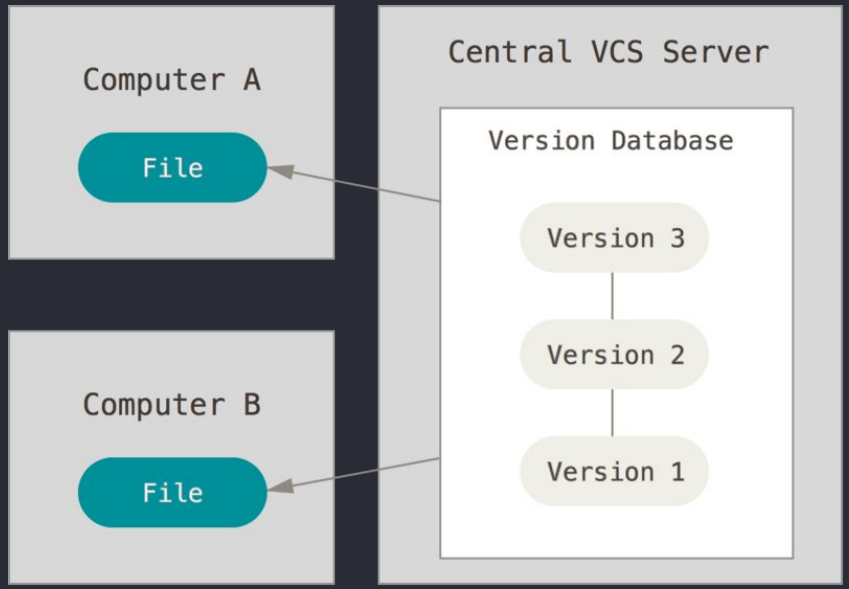
Version Control

WEEK 1

Advanced Native Mobile Programming
1604B062

VERSION CONTROL

Version control, also known as revision control or source control, is the management of changes to documents, computer programs, large websites, and other collections of information.



VERSION CONTROL

- Track developments and changes in your files
- Record the changes you made to your file in a way that you will be able to understand later
- Experiment with different versions of a file while maintaining the original version
- ‘Merge’ two versions of a file and manage conflicts between versions
- Revert changes, moving ‘backward’ through your history to previous versions of your file

TERMS

- **repository**

  contain a collection of files of various different versions of a Project.

- **There are two kind of repositories:**
  - Local repository
  - Remote repository

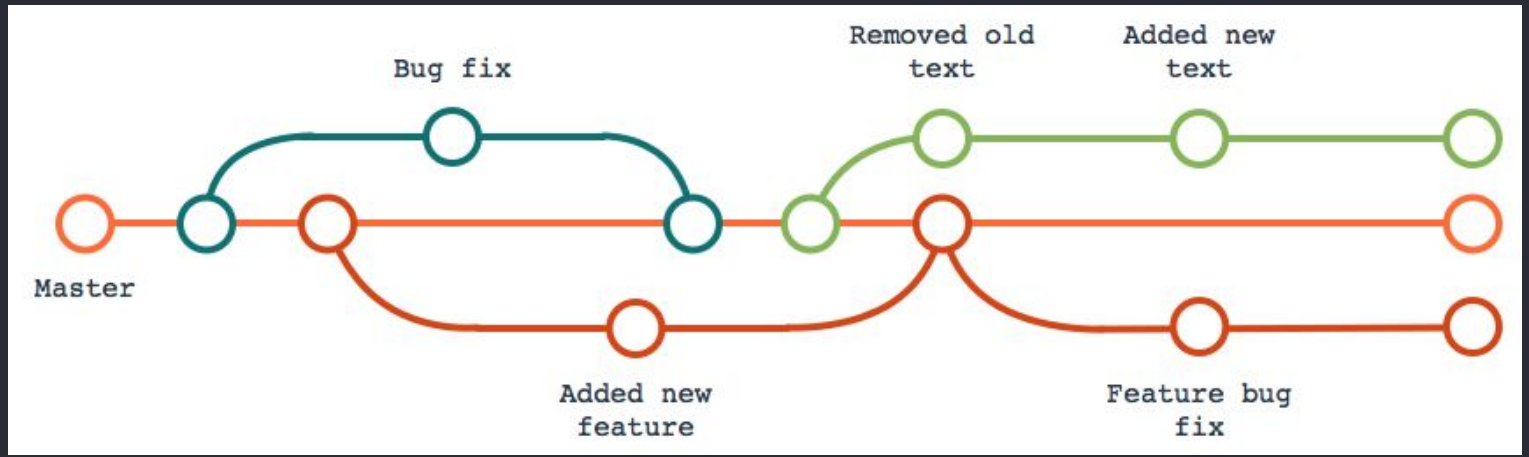
TERMS

- **clone**
  Create a new repository containing the revisions from another repository.
- **commit**
  To write or merge the changes made in the working copy back to the repository.
- **push**
  Copy revisions from the current repository to a remote repository.
- **pull**
  Copy revisions from a remote repository to the current repository.

TERMS

branch

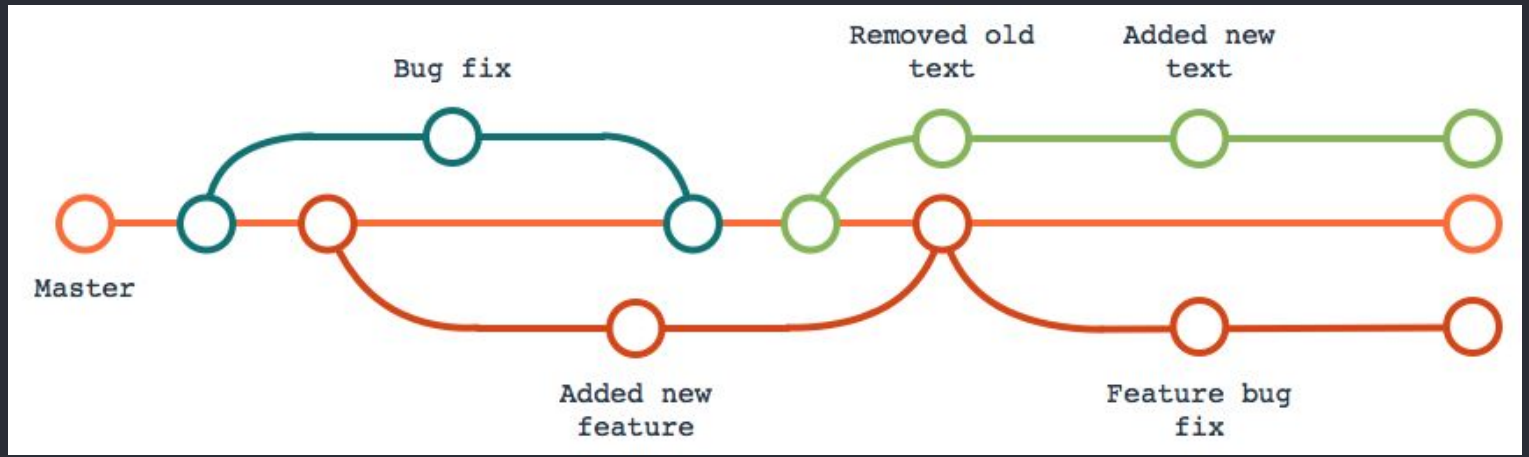
A separate working copy of files under version control which may be developed independently from the origin.



TERMS

merge

An operation in which two sets of changes are applied to a file or set of files.



