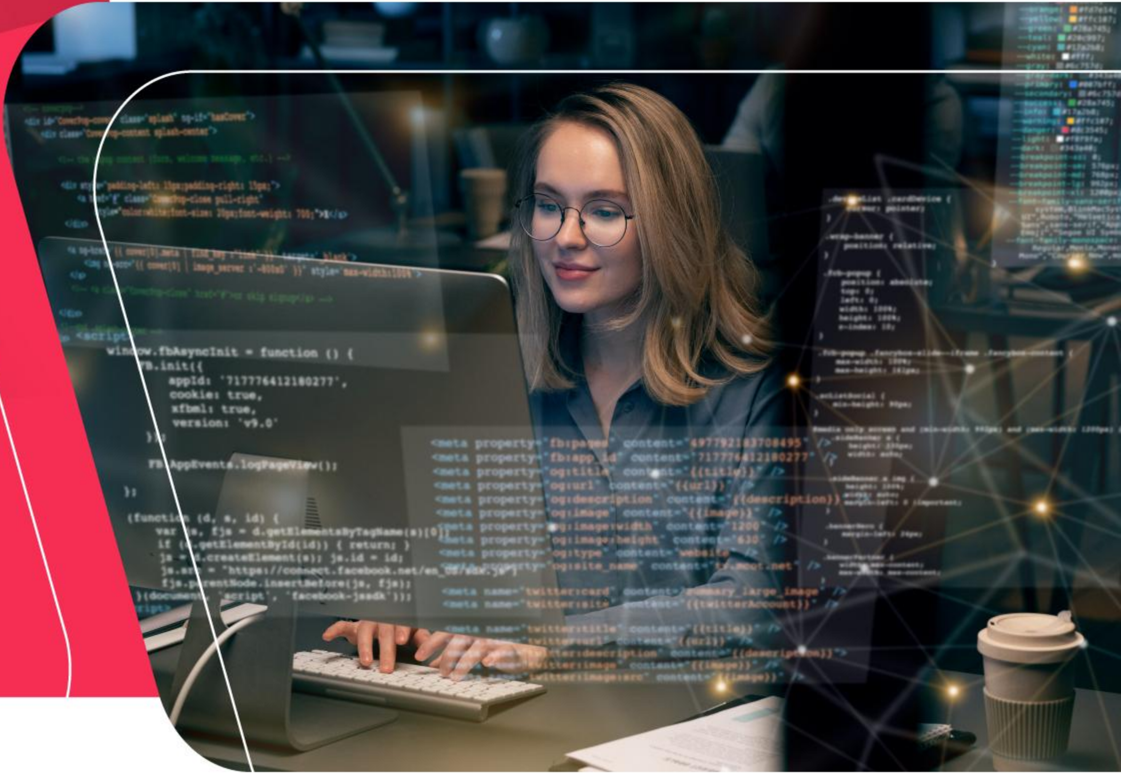
TIA-Academy

Java Bootcamp

Day 24



TIA-Academy

Technologies will be Use

- JDK 8/**11**/15
- JRE 8/**11**/15
- Intellij IDEA Community Edition
- Git

An Intro to Git and GitHub

Install git and create a GitHub account

- The first two things you'll want to do are install git and create a free GitHub account.
- Follow the instructions here to install git (if it's not already installed). Note that for this tutorial **we will be using git on the command line only**. While there are some great git GUIs (graphical user interfaces), I think it's easier to learn git using **git-specific commands** first and then to try out a git GUI once you're more comfortable with the command.
- A note: 95% of other online git resources and discussions will also be for the command-line interface.
- Once you've done that, create a GitHub account here.

Git and GitHub

- A quick aside: git and GitHub are ***not*** the same thing.
- **Git** is an open-source, version control tool created in 2005 by developers working on the Linux operating system; **GitHub** is a company founded in 2008 that makes **tools which integrate with git**.

TIA-Academy

Git and GitHub

git	GitHub
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the **Git** repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

TIA-Academy

Git and GitHub

- You do not need GitHub to use git, but you cannot use GitHub without using git.
- There are many other alternatives to **GitHub**, such as **GitLab**, **BitBucket**, and “host-your-own” solutions such as gogs and gittea.
- All of these are referred to in git-speak as “remotes”, and all are completely optional. You do not need to use a remote to use git, but it will make sharing your code with others easier.

TIA-Academy

Create a local git repository

- When creating a new project on your local machine using git, you'll first create a new repository (or often, 'repo', for short).
- To use git we'll be using the terminal. If you don't have much experience with the terminal and basic commands, check out this tutorial (If you don’t want/ need a short history lesson, skip to step three.)

TIA-Academy

Create a local git repository

- To begin, open up a terminal and move to where you want to place the project on your local machine using the cd (change directory) command.
- For example, if you have a 'projects' folder on your desktop, you'd do something like:

```
mnelson:Desktop mnelson$ cd ~/Desktop
mnelson:Desktop mnelson$ mkdir myproject
mnelson:Desktop mnelson$ cd myproject/
```

Create a local git repository

- To initialize a git repository in the root of the folder, run the **git init** command:

```
mnelson:myproject mnelson$ git init
Initialized empty Git repository in
/Users/mnelson/Desktop/myproject/.git/
```

Add a new file to the repo

- Go ahead and add a new file to the project, using any text editor you like or running a touch command. ‘touch newfile.txt’ just creates and saves a blank file named newfile.txt.
- Once you've added or modified files in a folder containing a git repo, git will notice that the file exists inside the repo. But, git won't track the file unless you explicitly tell it to.

Add a new file to the repo

- Git only saves/manages changes to files that it tracks, so we'll need to send a command to confirm that yes, we want git to track our new file.

```
mnelson:myproject mnelson$ touch mnelson.txt
mnelson:myproject mnelson$ ls
mnelson.txt
```

Add a new file to the repo

- After creating the new file, you can use the git status command to see which files git knows exist.

```
mnelson:myproject mnelson$ git status
On branch master

Initial commit

Untracked files:
   (use "git add <file>..." to include in what will be committed)

 mnelson.txt

nothing added to commit but untracked files present (use "git add" to track)
```

- What this basically says is, "Hey, we noticed you created a new file called mnelson.txt, but unless you use the 'git add' command we aren't going to do anything with it."

TIA-Academy

An interlude: The staging environment, the commit, and you

- One of the most confusing parts when you're first learning git is the concept of the staging environment and how it relates to a commit.
- A commit is a record of what changes you have made since the last time you made a commit. Essentially, you make changes to your repo (for example, adding a file or modifying one) and then tell git to put those changes into a commit.
- Commits make up the essence of your project and allow you to jump to the state of a project at any other commit.

An interlude:
The staging environment, the commit, and you

- One of the most confusing parts when you're first learning git is the concept of the staging environment and how it relates to a commit.
- A commit is a record of what changes you have made since the last time you made a commit. Essentially, you make changes to your repo (for example, adding a file or modifying one) and then tell git to put those changes into a commit.
- Commits make up the essence of your project and allow you to jump to the state of a project at any other commit.

TIA-Academy

An interlude:
The staging environment, the commit, and you

- So, how do you tell git which files to put into a commit? This is where the staging environment or index come in.
- As seen in Step 2, when you make changes to your repo, **git notices that a file has changed but won't do anything with it** (like adding it in a commit).
- To add a file to a commit, you first need to add it to the **staging environment**. To do this, you can use the **git add <filename>** command

An interlude:
The staging environment, the commit, and you

- Once you've used the **git add** command to add all the files you want to the staging environment, you can then tell git to package them into a commit using the **git commit** command.
- Note: The staging environment, also called 'staging', is the new preferred term for this, but you can also see it referred to as the 'index'.

TIA-Academy

Add a file to the staging environment

- Add a file to the staging environment using the git add command.
- If you rerun the git status command, you'll see that git has added the file to the staging environment (notice the "Changes to be committed" line).

```
mnelson:myproject mnelson$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

 new file:   mnelson.txt
```

- To reiterate, **the file has not yet been added to a commit**, but it's about to be.

Create a commit

- It's time to create your first commit!
- Run the command **git commit -m "Your message about the commit"**

```
mnelson:myproject mnelson$ git commit -m "This is my first commit!"
[master (root-commit) b345d9a] This is my first commit!
 1 file changed, 1 insertion(+)
 create mode 100644 mnelson.txt
```

TIA-Academy

Create a commit

- The message at the end of the commit should be something related to what the commit contains - **maybe it's a new feature, maybe it's a bug fix, maybe it's just fixing a typo**.
- **Don't put a message like "asdfadsf" or "foobar"**. That makes the other people who see your commit sad. Very, very, sad.

Create a commit

- Commits live forever in a repository (technically you *can* delete them if you really, really need to but it's messy), so **if you leave a clear explanation** of your changes it can be extremely helpful for future programmers (perhaps future you!) who are trying to **figure out why some change was made years later**.

TIA-Academy

Create a new branch

- Now that you've made a new commit, let's try something a little more advanced.
- Say you want to make a **new feature but are worried about making changes to the main project** while developing the feature. This is where **git branches come in.**
- Branches allow you to move back and forth between 'states' of a project. Official git docs describe branches this way: **'A branch in Git is simply a lightweight movable pointer to one of these commits.'**

TIA-Academy

Create a new branch

- Now that you've made a new commit, let's try something a little more advanced.
- Say you want to make a **new feature but are worried about making changes to the main project** while developing the feature. This is where **git branches come in.**
- Branches allow you to move back and forth between 'states' of a project. Official git docs describe branches this way: **'A branch in Git is simply a lightweight movable pointer to one of these commits.'**

TIA-Academy

Create a new branch

- For instance, if you want **to add a new page to your website** you can **create a new branch just for that page without affecting the main part of the project.**
- **Once you're done** with the page, you can **merge your changes from your branch into the primary branch.**
- When you **create a new branch**, Git keeps track of which **commit your branch 'branched' off of,** so it knows the history behind all the files.

TIA-Academy

Create a new branch

- Let's say you are on the primary branch and want to create a new branch to develop your web page.
- Here's what you'll do: Run **git checkout -b <my branch name>.** This command will automatically create a new branch and then **'check you out' on it, meaning git will move you to that branch**, **off of the primary branch.**

Create a new branch

- After running the above command, you can use the git branch command to confirm that your branch was created:

```
mnelson:myproject mnelson$ git branch
  master
* my-new-branch
```

- The branch name with the **asterisk** next to it indicates which branch you're on at that given time.

A note on branch names

- By default, every git repository’s first branch is named `**master**` (and is typically used as the **primary branch in the project**).
- As part of the tech industry’s general anti-racism work, some groups have begun to use alternate names for the default branch (we are using **“primary”** in this tutorial, for example).
- In other documentation and discussions, you may see “**master**”, or other terms, used to refer to the primary branch.

TIA-Academy

A note on branch names

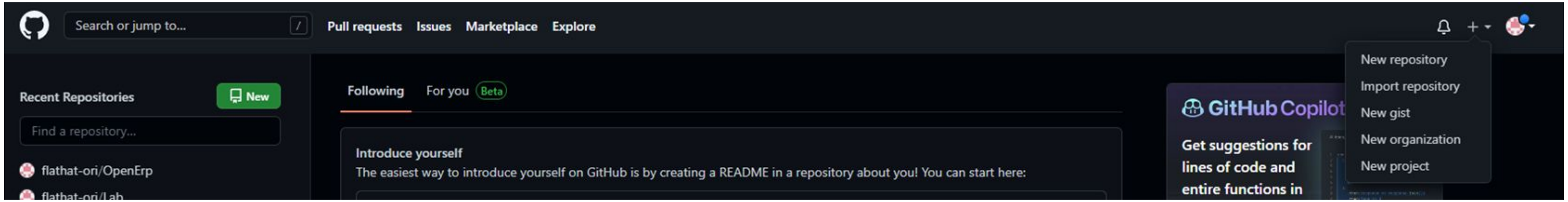
- Now, if you switch back to the primary branch and make some more commits, your new branch won't see any of those changes until you merge those changes onto your new branch.
- To switch to an existing branch (to the master branch for example), you can use git checkout again (without the -b flag) and pass the name of the branch you want to switch to:

```
mnelson:myproject mnelson$ git checkout master
Switched to branch 'master'
mnelson:myproject mnelson$
```

TIA-Academy

Create a new repository on GitHub

- If you only want to keep track of your code locally, you don't need to use GitHub. But if you want to work with a team, you can use GitHub to collaboratively modify the project's code.
- To create a new repo on GitHub, log in and go to the GitHub home page. You can find the “New repository” option under the “+” sign next to your profile picture, in the top right corner of the navbar:

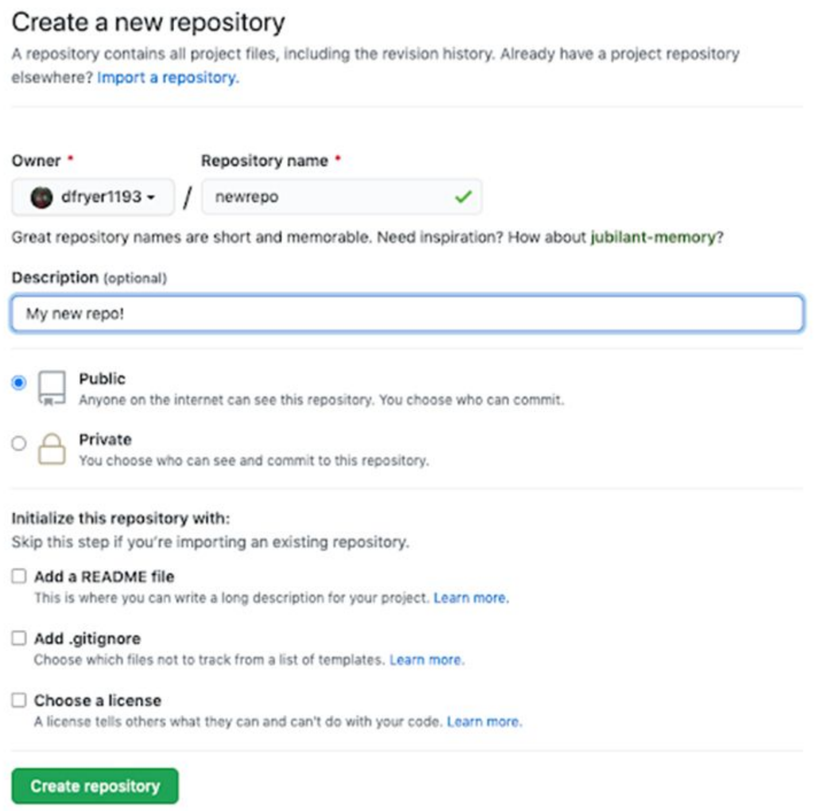


Introduce yourself
The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

TIA-Academy

Create a new repository on GitHub

- After clicking the button, GitHub will ask you to name your repo and provide a brief description.
- When you're done filling out the information, press the 'Create repository' button to make your new repo.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * dfryer1193 / Repository name * newrepo

Great repository names are short and memorable. Need inspiration? How about jubilant-memory?

Description (optional)

My new repo!

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. Learn more.

☐ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

☐ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Create a new repository on GitHub

- GitHub will ask if you want to create a new repo from scratch or if you want to add a repo you have created locally.
- In this case, since we've already created a new repo locally, we want to push that onto GitHub so follow the '....or push an existing repository from the command line' section:

```
mnelson:myproject mnelson$ git remote add origin https://github.com/flathat-ori/newrepo.git
mnelson:myproject mnelson$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 263 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/cubeton/mynewrepository.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Push a branch to GitHub

- Now we'll push the commit in your branch to your new GitHub repo.
- This allows other people to see the changes you've made.
- If they're approved by the repository's owner, the changes can then be merged into the primary branch.

TIA-Academy

Push a branch to GitHub

- To push changes onto a new branch on GitHub, you'll want to run **git push origin yourbranchname.**
- GitHub will automatically create the branch for you on the remote repository:

```
mnelson:myproject mnelson$ git push origin my-new-branch
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 313 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/cubeton/mynewrepository.git
 * [new branch]      my-new-branch -> my-new-branch
```

TIA-Academy

Push a branch to GitHub

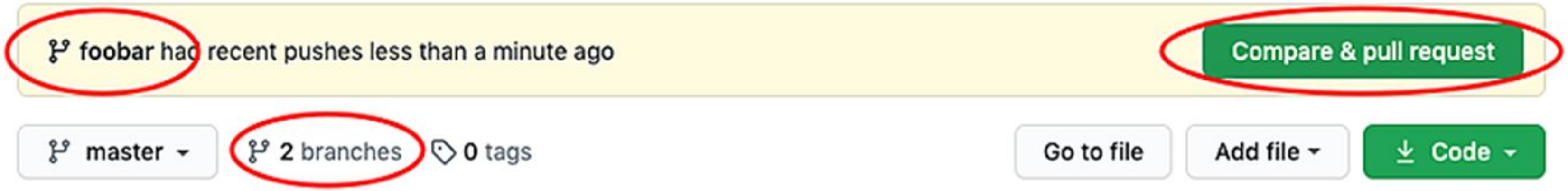
- You might be wondering what that "origin" word means in the command above. What happens is that when you clone a remote repository to your local machine, git creates an alias for you.
- In nearly all cases this alias is called "origin." It's essentially shorthand for the remote repository's URL. So, to push your changes to the remote repository, you could've used either the command: **git push git@github.com:git/git.git yourbranchname** or **git push origin yourbranchname**

(If this is your first time using GitHub locally, it might prompt you to log in with your GitHub username and password.)

TIA-Academy

Create a pull request (PR)

- If you refresh the GitHub page, you'll see note saying a branch with your name has just been pushed into the repository. You can also click the 'branches' link to see your branch listed there.

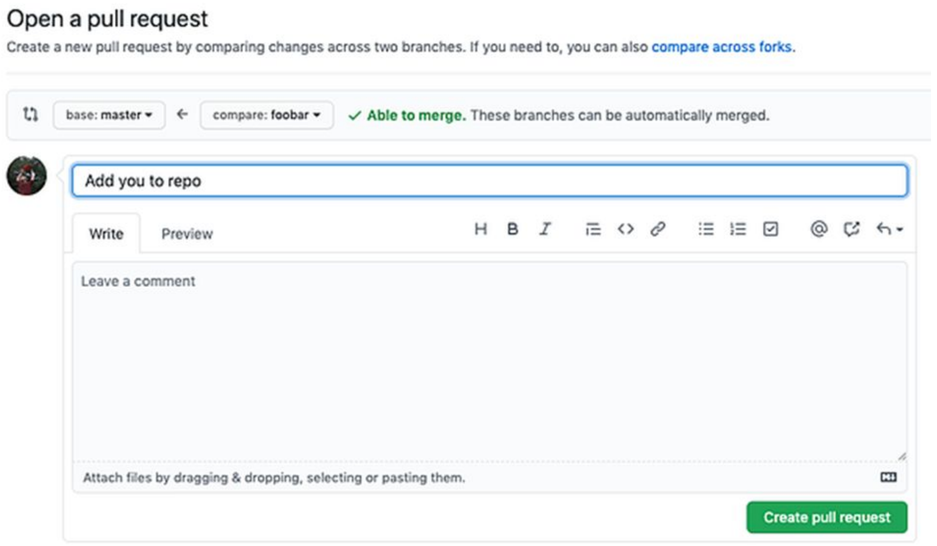

foobar had recent pushes less than a minute ago

- Now click the green button in the screenshot above. We're going to make a pull request!

TIA-Academy

Create a pull request (PR)

- A pull request (or PR) is a way to alert a repo's owners that you want to make some changes to their code. It allows them to review the code and make sure it looks good before putting your changes on the primary branch.
- This is what the PR page looks like before you've submitted it:



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.

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

Add you to repo

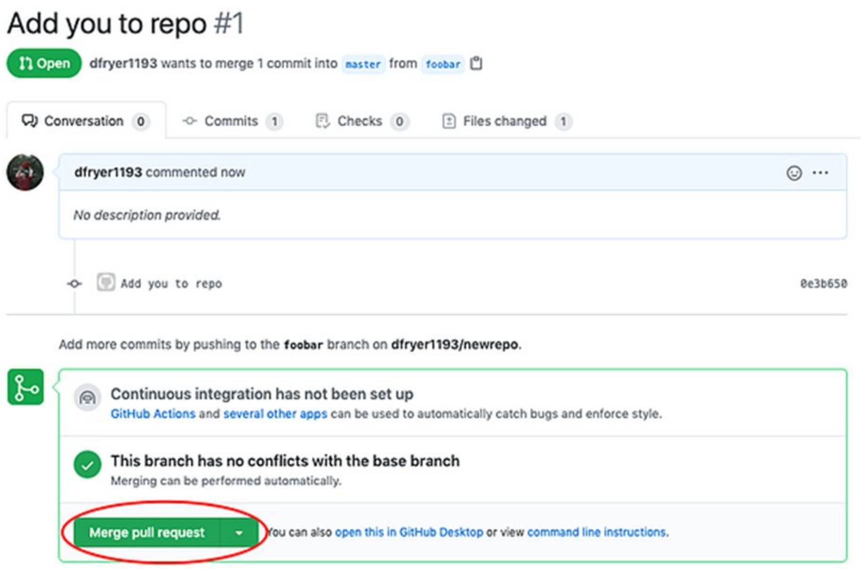
Leave a comment

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

TIA-Academy

Create a pull request (PR)

- And this is what it looks like once you've submitted the PR request:
- You might see a big green button at the bottom that says 'Merge pull request'. Clicking this means you'll merge your changes into the primary branch..


Add you to repo #1

Open dfryer1193 wants to merge 1 commit into master from foobar

dfryer1193 commented now

No description provided.

Add you to repo 0e3b650

Add more commits by pushing to the foobar branch on dfryer1193/newrepo.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

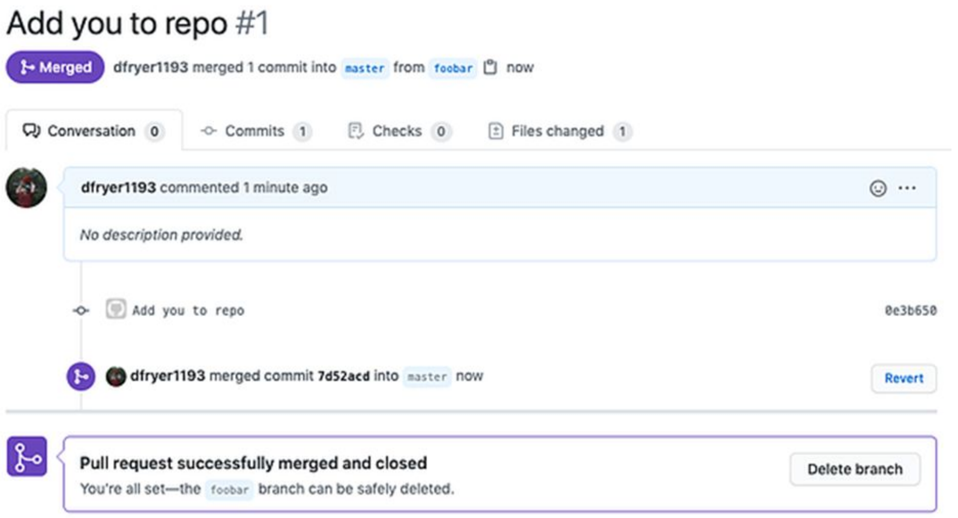
This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

TIA-Academy

Merge a PR

- Go ahead and click the green 'Merge pull request' button. This will merge your changes into the primary branch.
- When you're done, I recommend deleting your branch (too many branches can become messy), so hit that grey 'Delete branch' button as well.


Add you to repo #1

Merged dfryer1193 merged 1 commit into master from foobar now

dfryer1193 commented 1 minute ago

No description provided.

Add you to repo 0e3b650

dfryer1193 merged commit 7d52acd into master now

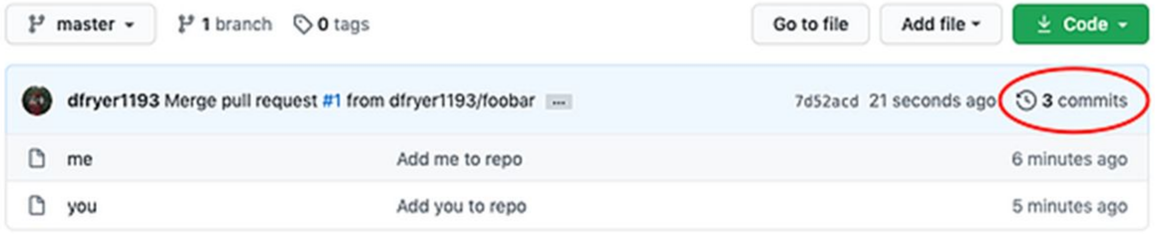
Pull request successfully merged and closed

You're all set—the foobar branch can be safely deleted.

TIA-Academy

Merge a PR

- You can double check that your commits were merged by clicking on the 'Commits' link on the first page of your new repo.

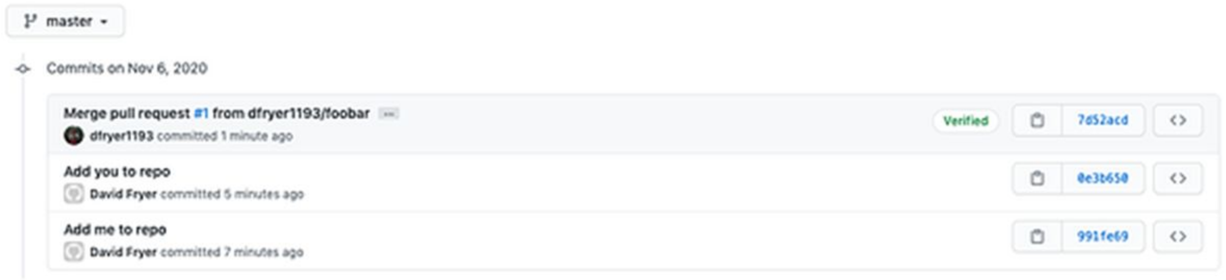

master 1 branch 0 tags Go to file Add file Code

dfryer1193 Merge pull request #1 from dfryer1193/foobar ... 7d52acd 21 seconds ago 3 commits

me	Add me to repo	6 minutes ago
you	Add you to repo	5 minutes ago

Merge a PR

- This will show you a list of all the commits in that branch. You can see the one I just merged right up top (Merge pull request #1).



master

Commits on Nov 6, 2020

Merge pull request #1 from dfryer1193/foobar
dfryer1193 committed 1 minute ago — Verified — 7d52acd

Add you to repo
David Fryer committed 5 minutes ago — 0e3b650

Add me to repo
David Fryer committed 7 minutes ago — 991fe69

Merge a PR

- You can also see the hash code of the commit on the right hand side. A hash code is a unique identifier for that specific commit.
- It's useful for referring to specific commits and when undoing changes (use the git revert <hash code number> command to backtrack).

TIA-Academy

Get changes on GitHub back to your computer

- Right now, the repo on GitHub looks a little different than what you have on your local machine.
- For example, the commit you made in your branch and merged into the primary branch doesn't exist in the primary branch on your local machine.
- In order to get the most recent changes that you or others have merged on GitHub, use the **git pull origin master** command **(when working on the primary branch).**

Get changes on GitHub back to your computer

- In most cases, this can be shortened to “git pull”.

