

# FirstAide-Android Set-Up Guide

**This Guide is made for Windows PC with Android Studio for Android development.**

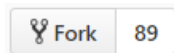
Using this Doc, you can easily set up FirstAide[Android] in your computer and contribute to the development of the project! The project can be set up through 2 methods – the command-line based approach, and the Android Studio based approach. Steps I, II and VI are common for both approaches. Following are the steps to set up the project -

## I. Install Git and create an account at GitHub.

- i. Signup to [www.github.com](https://www.github.com) if you haven't already to contribute to this project.
- ii. Download and set-up **git** from here – [git-for-windows.github.io](https://git-for-windows.github.io).
- iii. In this guide, I will use my username as "InfernoCoder". Replace it with your own username.

## II. Fork the FirstAide project repository from GitHub.

- i. Go to the project repo (<https://github.com/systers/FirstAide-Android>).
- ii. Click the 'Fork' button to fork the app.

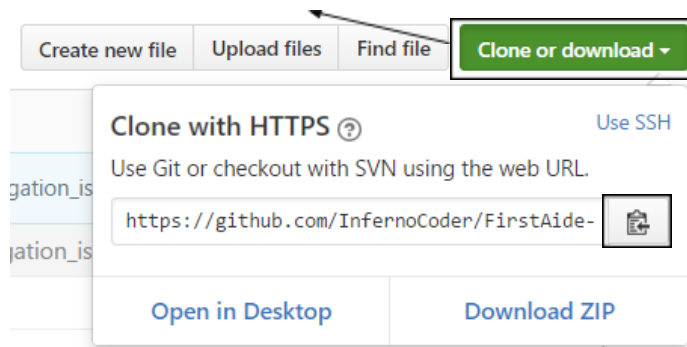


- iii. On clicking the 'Fork' button, you are redirected to your forked repository, whose URL is - [https://github.com/{YOUR\\_USERNAME}/FirstAide-Android](https://github.com/{YOUR_USERNAME}/FirstAide-Android). Here, YOUR\_USERNAME is the username you have in GitHub.

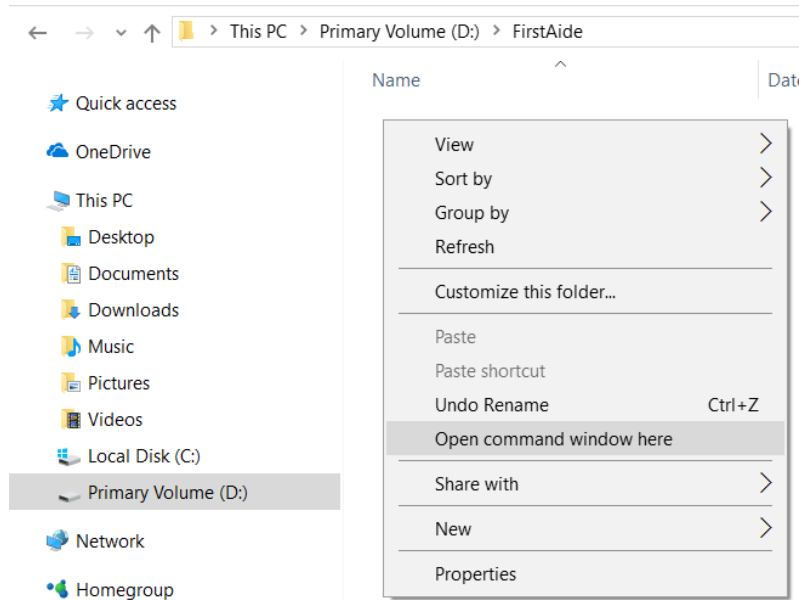
## (A) Command-line based approach -

### III. Clone the project to your local computer.

- i. Open the project repository you have forked at this URL - [https://github.com/{YOUR\\_USERNAME}/FirstAide-Android](https://github.com/{YOUR_USERNAME}/FirstAide-Android).
- ii. Click on the 'Clone or Download' button and then click on the 'Copy' button.



- iii. Open Windows Explorer and then go to/create a folder of your choice. Here I will create a folder at - “D:\FirstAide”. Open the folder after creating it.
- iv. Holding down **SHIFT** key, **RIGHT-CLICK** on the blank space. From the Options menu, select ‘Open command window here’ option.



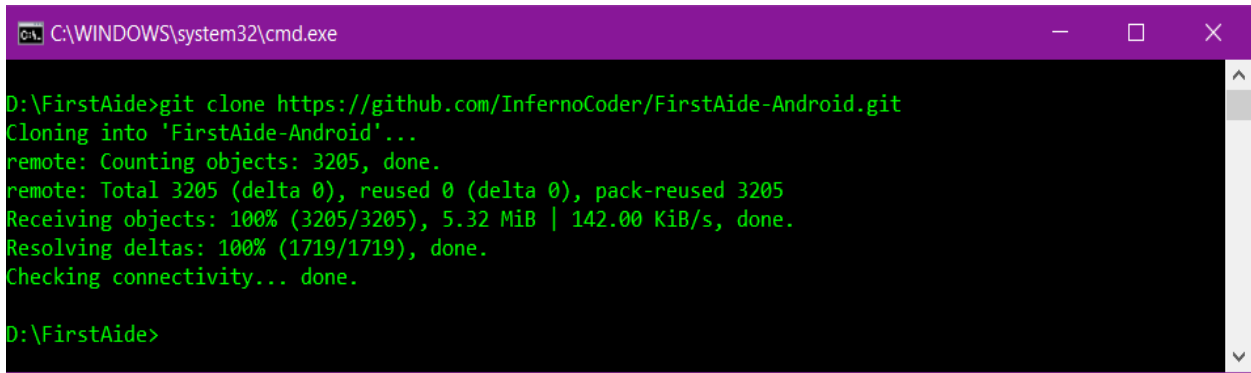
- v. Type the following in the Command Window and then ‘Paste’ the URL copied in step (ii).

```
git clone {PASTE HERE}
```

If you’re having problems pasting, type this command –

```
git clone https://github.com/{YOUR USERNAME}/FirstAide-Android.git
```

And replace {YOUR USERNAME} with your username. Then press ‘Enter’ key.