VARIOUS VERSION CONTROL SYSTEM

GIT

used for source code management in software development. Created by Linus Torvalds in 2005

MERCURIAL

written in python and intended for software developers

SUBVERSION

abbreviated as SVN. Founded in 2000 by CollabNet, distributed as open source under the Apache License

POPULAR GIT REPOSITORY

- **github**
- **gitlab**
- **bitbucket**
- **sourceforge**

...or create a new repository on the command line

```
echo "# CoolApp" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/chieger/CoolApp.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/chieger/CoolApp.git
git push -u origin master
```

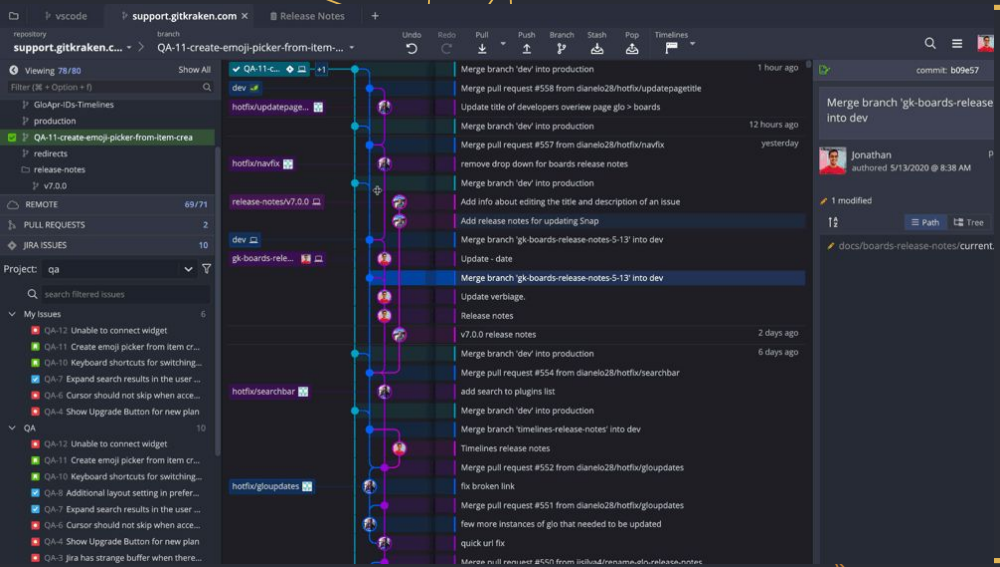
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

OPERATING GIT

Using terminal or command prompt



OPERATING GIT

Using GUI

ANDROID STUDIO

Android Studio has an integration with Git for source control management.

It means you can do the different Git operations (commit, push, pull, branch, etc.) from inside Android Studio

GROUP ASSESSMENT

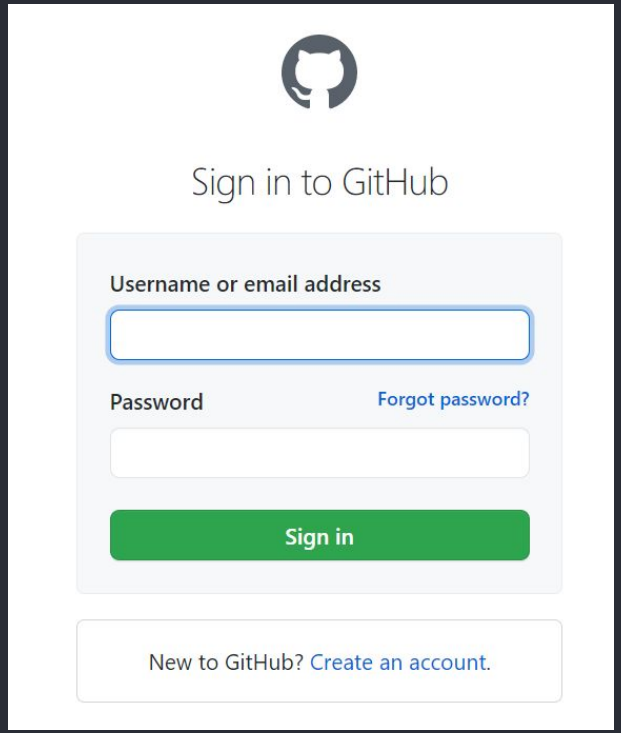
- Make a group of maximum three students
- Follow all steps on the next slides
- **Each student must create their own repository**
- Practice git operation (pull, push, clone, merge, branch, checkout, and so on) have been executed on one of repositories
- Group leader must submit / writes github repository URL of all member at ULS
- Deadline: next week

CREATE GITHUB ACCOUNT

Go to https://github.com/login, click on “Create an account” link to create a new github account. Follow all steps.

Don't forget to try sign in into github after completing account creation.

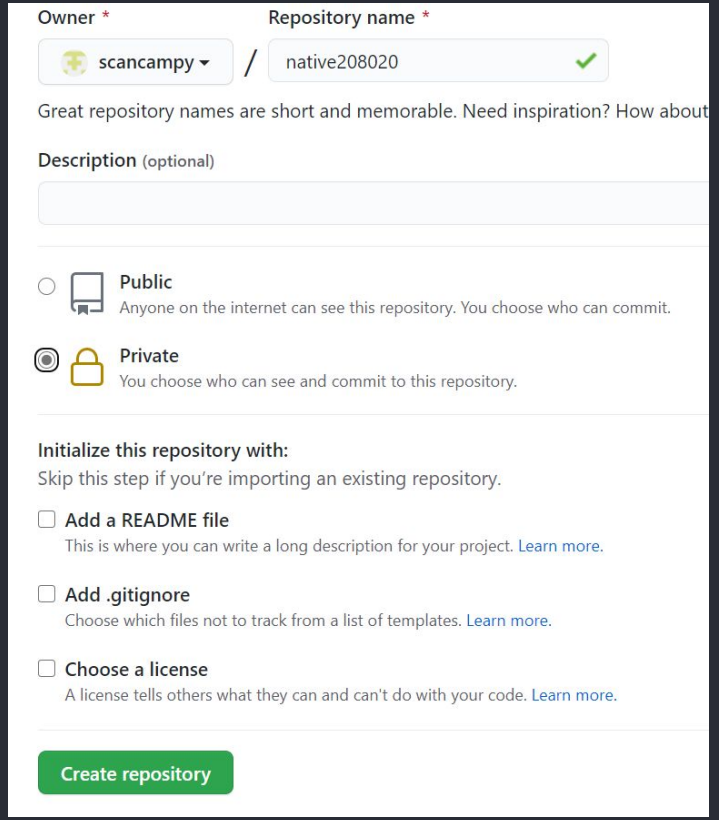
(if you already have github account, you may skip this step)



CREATE NEW REPOSITORY

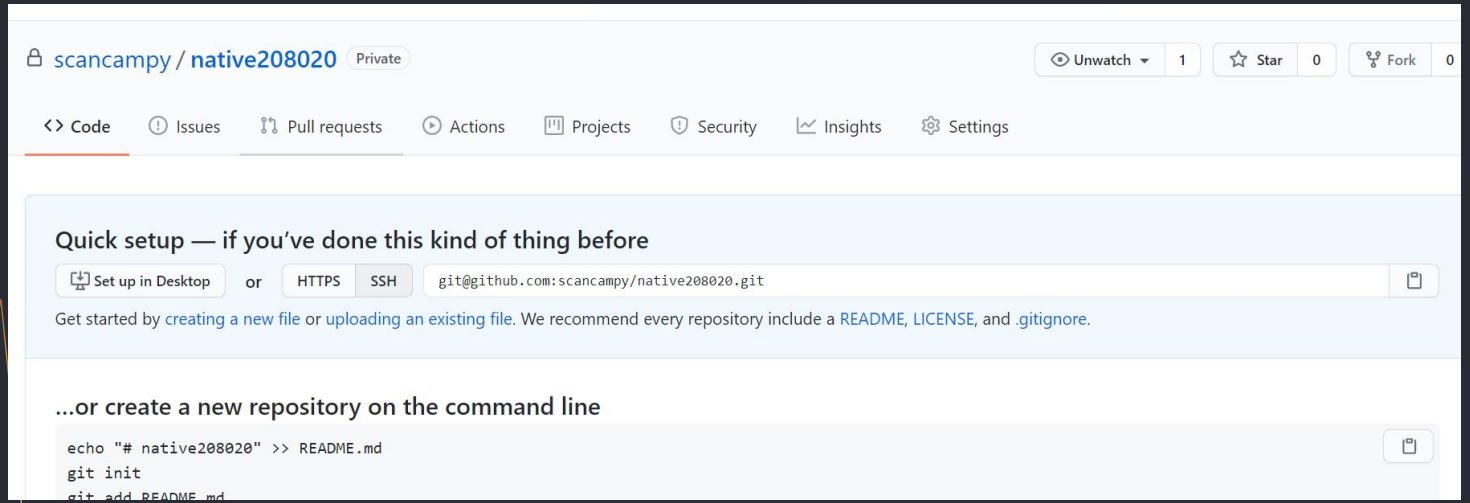
Once sign in into github, click new button to create new repository