```
mnelson:myproject mnelson$ git pull origin master
remote: Counting objects: 1, done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
From https://github.com/cubeton/mynewrepository
 * branch            master     -> FETCH_HEAD
   b345d9a..5381b7c  master     -> origin/master
Merge made by the 'recursive' strategy.
 mnelson.txt | 1 +
 1 file changed, 1 insertion(+)
```

- This shows you all the files that have changed and how they've changed.

TIA-Academy

Get changes on GitHub back to your computer

- Now we can use the git log command again to see all new commits.
- (You may need to switch branches back to the primary branch. You can do that using the git checkout master command.)

```
mnelson:myproject mnelson$ git log
commit 3e270876db0e5ffd3e9bfc5edede89b64b83812c
Merge: 4f1cb17 5381b7c
Author: Meghan Nelson <mnelson@hubspot.com>
Date:   Fri Sep 11 17:48:11 2015 -0400

    Merge branch 'master' of https://github.com/cubeton/mynewrepository

commit 4f1cb1798b6e6890da797f98383e6337df577c2a
Author: Meghan Nelson <mnelson@hubspot.com>
Date:   Fri Sep 11 17:48:00 2015 -0400

    added a new file

………………………………….

    Added some more text to my file

commit b345d9a25353037afdeaa9fcaf9f330effd157f1
Author: Meghan Nelson <mnelson@hubspot.com>
Date:   Thu Sep 10 17:42:15 2015 -0400

    This is my first commit!
```

TIA-Academy

ASSIGNMENT 01

TIA-Academy

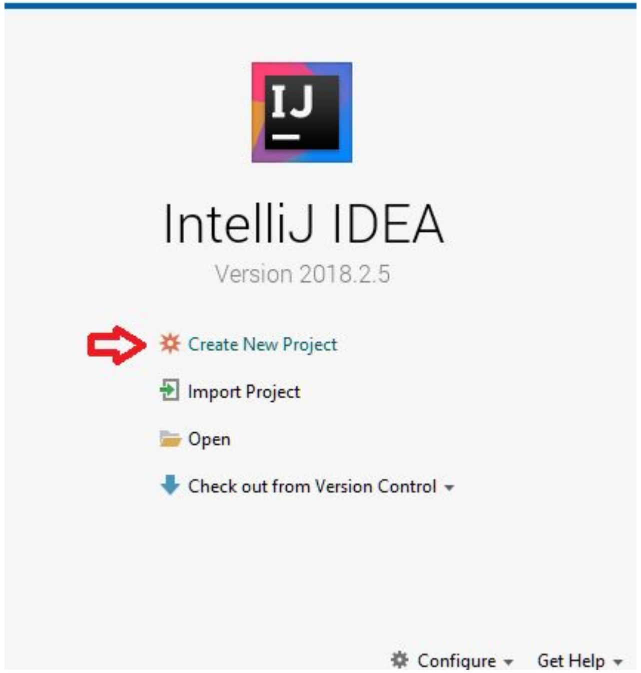
Git - Using Git in Intellij IDE

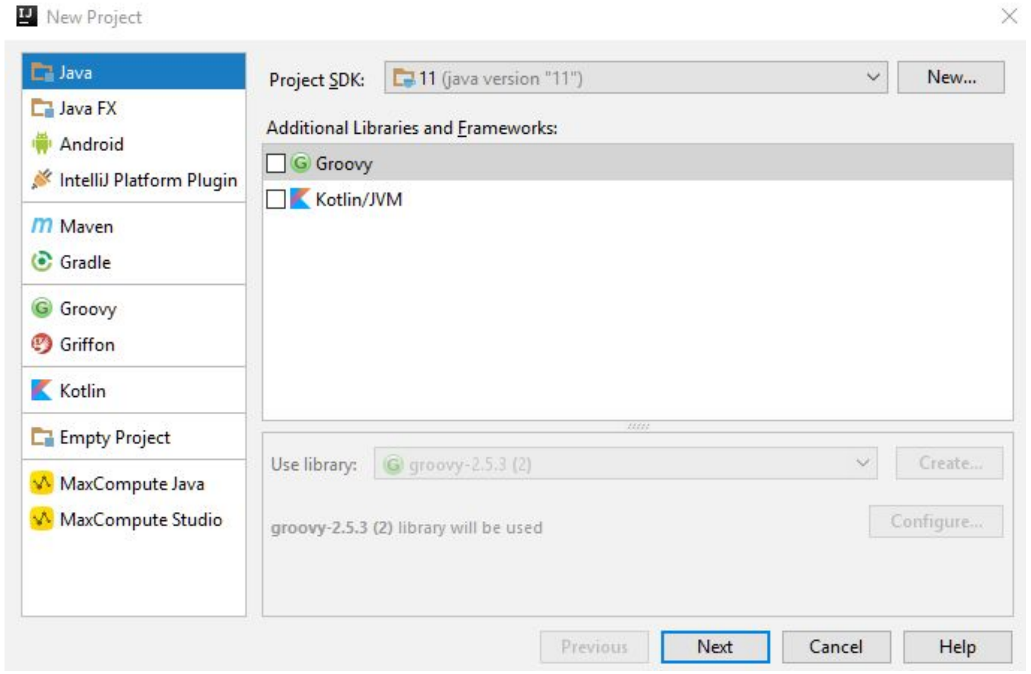
Git - Using Git in Intellij IDE

- Intellij (Ultimate and Community versions) have built-in support for Git. This tutorial will show how to configure and use Git for a Java project.

TIA-Academy

Creating a Project





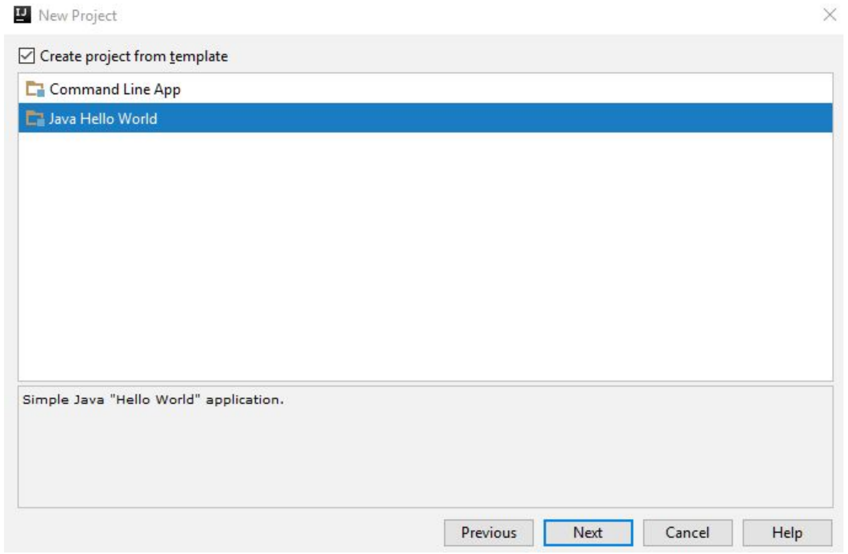
Project SDK: 11 (java version "11")

Additional Libraries and Frameworks:
- [ ] Groovy
- [ ] Kotlin/JVM

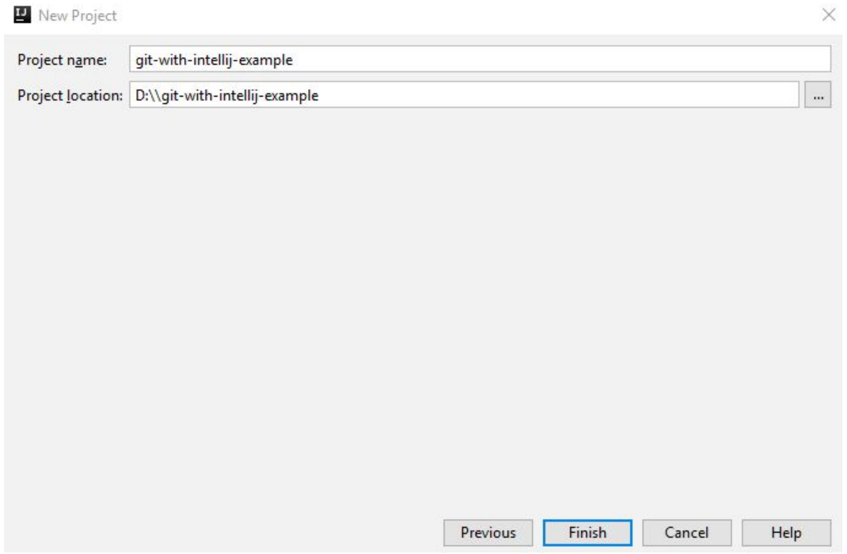
Use library: groovy-2.5.3 (2)

groovy-2.5.3 (2) library will be used

Creating a Project



Simple Java "Hello World" application.



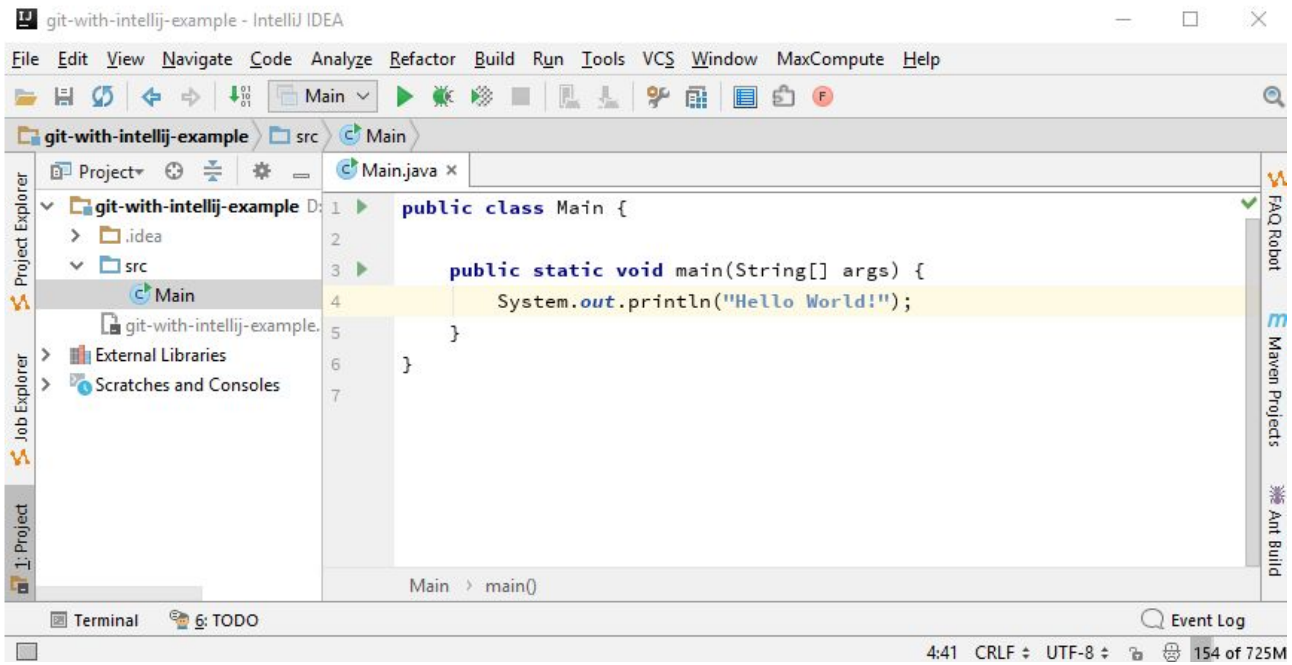
Project name: git-with-intellij-example

Project location: D:\\git-with-intellij-example

TIA-Academy

Creating a Project

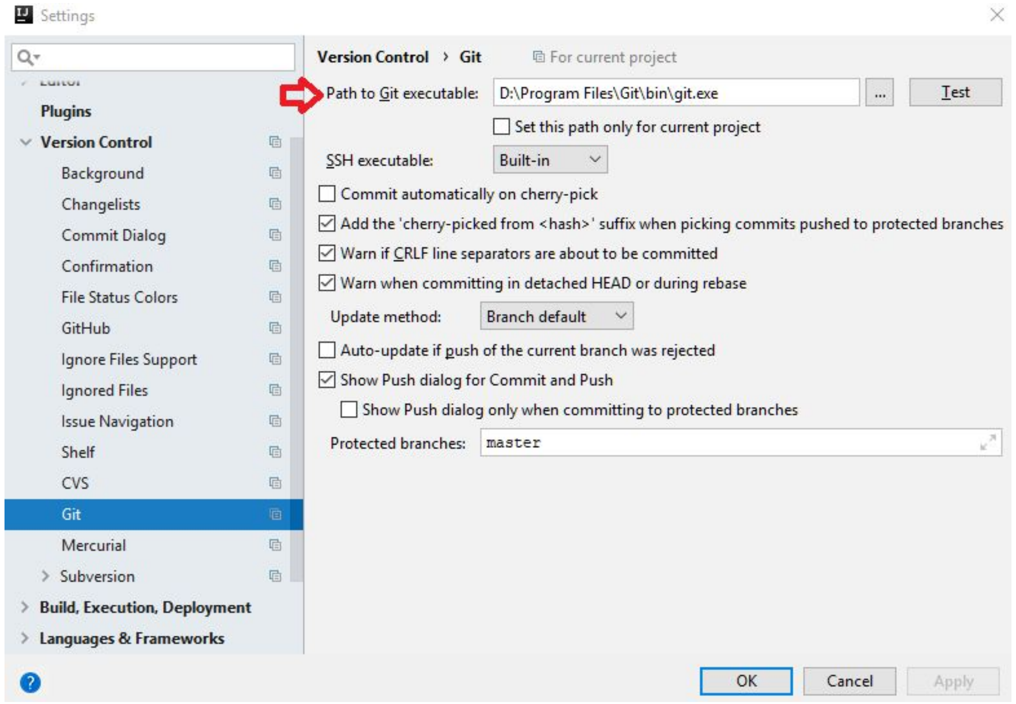
- The project with a Java class is created:



```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Configuring Git

- Since Intellij does not come with Git distribution, we need to install it.
- Open Settings>Version Control>Git (Ctrl+Alt+S) and specify git external path as shown (this is just a one time configuration):


Settings

Version Control › Git For current project

Path to Git executable: D:\Program Files\Git\bin\git.exe

☐ Set this path only for current project

SSH executable: Built-in

☐ Commit automatically on cherry-pick
☑ Add the 'cherry-picked from <hash>' suffix when picking commits pushed to protected branches
☑ Warn if CRLF line separators are about to be committed
☑ Warn when committing in detached HEAD or during rebase

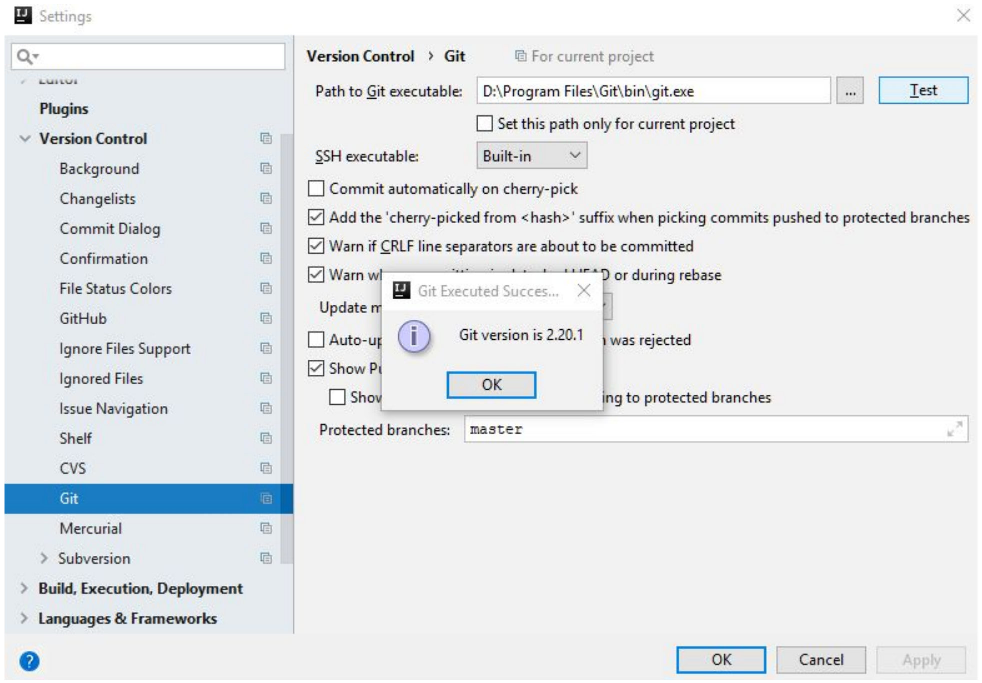
Update method: Branch default

☐ Auto-update if push of the current branch was rejected
☑ Show Push dialog for Commit and Push
☐ Show Push dialog only when committing to protected branches

Protected branches: master

Configuring Git

- Click on the Test button:



Settings

Version Control › Git   For current project

Path to Git executable: D:\Program Files\Git\bin\git.exe Test

☐ Set this path only for current project

SSH executable: Built-in

☐ Commit automatically on cherry-pick

☑ Add the 'cherry-picked from <hash>' suffix when picking commits pushed to protected branches

☑ Warn if CRLF line separators are about to be committed

☑ Warn wh… …D or during rebase

Update m…

☐ Auto-up… …was rejected

☑ Show Pu…

☐ Show… …ing to protected branches

Protected branches: master

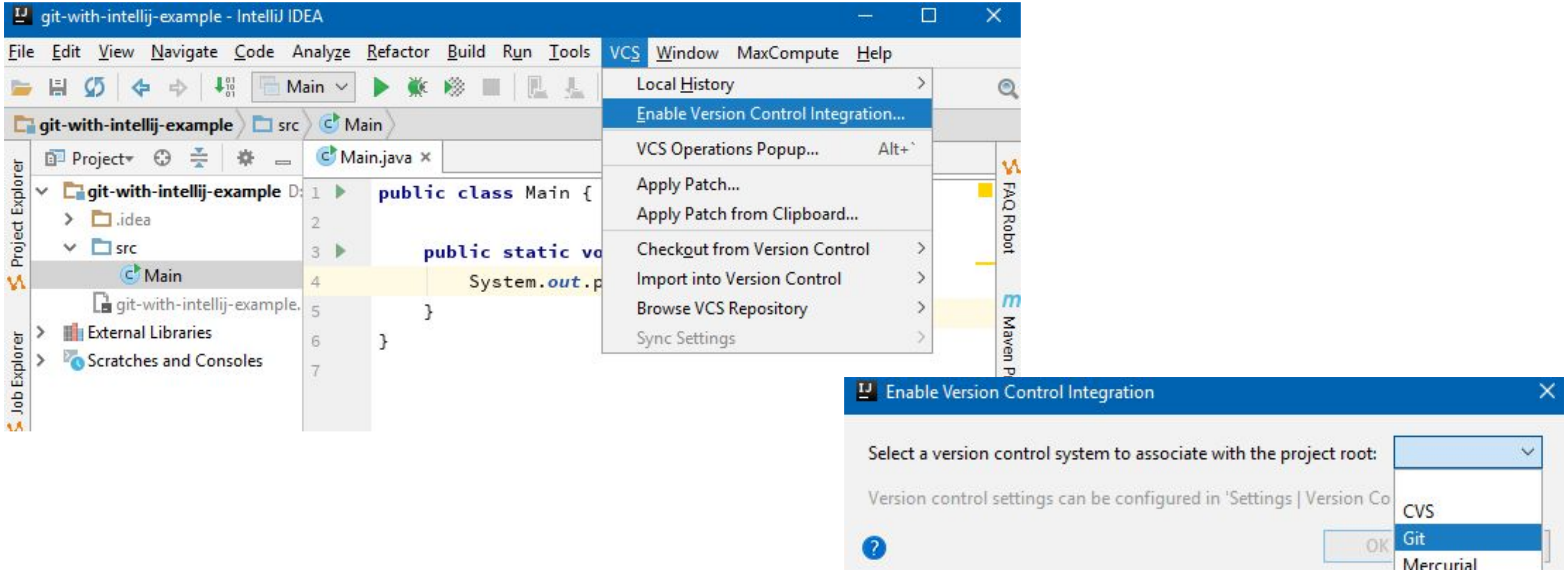
Git Executed Succes...

Git version is 2.20.1

OK

Creating Git repository

- Go to VCS>Enable Version Control Integration.., it will ask us to select the version control system and then the project where .git folder will be created:


git-with-intellij-example - IntelliJ IDEA
File
Edit
View
Navigate
Code
Analyze
Refactor
Build
Run
Tools
VCS
Window
MaxCompute
Help
Main
git-with-intellij-example
src
Main
Project
Main.java
git-with-intellij-example
.idea
src
Main
git-with-intellij-example.
External Libraries
Scratches and Consoles
Project Explorer
Job Explorer
FAQ Robot
Maven P
Local History
Enable Version Control Integration...
VCS Operations Popup... Alt+`
Apply Patch...
Apply Patch from Clipboard...
Checkout from Version Control
Import into Version Control
Browse VCS Repository
Sync Settings
Enable Version Control Integration
CVS
Git
Mercurial
OK

```
public class Main {

    public static vo
        System.out.p
    }
}
```

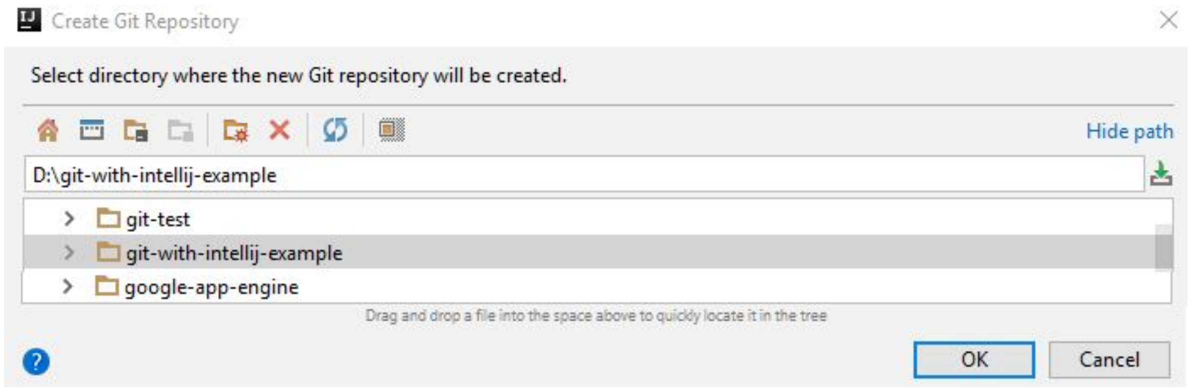
Select a version control system to associate with the project root:

Version control settings can be configured in 'Settings | Version Co

TIA-Academy

Creating Git repository

- Selecting our project:


Create Git Repository

Select directory where the new Git repository will be created.

