovs Request

Assignees



Request



Assign up to 10 people to this pull request

Filter people

javagururt

ruslanstufetulovs



Reviewers
Suggestions
ruslanstufetulovs Request



Assignees



Assign up to 10 people to this pull request

Filter people

javagururt

ruslanstufetulovs



product id sequence fix #2

Edit

 Open javagururt wants to merge 1 commit into master from feature/1

Conversation 0 Commits 1 Checks 0 Files changed 1 +4 -4



javagururt commented just now

Owner



...

No description provided.

 product id sequence fix

cc181b7

 javagururt assigned ruslanstufetulovs just now

Wait until some comments appear

Add more commits by pushing to the feature/1 branch on javagururt/java2_2019_winter.



 Continuous integration has not been set up

Several apps are available to automatically catch bugs and enforce style.

 This branch has no conflicts with the base branch

Merging can be performed automatically.

 Merge pull request

(wait for review)

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

 Write

Preview

AA B i “ <> ↻ ≡ 1/3 ≡ @ ⌂ ↵

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Reviewers



Suggestions

 ruslanstufetulovs

Request

Assignees



 ruslanstufetulovs

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

 Unsubscribe

You're receiving notifications because you authored the thread.

2 participants



 Lock conversation

Conversation 1

Commits 1

Checks 0

Files changed 1

+4 -4



javagururt commented 2 minutes ago

Owner



...

No description provided.

product id sequence fix

cc181b7

javagururt assigned ruslanstufetulovs 2 minutes ago

javagururt reviewed just now

[View changes](#)

src/com/javaguru/shoppinglist/DefaultProductService.java

```
24      24  
25      -      database.put(USER_ID_SEQUENCE, product);  
26      -      return USER_ID_SEQUENCE++;  
25      +      database.put(PRODUCT_ID_SEQUENCE, product);
```



javagururt just now

Owner

my comment

Step 6. Check comments



Reply...

[Resolve conversation](#)

**Step 7. Resolve all conversations
(just commit fixes into same branch)**

Reviewers



Suggestions

ruslanstufetulovs

[Request](#)

Assignees



ruslanstufetulovs

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

[Unsubscribe](#)

You're receiving notifications
because you authored the thread.

2 participants



Add more commits by pushing to the **feature/1** branch on [javagururt/java2_2019_winter](#).



Continuous integration has not been set up

Several apps are available to automatically catch bugs and enforce style.



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view command line instructions.

Step 8. Press “Merge pull request”

Projects



None yet

Milestone



No milestone

Notifications

Unsubscribe

You're receiving notifications because you authored the thread.

2 participants



Add more commits by pushing to the **feature/1** branch on [javagururt/java2_2019_winter](#).



Merge pull request #2 from [javagururt/feature/1](#)

product id sequence fix

Confirm merge

Cancel

Step 9. Press “Confirm merge”

Successfully merged!

product id sequence fix #2

Merged

javagururt merged 1 commit into master from feature/1 just now

Conversation 1

Commits 1

Checks 0

Files changed 1



javagururt commented 3 minutes ago

Owner

+ ...

Reviewers

No reviews

Assignees



No description provided.

product id sequence fix

cc181b7

javagururt assigned ruslanstufetulovs 3 minutes ago



javagururt reviewed a minute ago

[View changes](#)

Labels

None yet

Projects

None yet

javagururt merged commit d6851a0 into master just now

Show resolved

[Revert](#)

Milestone

No milestone

REFERENCES

- ▶ <https://github.com/>
- ▶ <https://www.jetbrains.com/help/idea/set-up-a-git-repository.html>
- ▶ <https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html>
- ▶ <https://www.atlassian.com/git/tutorials>