1. Write repo name (without space)
   native[NRP]
2. Choose public repo
3. Click create repository



NEW REPOSITORY CREATED

New empty (remote) repository created

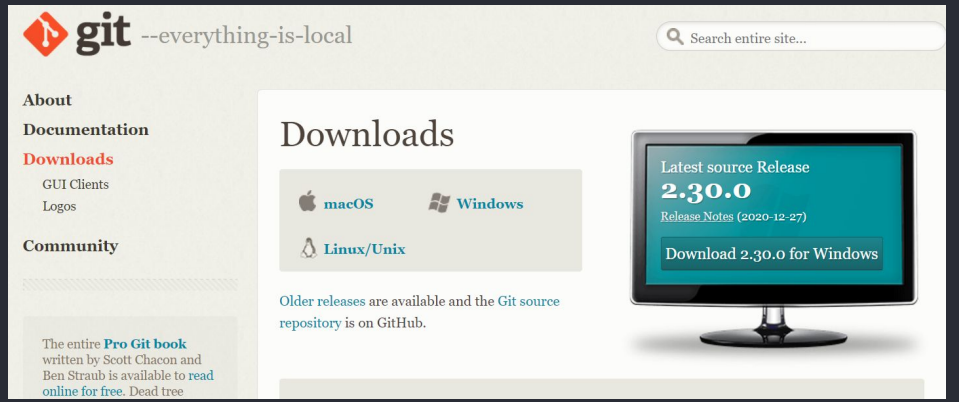


DOWNLOAD AND INSTALL GIT

Go to https://git-scm.com/downloads to download Git system to your computer

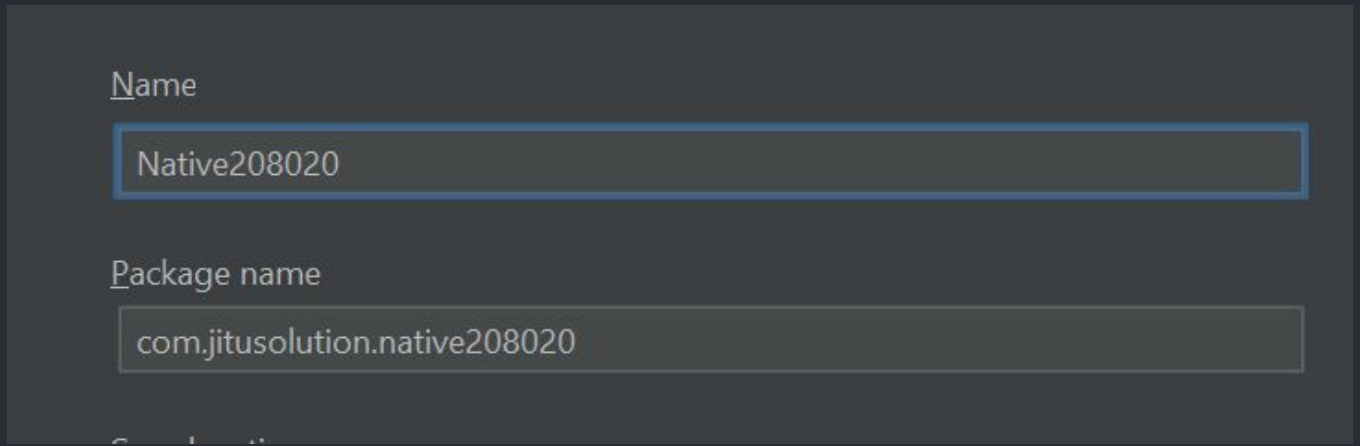
Install and follow the default settings

You may follow installation tutorial here: https://phoenixnap.com/kb/how-to-install-git-windows



CREATE NEW ANDROID STUDIO PROJECT

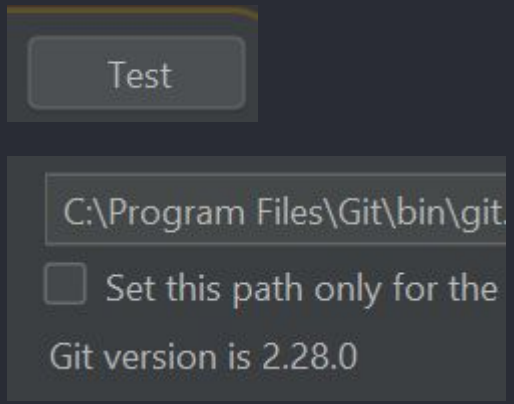
Create new project, use empty activity, and name project as **Native[NRP]**



CHECK GIT INTEGRATION

Next, we need to test whether Git is configured in Android Studio

1. Click file > settings
2. Type git in search bar
3. Click “test” button
4. If you can see git version, means that android studio is configured properly with Git



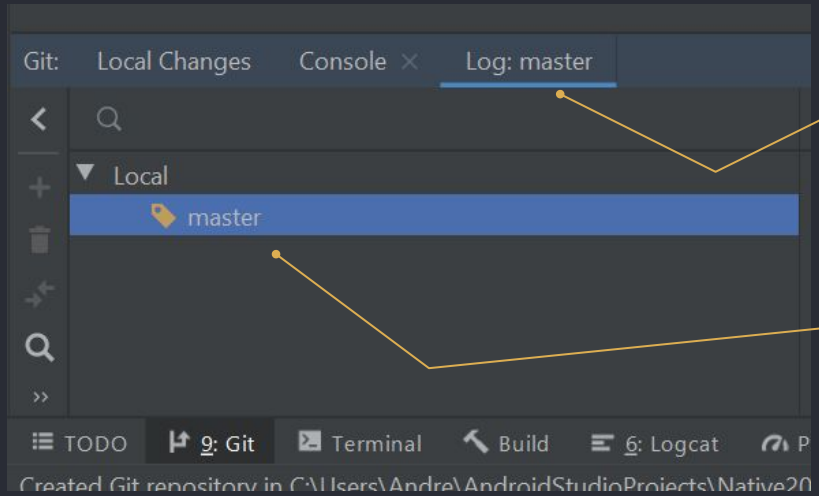
INTEGRATED GIT INTO PROJECT

Close setting dialog. Next, we need to enable VCS integration into our current project

1. Click VCS > Enable Version Control Integration
2. Popup dialog shown, choose git and press OK
3. Now a new repository with “master” branch have been created on your local drive

GIT TAB

New git tab shown on the bottom part of android studio



Click log tab

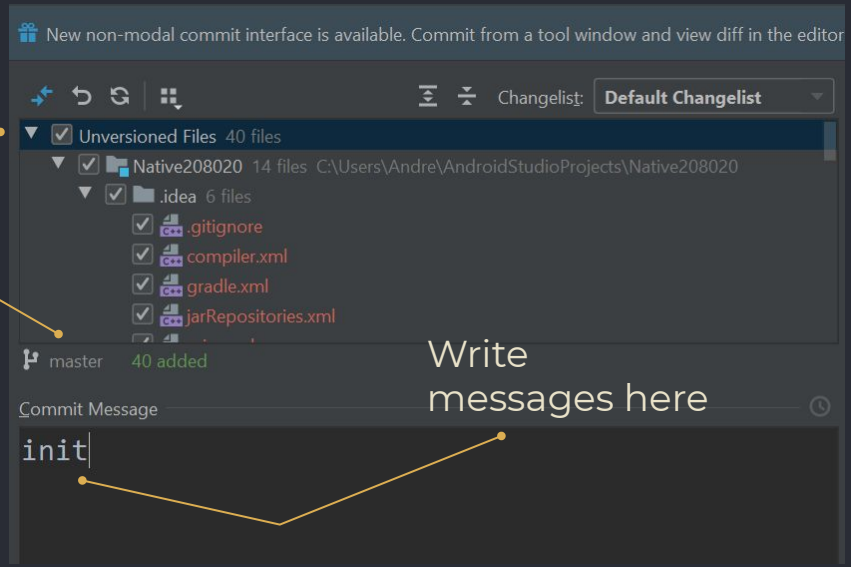
Notice that new branch “master” created

STAGE & COMMIT

The project is ready to use with Git version control. To stage and commit your changes, go to **VCS > Commit**.