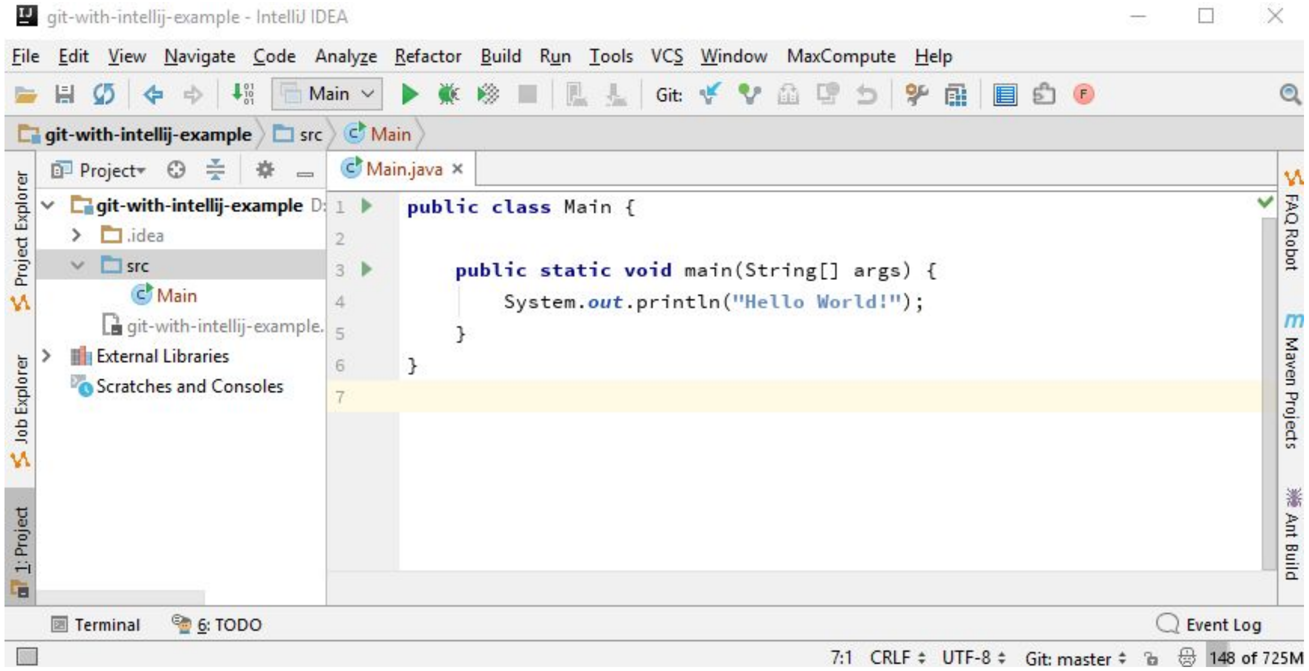
D:\git-with-intellij-example

Drag and drop a file into the space above to quickly locate it in the tree

TIA-Academy

Creating Git repository

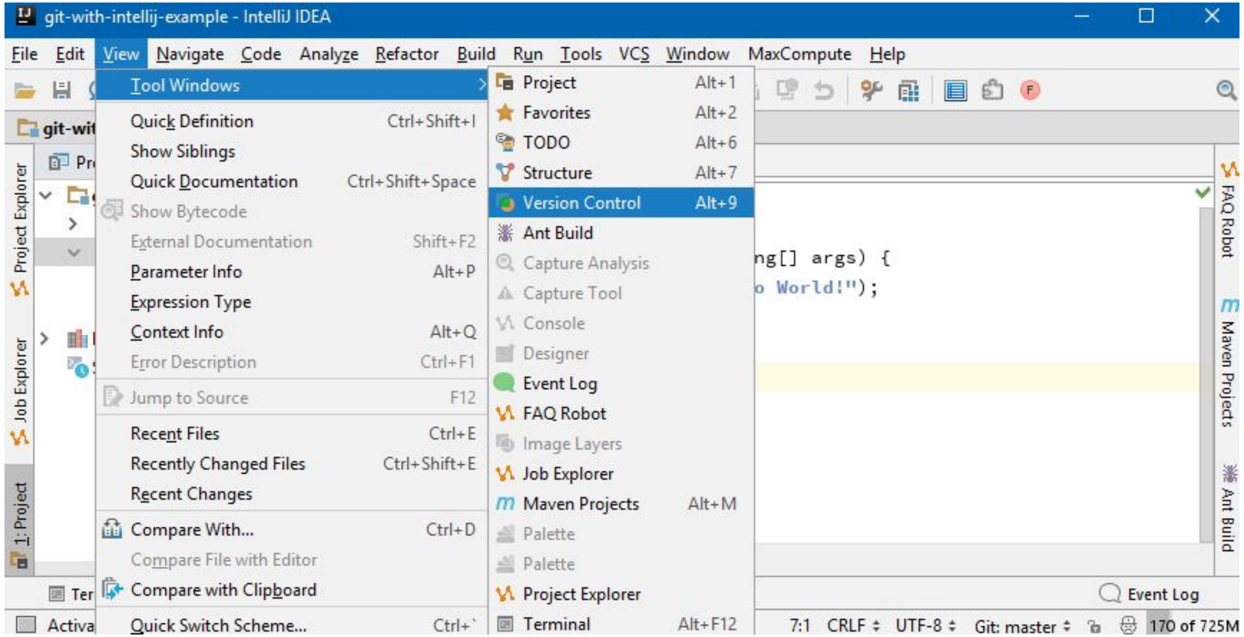
- Now the file colors in the project changed to red (red is used for untracked files):


git-with-intellij-example - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window MaxCompute Help
Main
Git:
git-with-intellij-example
src
Main
Project
Main.java
git-with-intellij-example D:
.idea
src
Main
git-with-intellij-example.
External Libraries
Scratches and Consoles
Project Explorer
Job Explorer
1: Project
FAQ Robot
Maven Projects
Ant Build
Terminal
6: TODO
Event Log
7:1 CRLF UTF-8 Git: master 148 of 725M

```
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

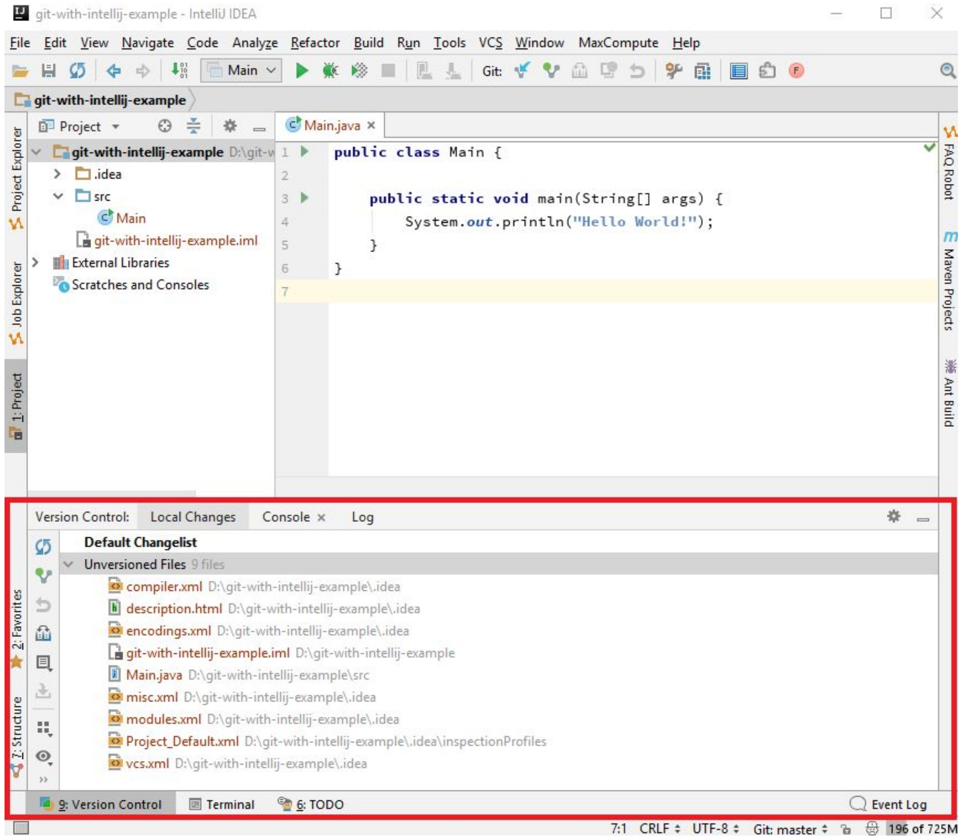
Version Control view

- Open View>Tool Windows>Version Control(Alt+9):



Version Control view

- As seen above 'Local Changes' tab shows all untracked files in red color.
- Let's ignore Intellij specific files.



git-with-intellij-example - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window MaxCompute Help
Main
Git:
git-with-intellij-example
Project
git-with-intellij-example D:\git-w
.idea
src
Main
git-with-intellij-example.iml
External Libraries
Scratches and Consoles
Main.java
Version Control: Local Changes Console Log
9: Version Control Terminal 6: TODO Event Log
7:1 CRLF UTF-8 Git: master 196 of 725M

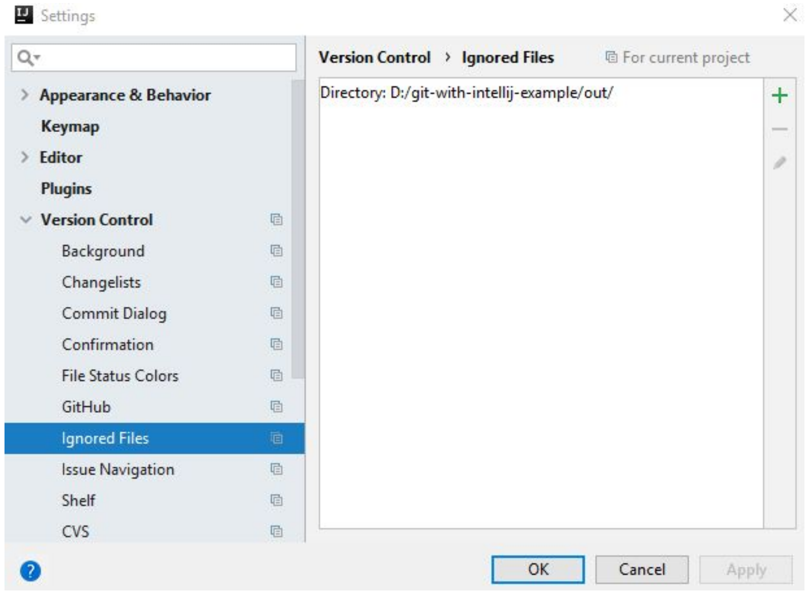
```
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Default Changelist
Unversioned Files 9 files
- compiler.xml D:\git-with-intellij-example\.idea
- description.html D:\git-with-intellij-example\.idea
- encodings.xml D:\git-with-intellij-example\.idea
- git-with-intellij-example.iml D:\git-with-intellij-example
- Main.java D:\git-with-intellij-example\src
- misc.xml D:\git-with-intellij-example\.idea
- modules.xml D:\git-with-intellij-example\.idea
- Project_Default.xml D:\git-with-intellij-example\.idea\inspectionProfiles
- vcs.xml D:\git-with-intellij-example\.idea

TIA-Academy

Ignoring Files

- Intellij has a setting to ignore files:

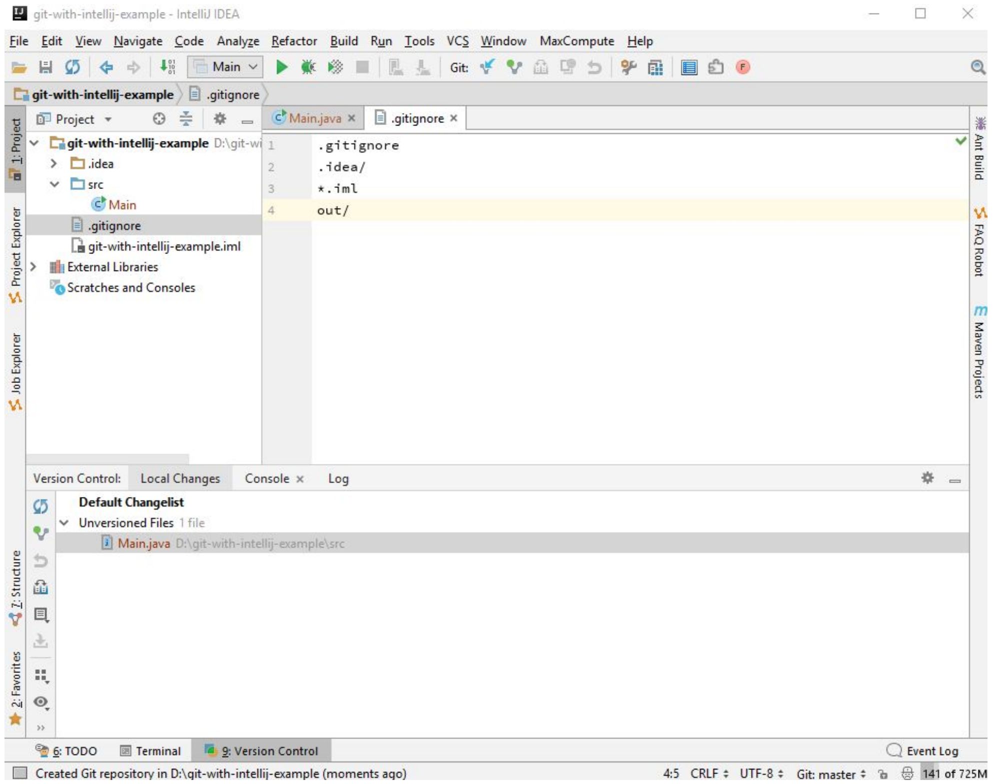


Directory: D:/git-with-intellij-example/out/

- Above view does not create .gitignore file, but it maintains an internal Intellij file for ignoring artifacts. We will not use this view and will create .gitignore file manually at the project's root.

Ignoring Files

- As seen in 'Local Changes' tab, all Intellij specific files disappeared and their color in the 'Project' tree view also turned to normal color.
- A plugin called .ignore, can be used which has a various useful functionality for creating/editing .gitignore files.



git-with-intellij-example - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window MaxCompute Help
Main
Git:
git-with-intellij-example
.gitignore
Project
Main.java
.gitignore
git-with-intellij-example D:\git-wi
.idea
src
Main
.gitignore
git-with-intellij-example.iml
External Libraries
Scratches and Consoles
Version Control:
Local Changes
Console
Log
6: TODO
Terminal
9: Version Control
Event Log
4:5 CRLF UTF-8 Git: master 141 of 725M

```
.gitignore
.idea/
*.iml
out/
```

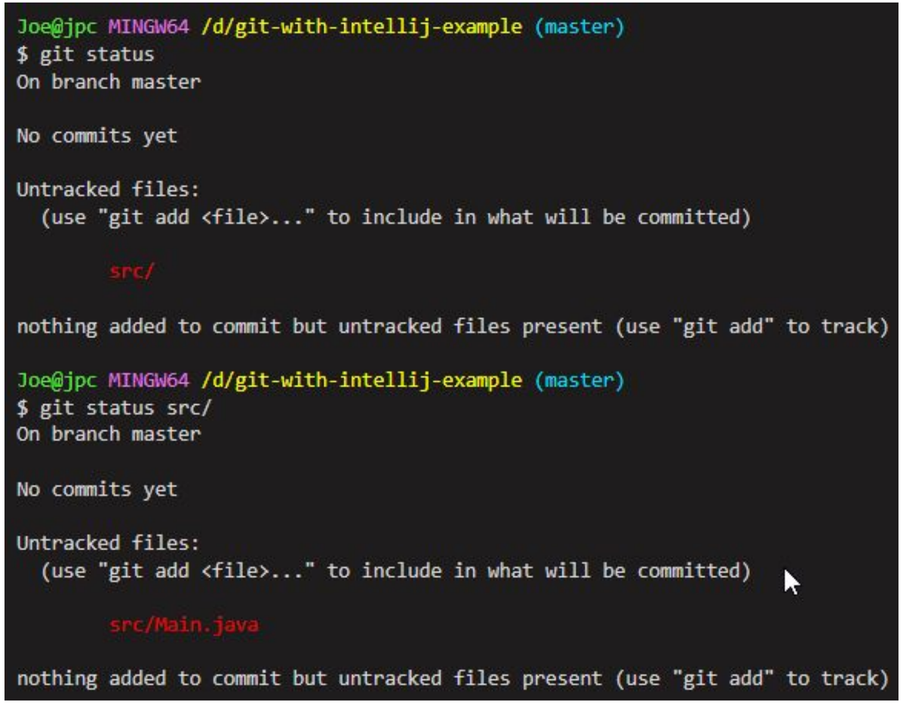
Default Changelist
Unversioned Files 1 file
Main.java D:\git-with-intellij-example\src

Created Git repository in D:\git-with-intellij-example (moments ago)

TIA-Academy

File status

- In Intellij, each file has its own status marked with a specific color, check out this for color-to-status listing. The above color (red) shows that Main.java file is unversioned (untracked). Let's confirm that from git-bash:

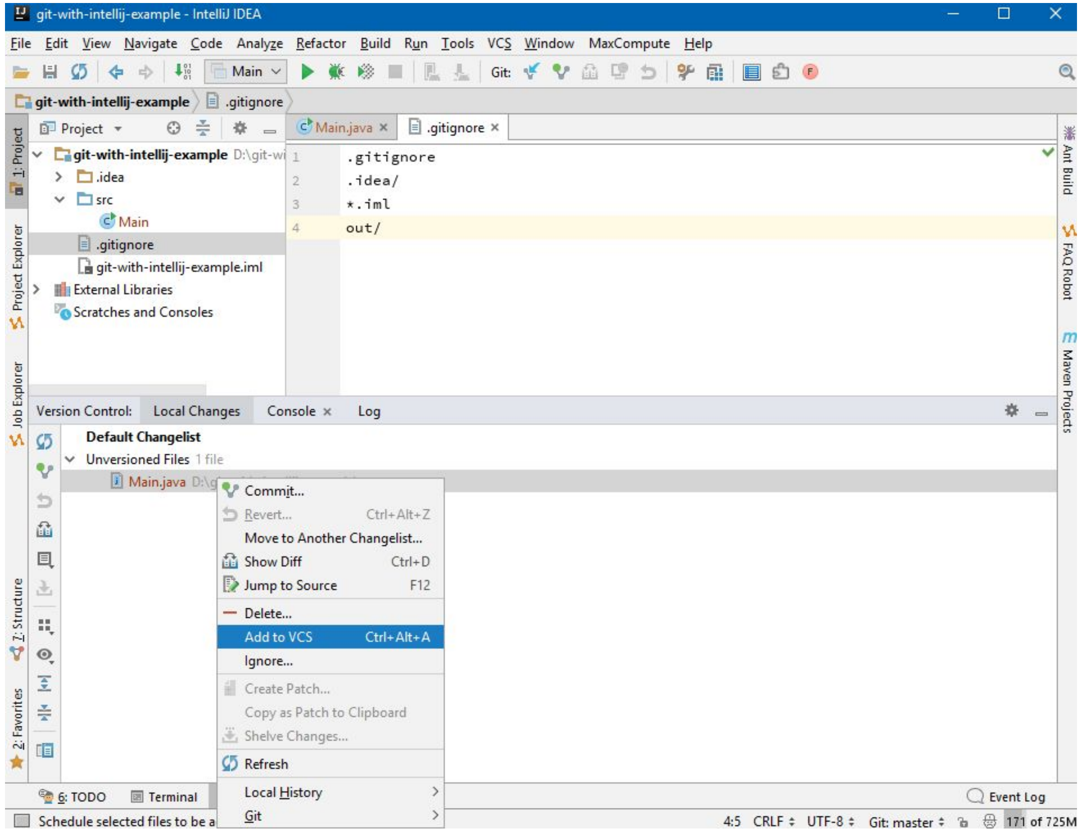


```
Joe@jpc MINGW64 /d/git-with-intellij-example (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        src/

nothing added to commit but untracked files present (use "git add" to track)

Joe@jpc MINGW64 /d/git-with-intellij-example (master)
$ git status src/
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        src/Main.java

nothing added to commit but untracked files present (use "git add" to track)
```

TIA-Academy

Adding files to staging

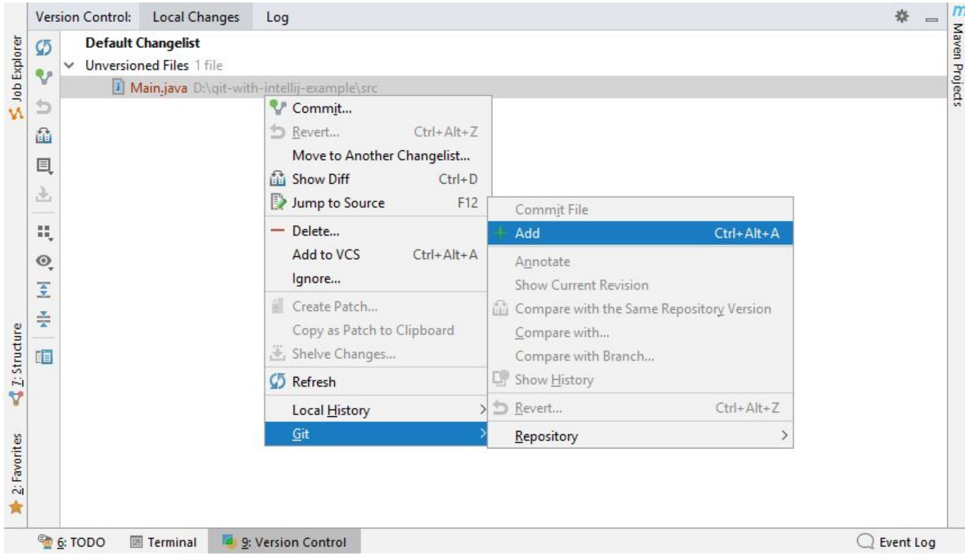
- Right click on the selected files to add them to staging as shown:


git-with-intellij-example - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window MaxCompute Help
git-with-intellij-example
.gitignore
Project
Main.java
.idea
src
Main
git-with-intellij-example.iml
External Libraries
Scratches and Consoles
Version Control: Local Changes Console Log
Default Changelist
Unversioned Files 1 file
Commit...
Revert... Ctrl+Alt+Z
Move to Another Changelist...
Show Diff Ctrl+D
Jump to Source F12
Delete...
Add to VCS Ctrl+Alt+A
Ignore...
Create Patch...
Copy as Patch to Clipboard
Shelve Changes...
Refresh
Local History
Git
6: TODO
Terminal
Event Log
Schedule selected files to be a
4:5 CRLF UTF-8 Git: master 171 of 725M

```
.gitignore
.idea/
*.iml
out/
```

Adding files to staging

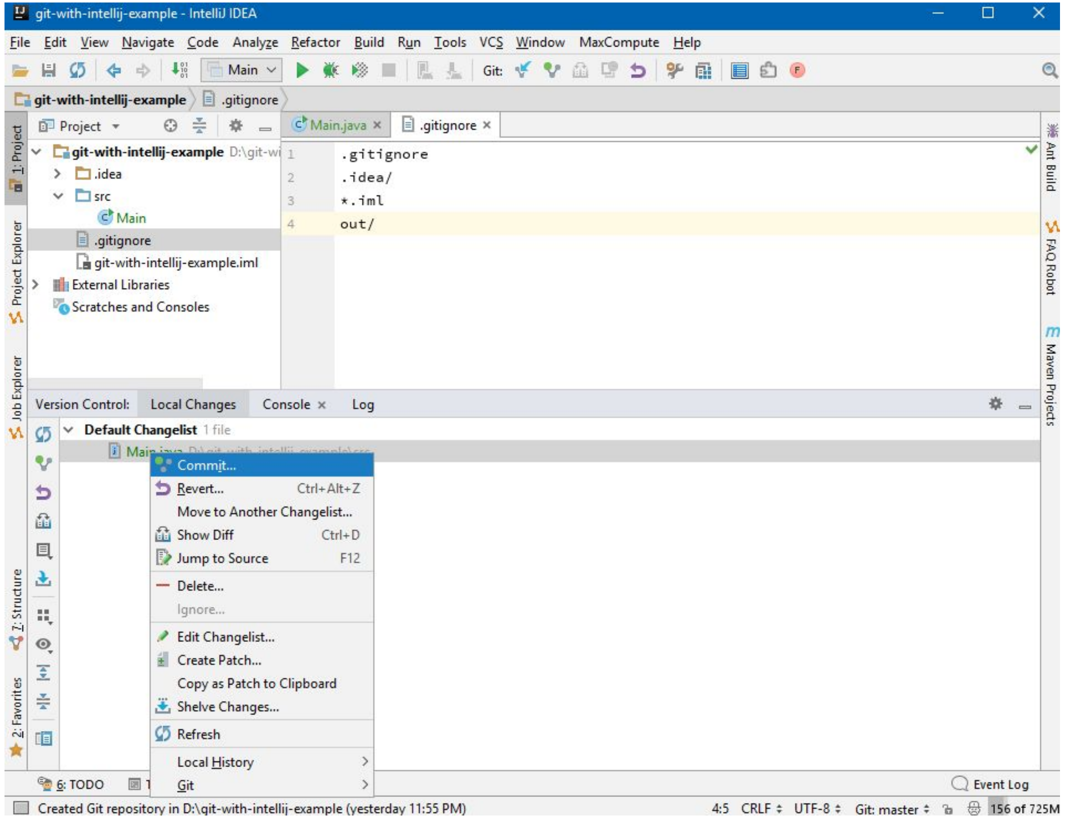
- Or we can also use right click>Git>Add as shown:



- After adding, the file color has changed to green (green is for newly added staged file).