Tick this checkbox

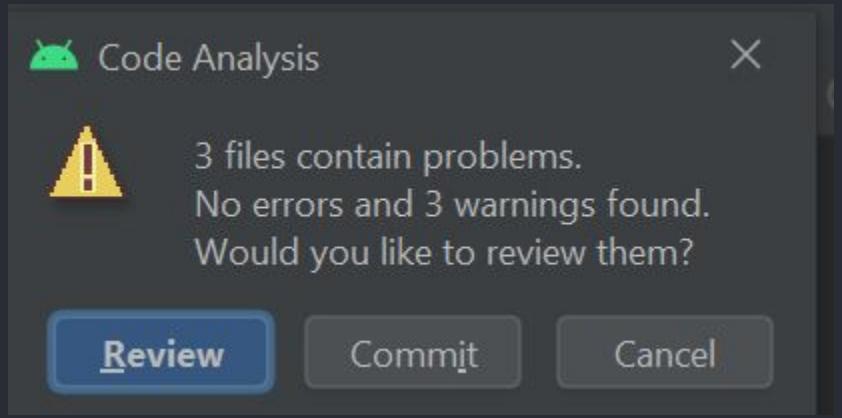
Current working branch



Write messages here

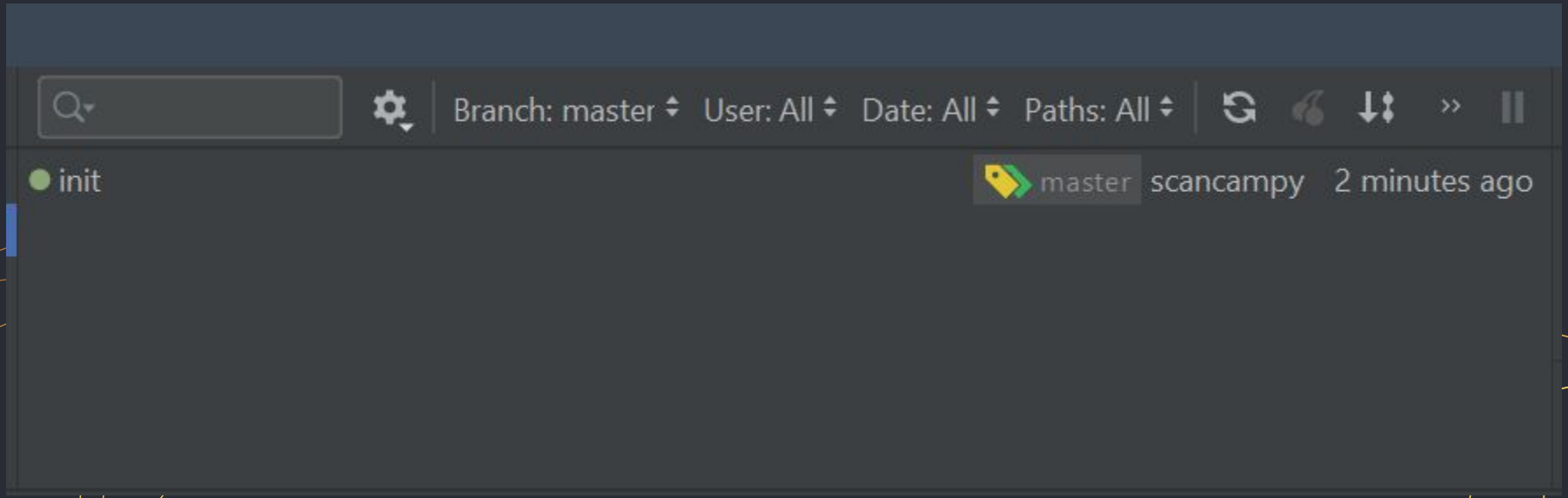
STAGE & COMMIT

Press commit button. Code analysis will show warning dialog like this. Warnings usually related with unfinished TODO, warning on UI or others. For now just press **Commit**



STAGE & COMMIT

Notice in git tab log, a new commit shown

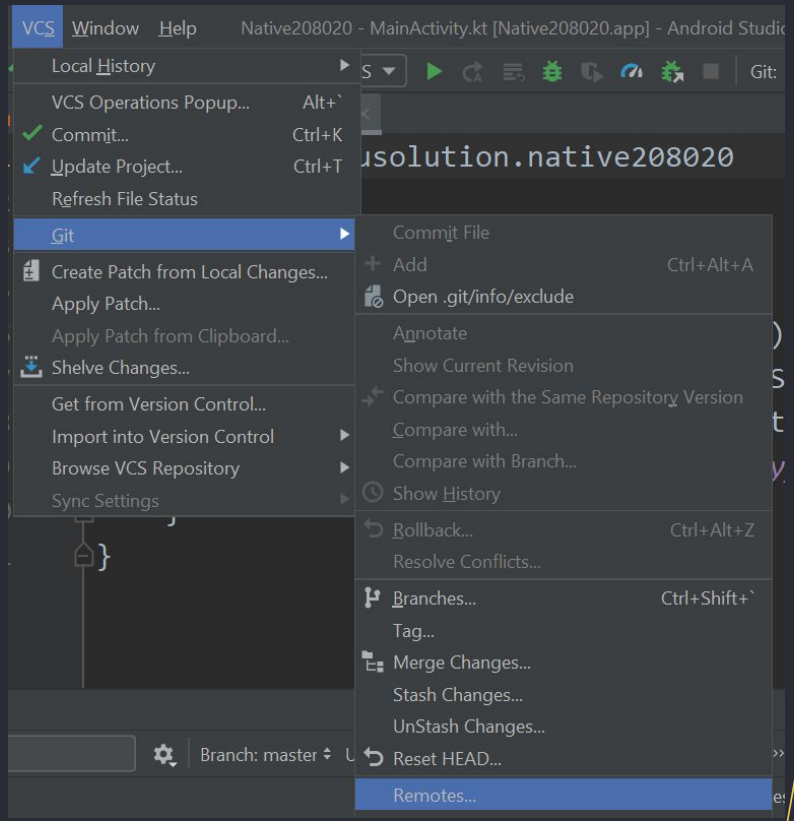


SETUP REMOTE CONNECTION

Previously we only staged and commited files into our own local repository

We need to push our commit to remote repository

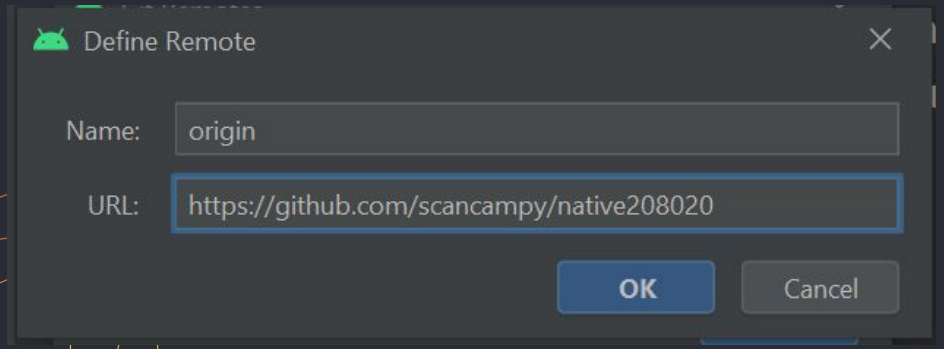
To configure remote repository click **VCS > Git > remote**

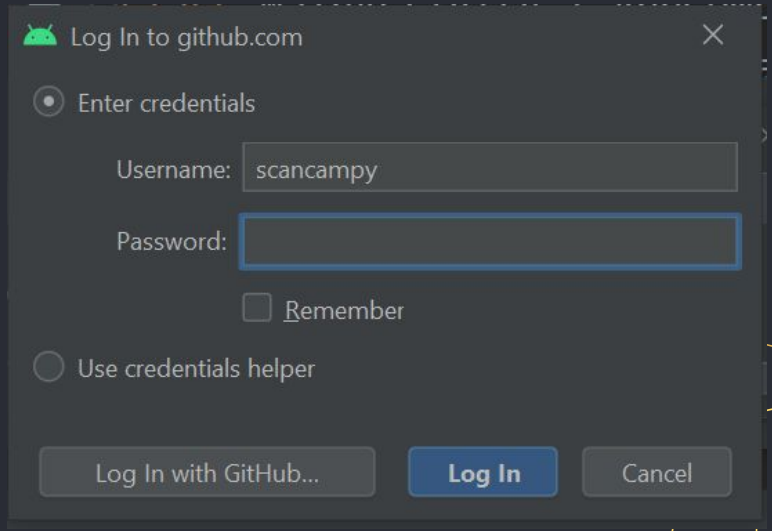


SETUP REMOTE CONNECTION

Click on + button, enter URL of your github repository that you created previously. Press OK

If authentication dialog shows up, you may need to provide github username and password.



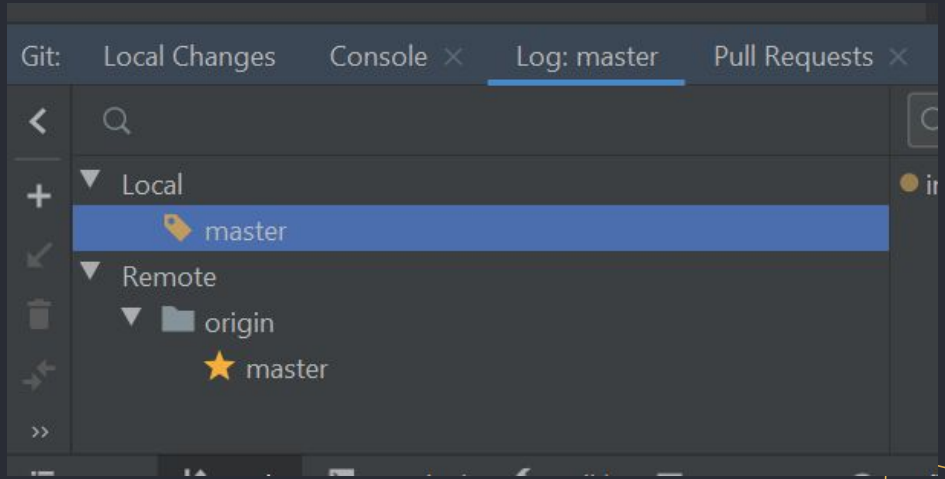


PUSH THE CHANGES TO REMOTE

To push your local changes to the remote repository, go to **VCS > Git > Push**

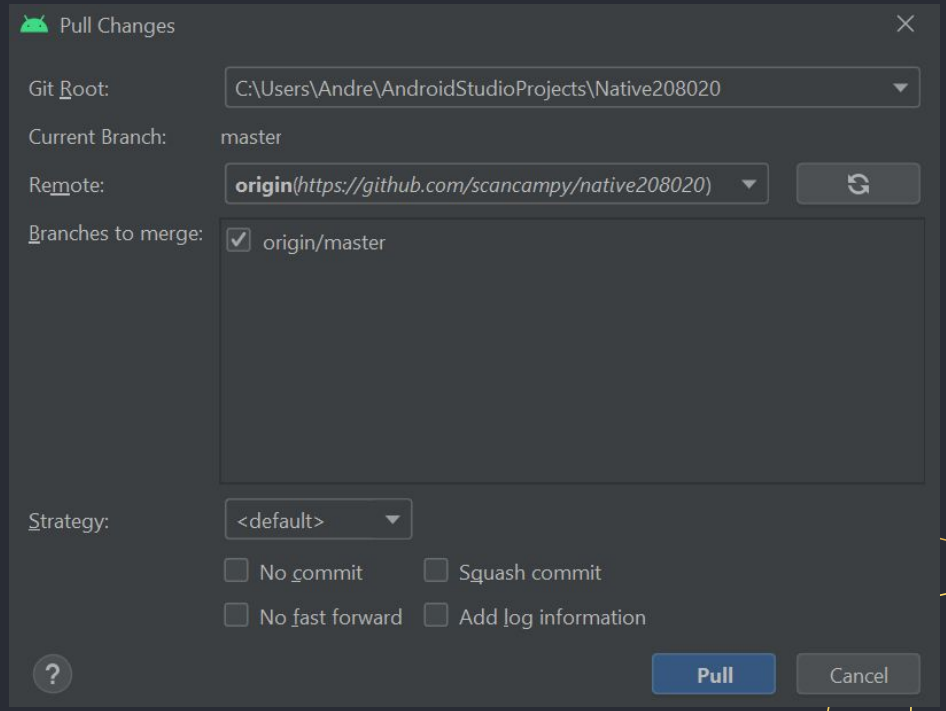
And then press push button again to begin push operation

If success, you can see two repositories (local and remote) shown on git log tab



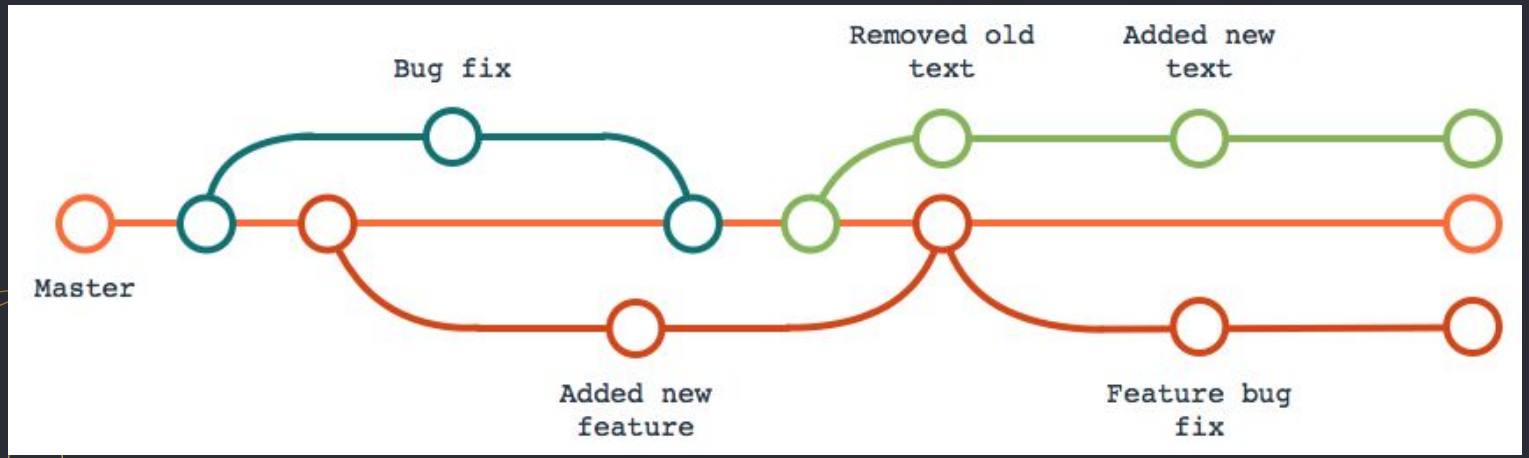
FETCH THE CHANGES FROM REMOTE

To download the latest changes from remote, go to **VSC > Git > Pull**



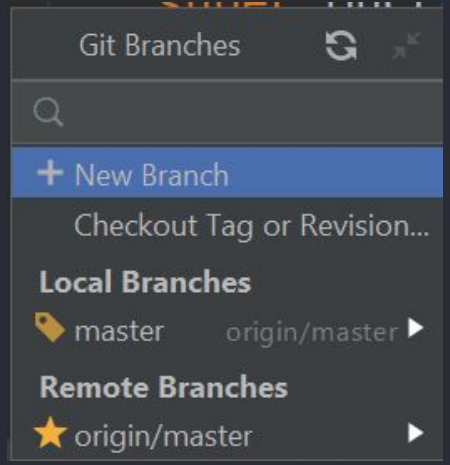
CREATE BRANCHES

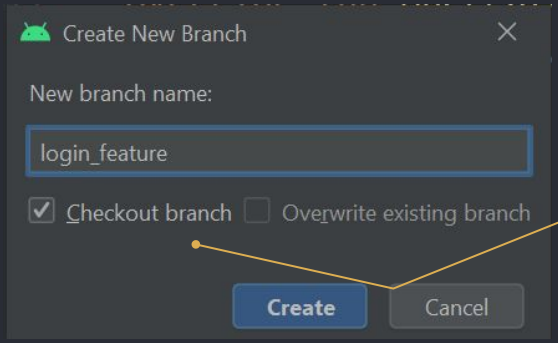
- Branching is used to maintain stability while isolated changes.
- Usually branch used for working on bug fixes, develop new features, integrate new versions after they have been tested in isolation



CREATE NEW BRANCH

- Go to VCS > Git > Branches.
- Choose + New Branch
- Name new branch
- Press create