Committing files

- Right click on the selected files to commit:



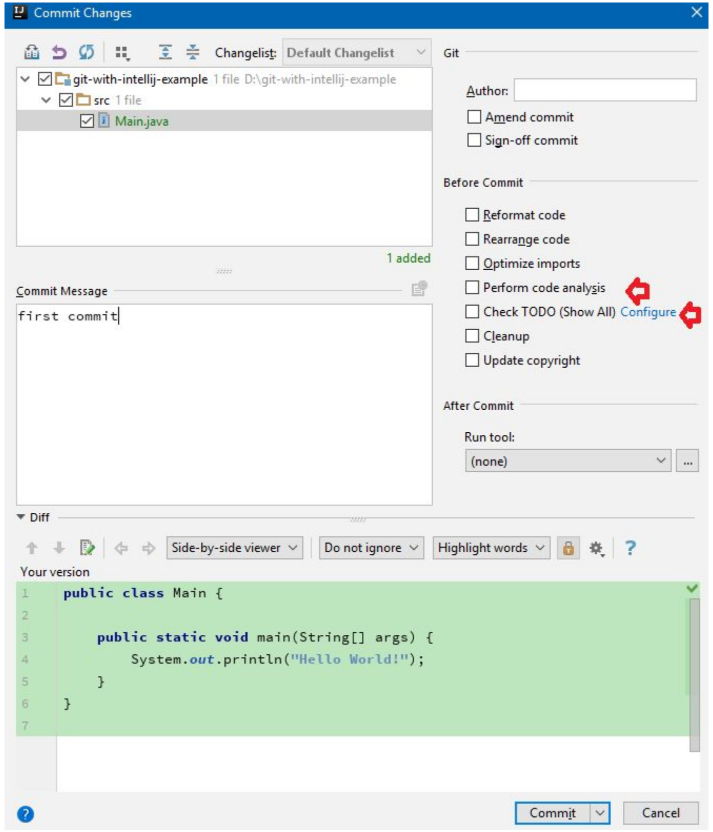
```
.gitignore
.idea/
*.iml
out/
```

Created Git repository in D:\git-with-intellij-example (yesterday 11:55 PM)

TIA-Academy

Committing files

- Enter commit message.
- Also if you don't want to 'perform code analysis' and 'check TODO', uncheck those options (they are checked by default):


Commit Changes

Changelist: Default Changelist

- [x] git-with-intellij-example 1 file D:\git-with-intellij-example
  - [x] src 1 file
    - [x] Main.java

1 added

Commit Message

```
first commit
```

Git

Author:

- [ ] Amend commit
- [ ] Sign-off commit

Before Commit

- [ ] Reformat code
- [ ] Rearrange code
- [ ] Optimize imports
- [ ] Perform code analysis
- [ ] Check TODO (Show All) Configure
- [ ] Cleanup
- [ ] Update copyright

After Commit

Run tool: (none)

Diff

Side-by-side viewer | Do not ignore | Highlight words

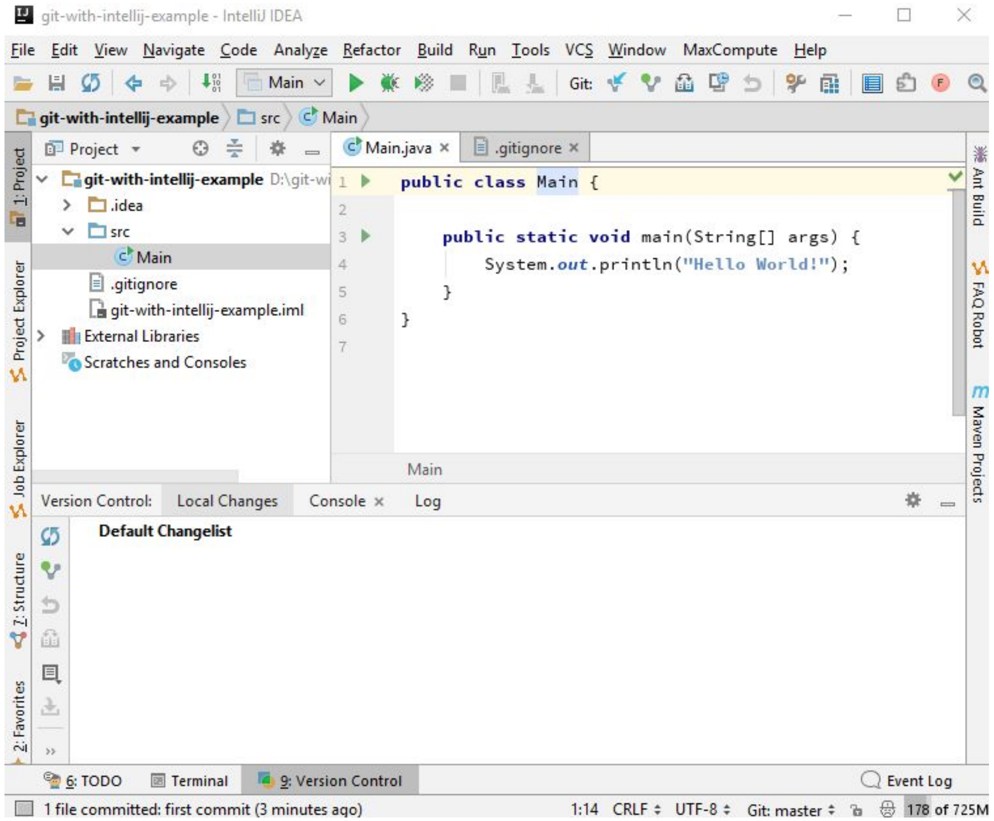
Your version

```
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Commit | Cancel

Committing files

- Click on 'commit'. Now Files from 'Default changelist' will disappear.
- The committed files' color will change to the default color:



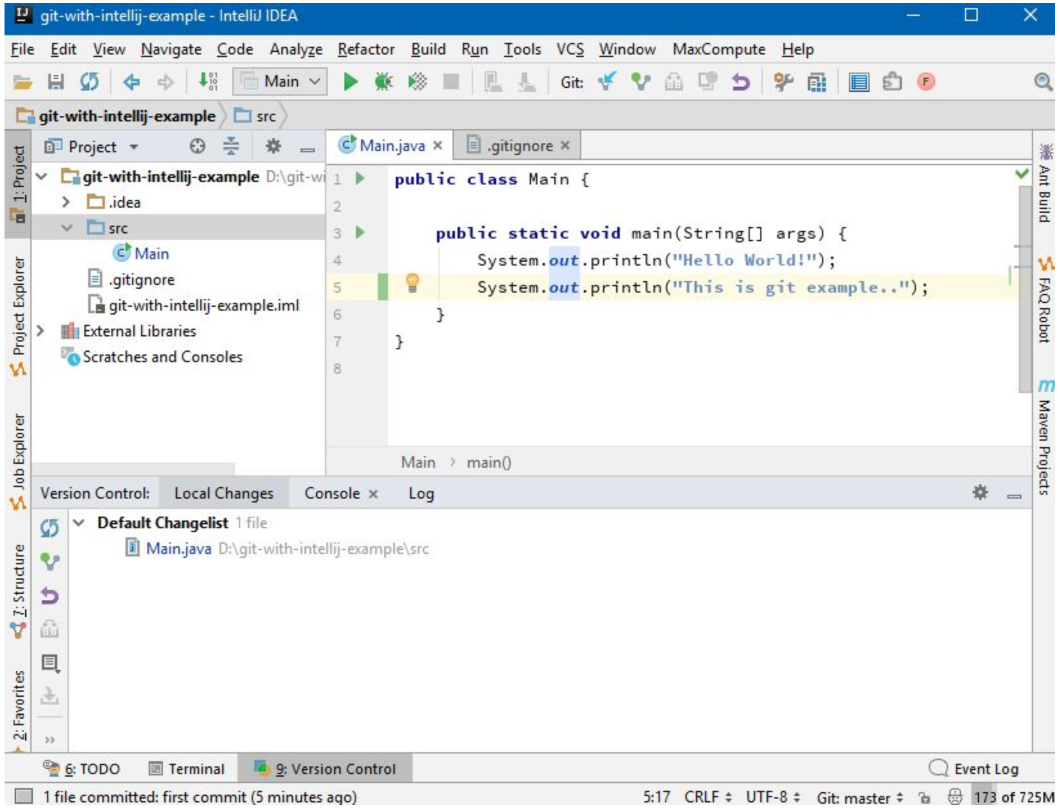
```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

1 file committed: first commit (3 minutes ago)

TIA-Academy

Making changes to versioned files

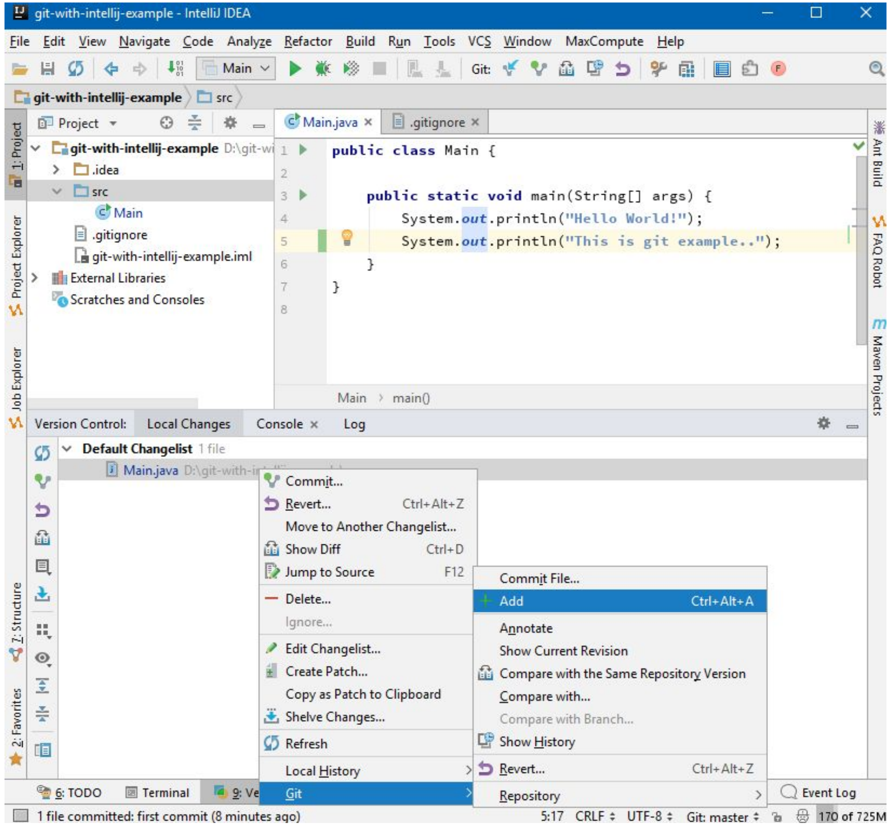
- Let's make some changes to our previously committed Java file.
- That will turn its color to blue:


git-with-intellij-example - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window MaxCompute Help
Main
Git:
git-with-intellij-example
src
Project
git-with-intellij-example D:\git-wi
.idea
src
Main
.gitignore
git-with-intellij-example.iml
External Libraries
Scratches and Consoles
Main.java
.gitignore
Main
main()
Version Control:
Local Changes
Console
Log
Default Changelist 1 file
Main.java D:\git-with-intellij-example\src
6: TODO
Terminal
9: Version Control
Event Log
1 file committed: first commit (5 minutes ago)
5:17 CRLF UTF-8 Git: master 173 of 725M

```
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
        System.out.println("This is git example..");
    }
}
```

Making changes to versioned files

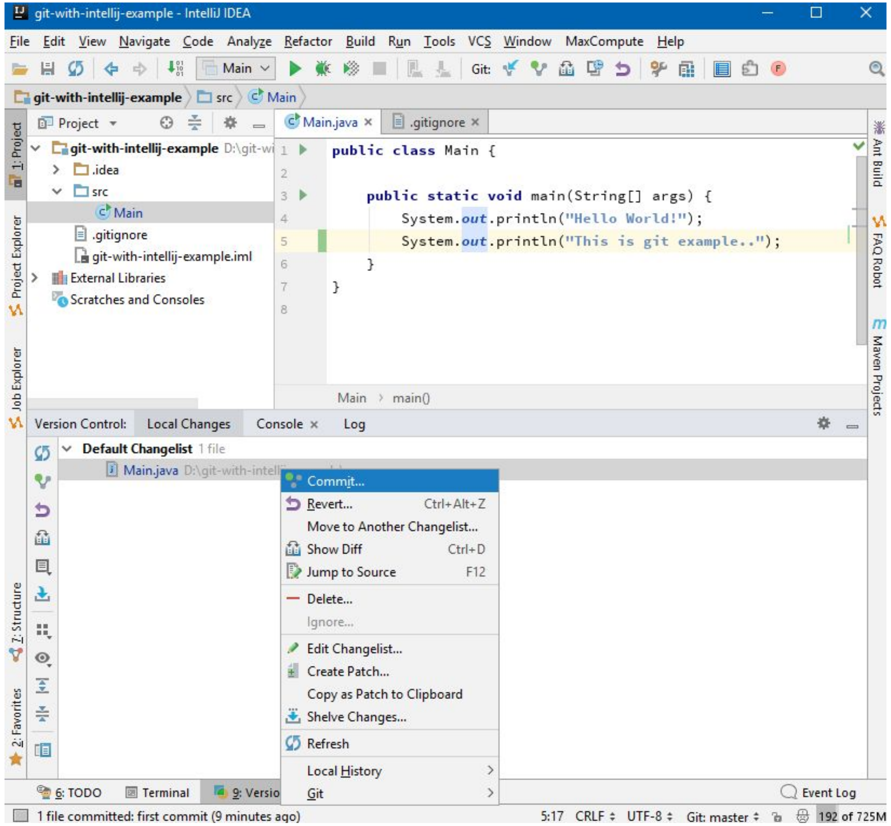
- Let's add the file to stagging and then commit.
- In intellij we can also commit files directly without adding them to staging first.
- The direct commit will perform both actions for us.



```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
        System.out.println("This is git example..");
    }
}
```