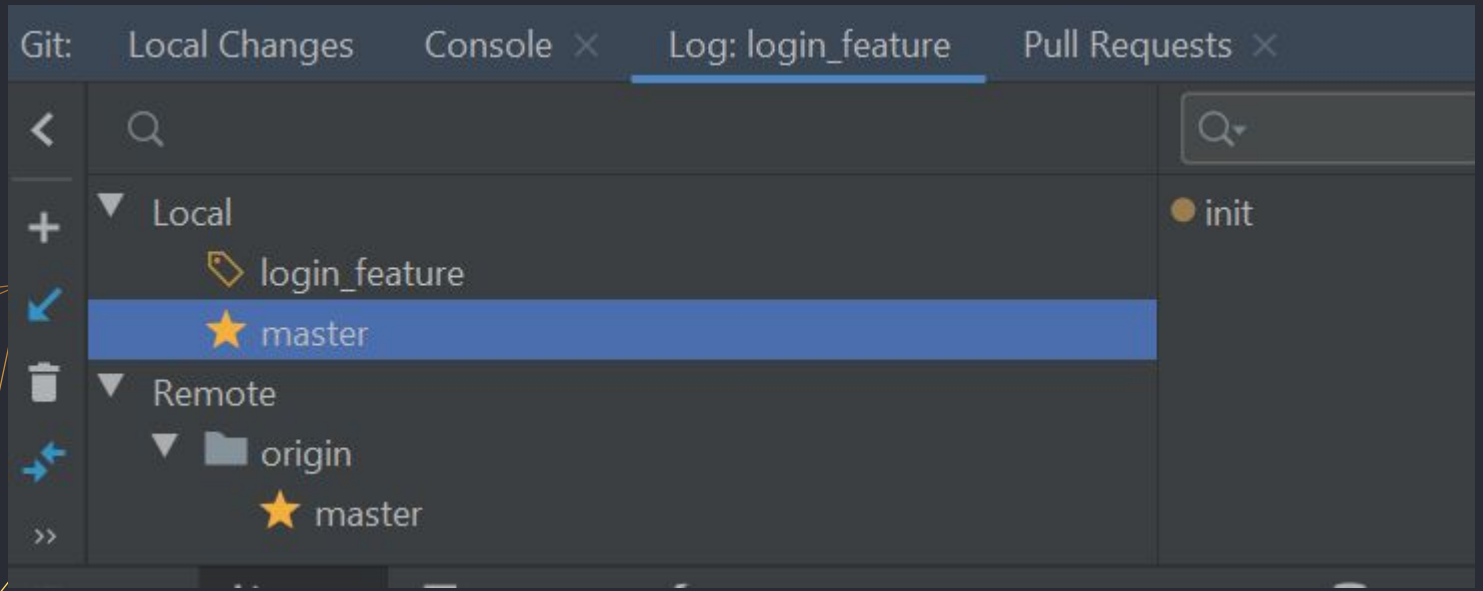


Checking out a branch updates the files in the working directory to match the version stored in that branch, and it tells Git to record all new commits on that branch

NEW BRANCH CREATED

Notice that now in local repository, there are a new branch “login feature”

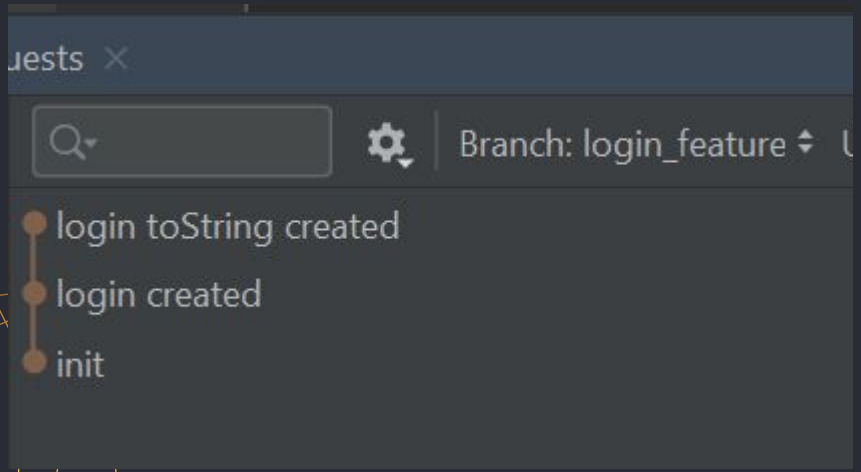
Any commit and push will affect this new branch

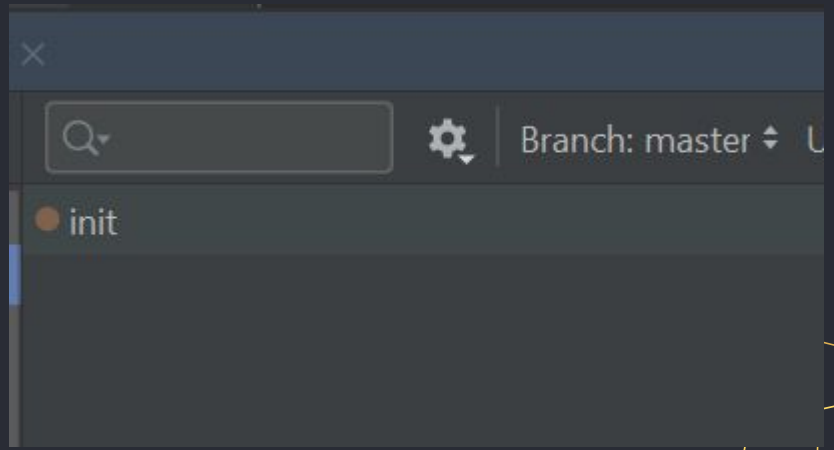


BRANCH

Below images are example of new “login_feature” branch already had two commit.

Master branch still only have 1 commit “init”

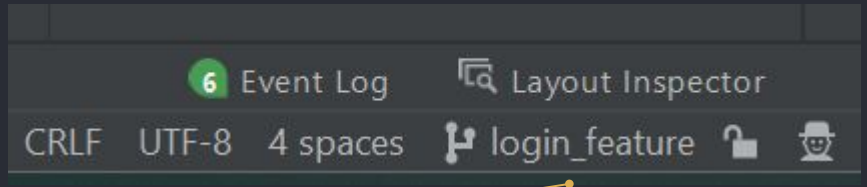


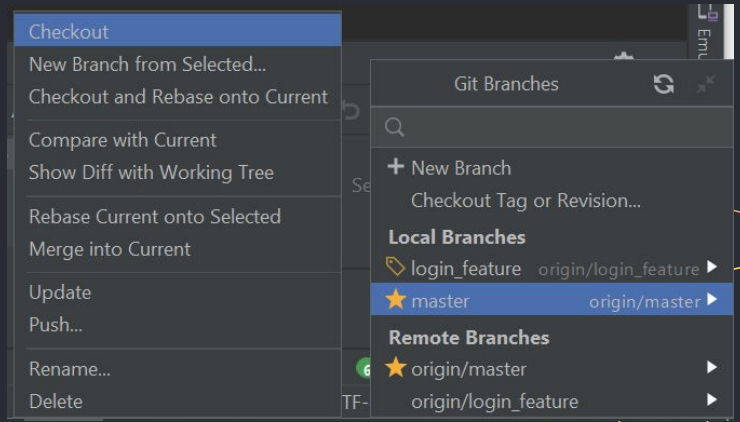


SWITCH/CHECKOUT BETWEEN BRANCH

Its very easy to switch between branch.

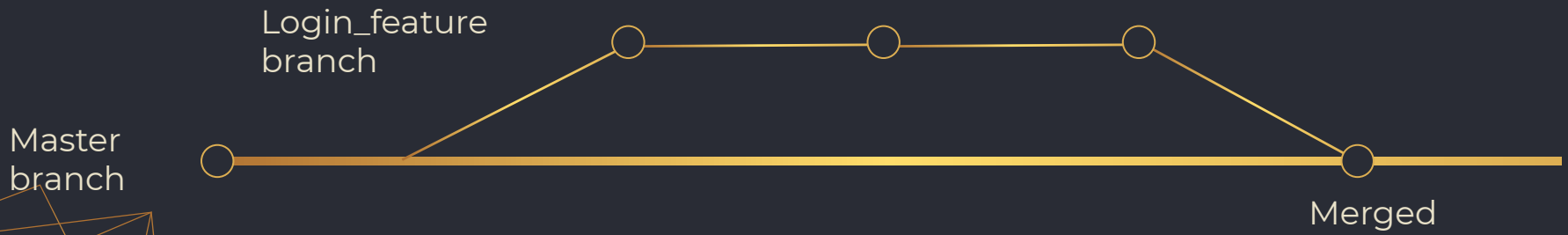
1. Look at the bottom right corner and you'll see the current working branch
2. Click on it to reveal all branches.
3. Click on branch that you want to switch/checkout
4. Choose checkout