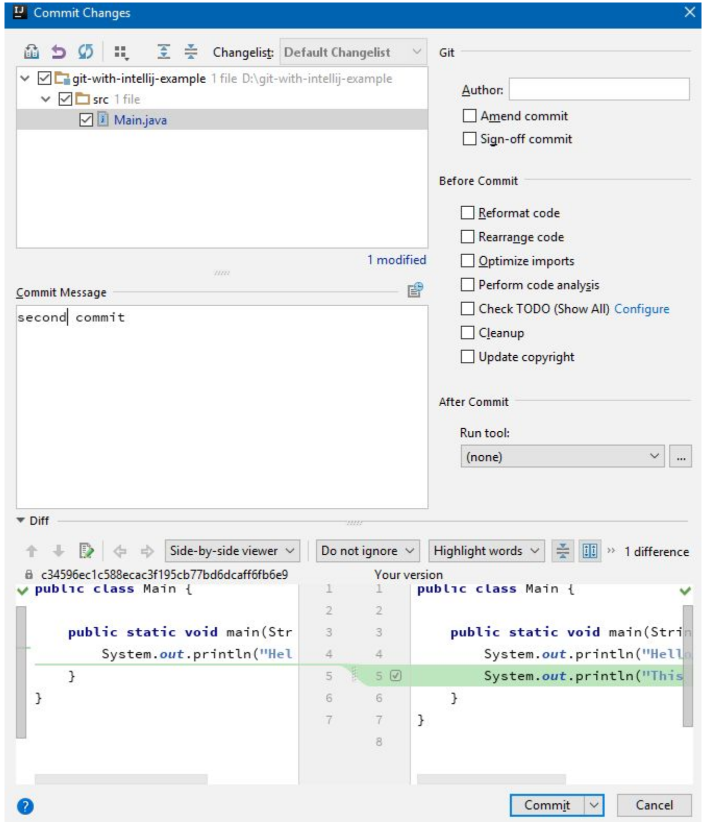
```
1 file committed: first commit (8 minutes ago)        5:17  CRLF ÷  UTF-8 ÷  Git: master ÷    170 of 725M
```

Making changes to versioned files



```
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
        System.out.println("This is git example..");
    }
}
```

1 file committed: first commit (9 minutes ago) 5:17 CRLF UTF-8 Git: master 192 of 725M



Changelist: Default Changelist

- ☑ git-with-intellij-example 1 file D:\git-with-intellij-example
  - ☑ src 1 file
    - ☑ Main.java

1 modified

Author:

Commit Message

```
second commit
```

Run tool: (none)

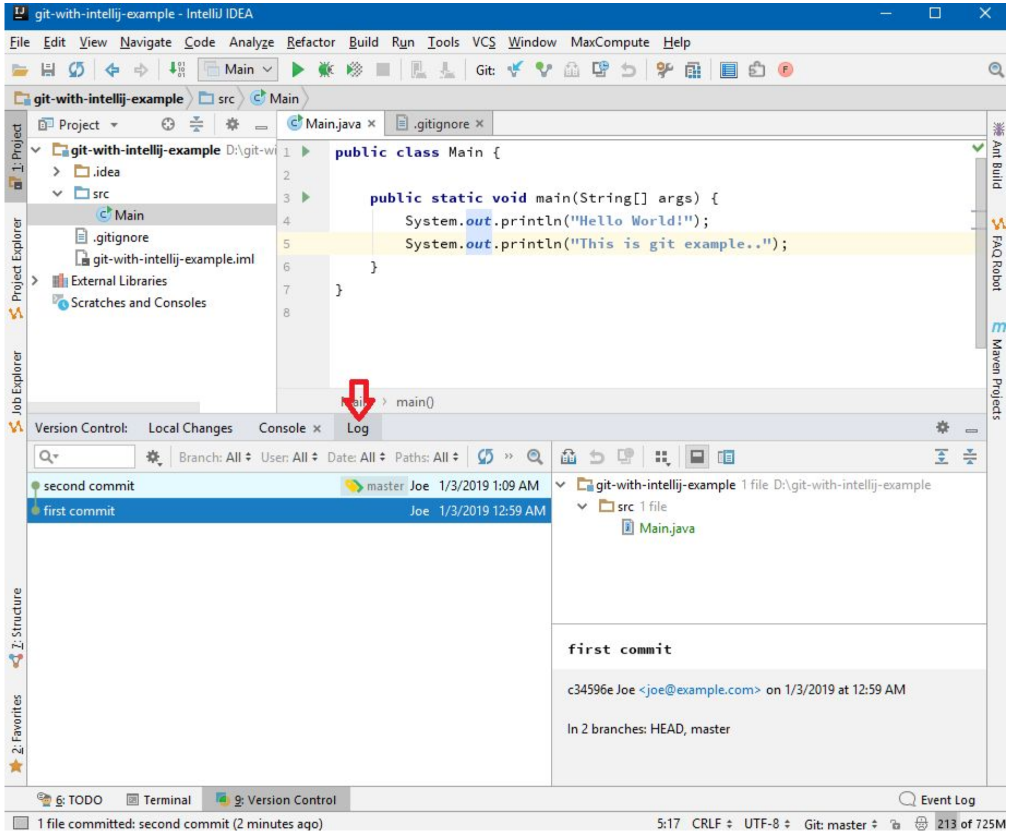
c34596ec1c588ecac3f195cb77bd6dcaff6fb6e9 | Your version

```
public class Main {

    public static void main(Str
        System.out.println("Hel
    }
}
```

```
public class Main {

    public static void main(Stri
        System.out.println("Hell
        System.out.println("This
    }
}
```

Git logs

- To see git logs click on 'Log' tab in Version Control view:


git-with-intellij-example - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window MaxCompute Help
Version Control: Local Changes Console Log
Branch: All User: All Date: All Paths: All
6: TODO Terminal 9: Version Control Event Log

```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
        System.out.println("This is git example..");
    }
}
```

```
second commit    master  Joe  1/3/2019 1:09 AM
first commit             Joe  1/3/2019 12:59 AM
```

git-with-intellij-example 1 file D:\git-with-intellij-example
src 1 file
Main.java

first commit

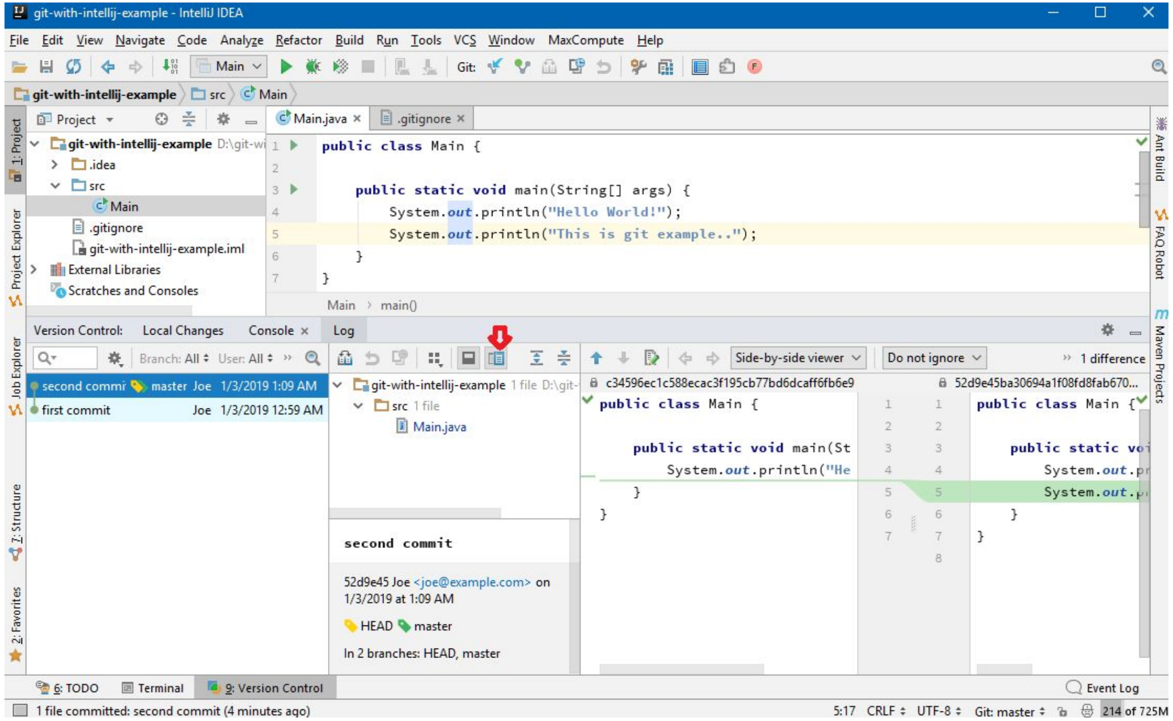
c34596e Joe <joe@example.com> on 1/3/2019 at 12:59 AM

In 2 branches: HEAD, master

1 file committed: second commit (2 minutes ago)

Git logs

- Toggle Select the "Preview Diff", that will show the change difference as well:


git-with-intellij-example - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window MaxCompute Help
git-with-intellij-example src Main
Project
git-with-intellij-example D:\git-wi
.idea
src
Main
.gitignore
git-with-intellij-example.iml
External Libraries
Scratches and Consoles
Main.java
.gitignore
Version Control: Local Changes Console Log
Branch: All User: All
Side-by-side viewer
Do not ignore
1 difference
Event Log
6: TODO
Terminal
9: Version Control

```java
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
        System.out.println("This is git example..");
    }
}
```

second commit	master	Joe	1/3/2019 1:09 AM
first commit		Joe	1/3/2019 12:59 AM

git-with-intellij-example 1 file D:\git-
src 1 file
Main.java

c34596ec1c588ecac3f195cb77bd6dcaff6fb6e9
52d9e45ba30694a1f08fd8fab670...

second commit

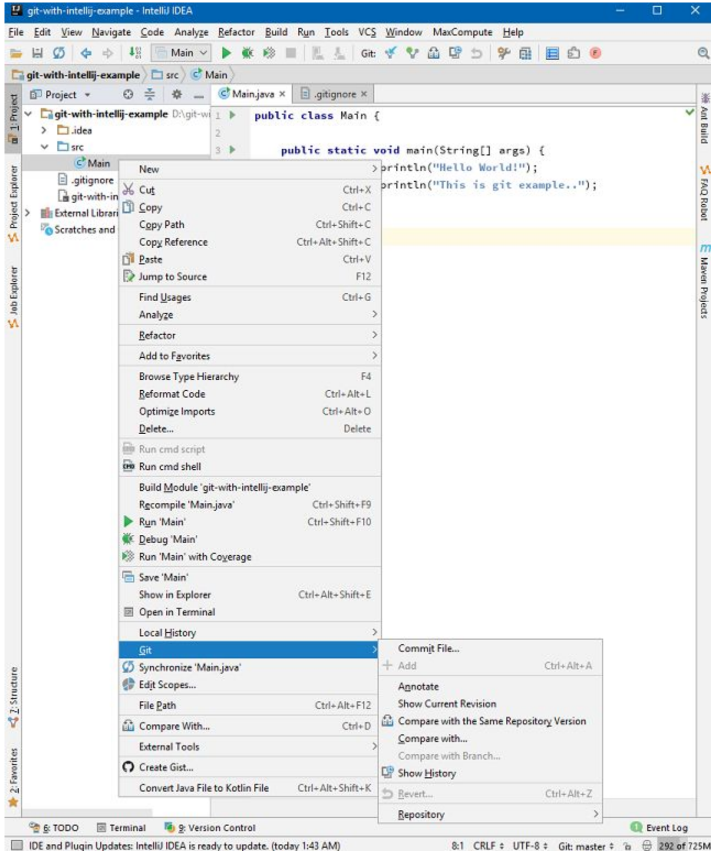
52d9e45 Joe <joe@example.com> on 1/3/2019 at 1:09 AM
HEAD master
In 2 branches: HEAD, master

1 file committed: second commit (4 minutes ago)
5:17 CRLF UTF-8 Git: master 214 of 725M

TIA-Academy

Right Click Git menu

- Various Git operations can also be performed by invoking right click menu on the selected files:



```
public class Main {

    public static void main(String[] args) {
        println("Hello World!");
        println("This is git example..");
```

www.ti-asia.com

ASSIGNMENT 02

TIA-Academy

Github & Intellij IDE

TIA-Academy

Sources

- https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

- https://www.jetbrains.com/help/idea/github.html

- https://www.youtube.com/watch?v=mM_drNdss4c

ASSIGNMENT 03 (HOME ASSIGNMENT)

TIA-Academy

Thank You