MERGE BRANCH

Merging is Git's way of putting a forked history back together again

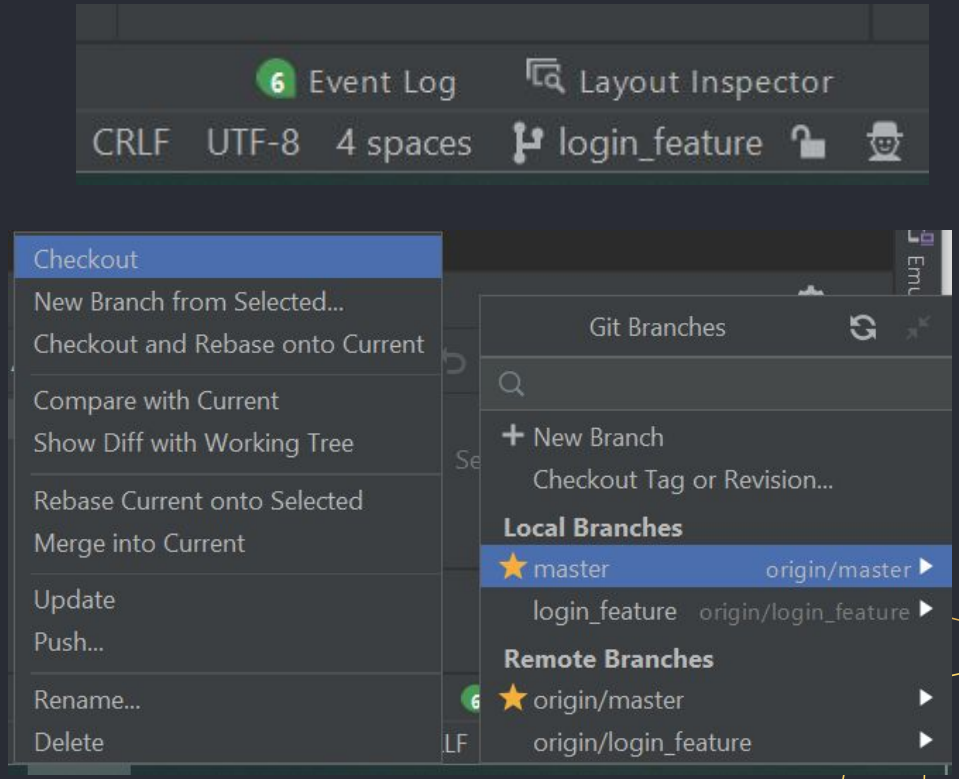


MERGE BRANCH

Let's say we want to merge **login_feature** branch into master branch

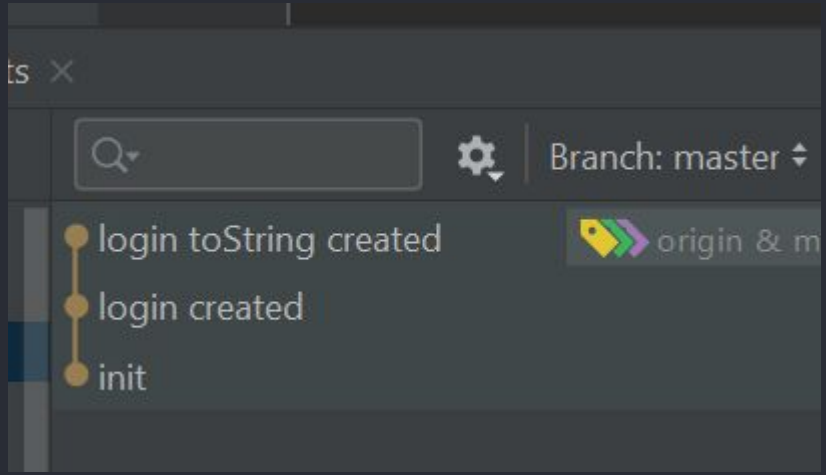
To do that look at the bottom right corner, click on the branch name

Choose master branch and click on “Checkout and Rebase onto Current”



BRANCH HAVE BEEN MERGED

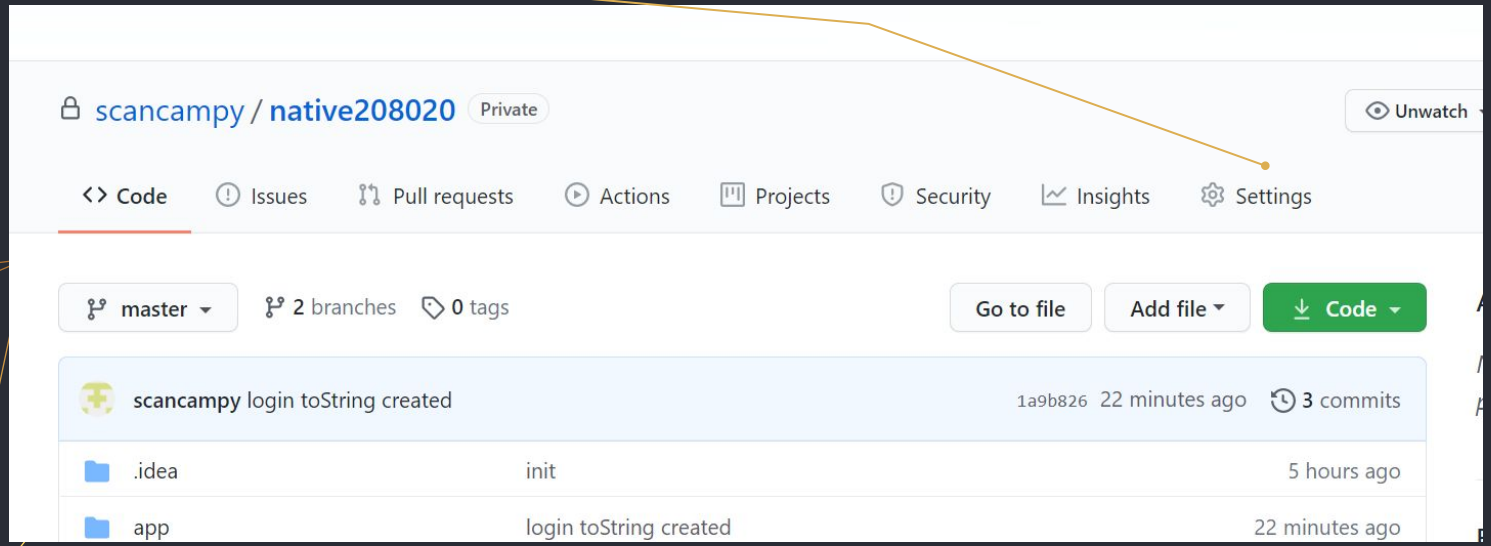
If you look on git log tab, you will see that “master” branch have been merged with “login_feature”



ALLOW ACCESS FOR OTHER TEAM MEMBER

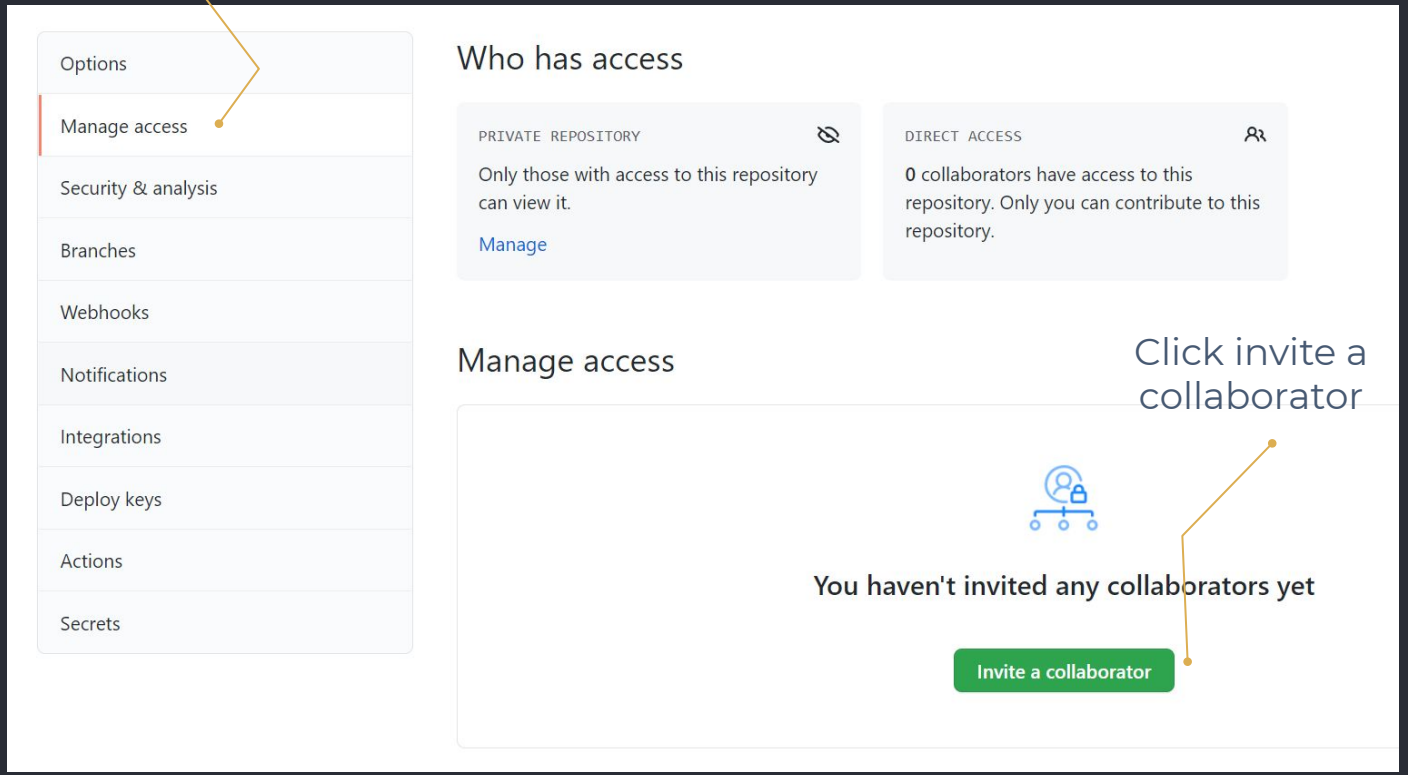
Since your repository is private, you must manually add other github account to give access privilege

To manage user access, open your repository from browser and click on “Settings”



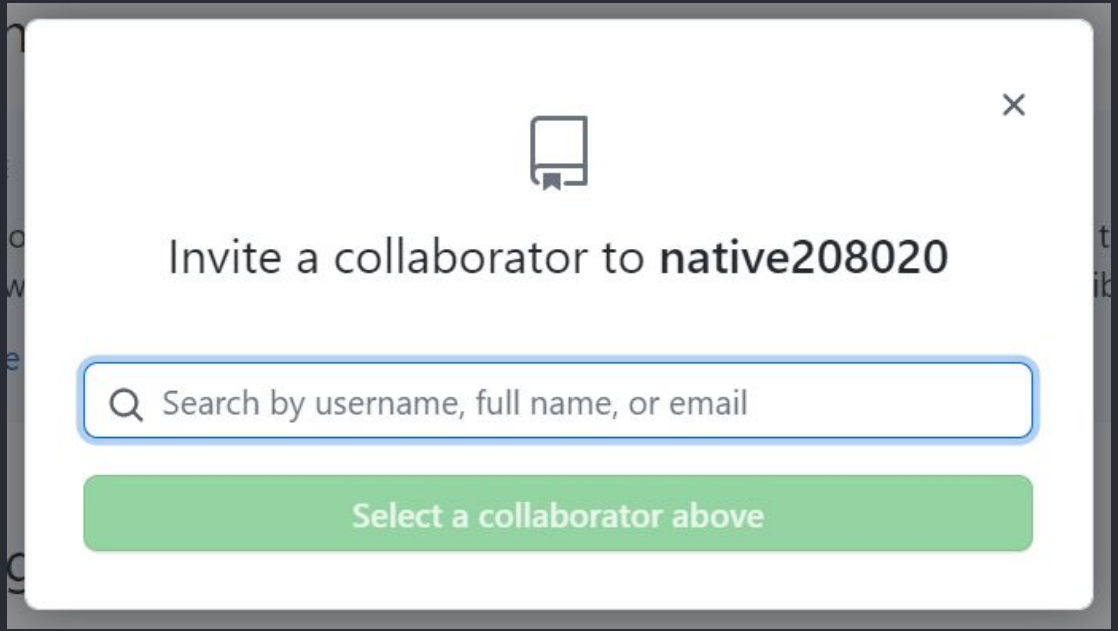
ALLOW ACCESS FOR OTHER TEAM MEMBER

Click menu
Manage access



ALLOW ACCESS FOR OTHER TEAM MEMBER

Write your friend email or github username. If success, your friend will have access to run git operation on this repository.



CLONE EXISTING PROJECT

There are tons of available github project on the internet.

Using android studio you can easily clone github project and start work on it

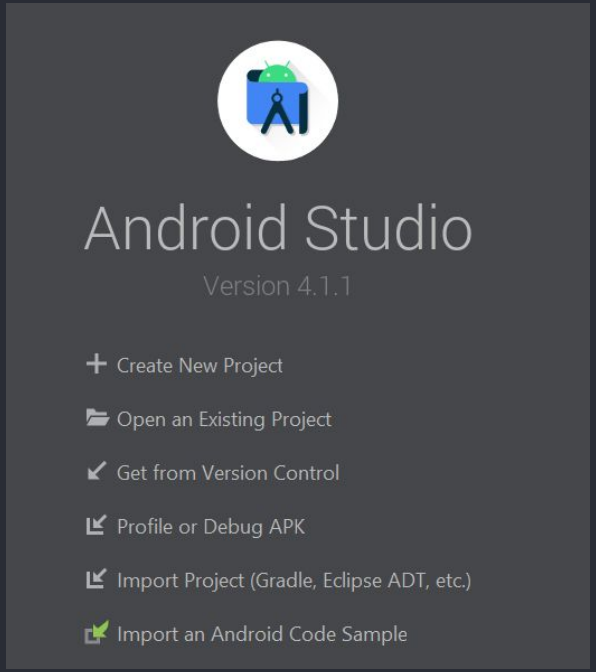
In this case lets clone your friend repository

Images of Clone Trooper
(courtesy of Starwars Universe)

CLONE EXISTING PROJECT

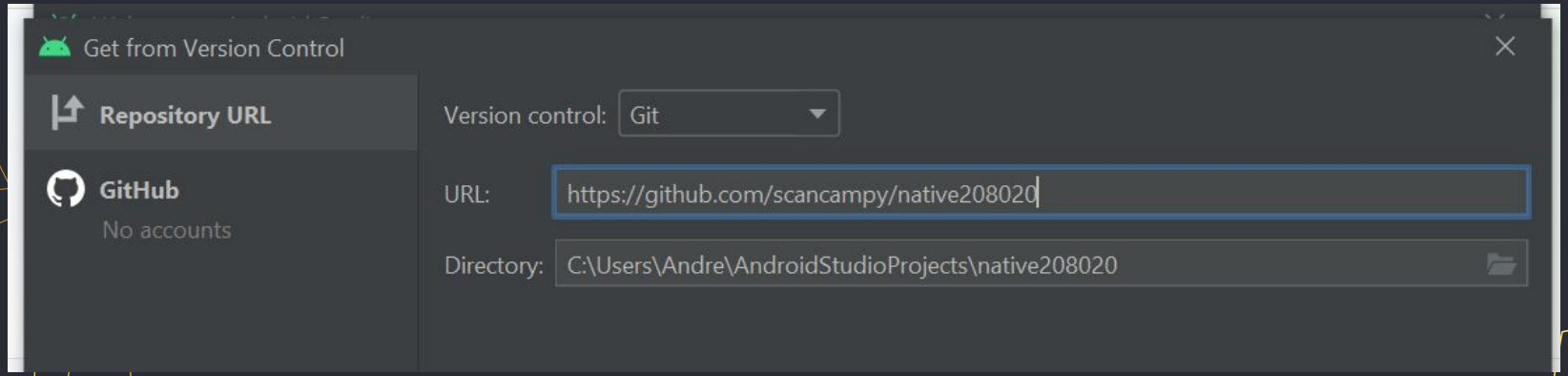
First close all project in android studio

On the Android Studio Welcome screen, click on “Get from Version Control” menu



CLONE EXISTING PROJECT

Copy paste your friend repository URL into URL textbox. Make sure your account already have access.



THANKS!

DO YOU HAVE ANY QUESTION?

andre@staff.ubaya.ac.id


slidesgo