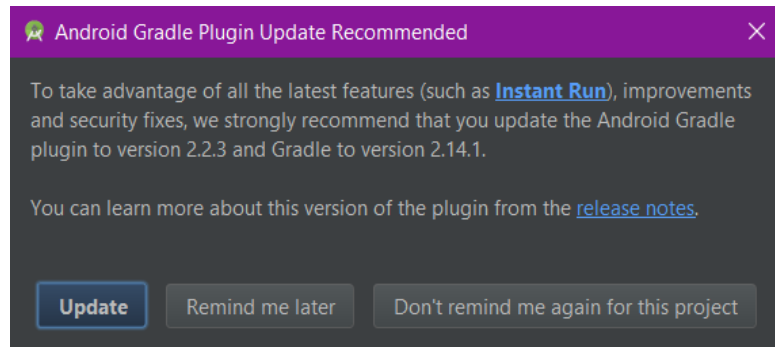
```
C:\WINDOWS\system32\cmd.exe

D:\FirstAide>git clone https://github.com/InfernoCoder/FirstAide-Android.git
Cloning into 'FirstAide-Android'...
remote: Counting objects: 3205, done.
remote: Total 3205 (delta 0), reused 0 (delta 0), pack-reused 3205
Receiving objects: 100% (3205/3205), 5.32 MiB | 142.00 KiB/s, done.
Resolving deltas: 100% (1719/1719), done.
Checking connectivity... done.

D:\FirstAide>
```

- vi. After downloading, the repo will be cloned to the folder 'FirstAide-Android'.

**NOTE:** In case the below dialog is shown when Android Studio is opened, click on “Don’t remind me again for this project” as this will prevent from changing the project’s Gradle version which may cause errors while uploading your changes.



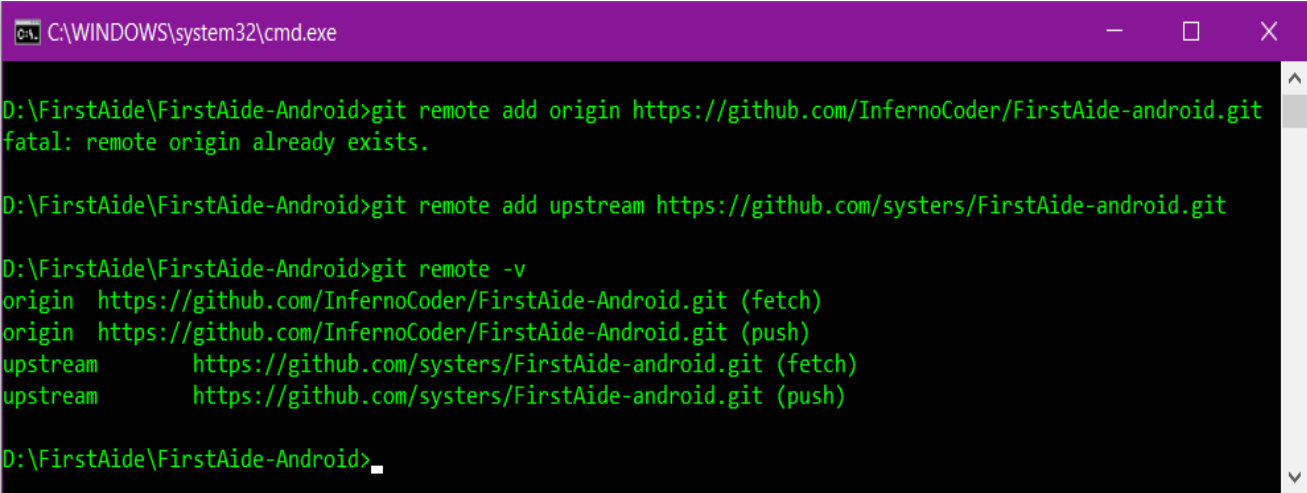
#### IV. Making the local repo ready for development

- i. Navigate to the folder to which the repo was cloned. In my case, it is **D:\FirstAide\FirstAide-Android**.
- ii. Open command window, as mentioned in **Step (III. -> iv.)**.
- iii. Set up the origin:

```
git remote add origin
https://github.com/{YOUR_USERNAME}/FirstAide-android.git
```
- iv. Set the upstream:

```
git remote add upstream
https://github.com/systers/FirstAide-android.git
```

- v. Run `git remote -v` to check the status. It should be something similar to the below picture.



```
C:\WINDOWS\system32\cmd.exe

D:\FirstAide\FirstAide-Android>git remote add origin https://github.com/InfernoCoder/FirstAide-android.git
fatal: remote origin already exists.

D:\FirstAide\FirstAide-Android>git remote add upstream https://github.com/systems/FirstAide-android.git

D:\FirstAide\FirstAide-Android>git remote -v
origin  https://github.com/InfernoCoder/FirstAide-Android.git (fetch)
origin  https://github.com/InfernoCoder/FirstAide-Android.git (push)
upstream https://github.com/systems/FirstAide-android.git (fetch)
upstream https://github.com/systems/FirstAide-android.git (push)

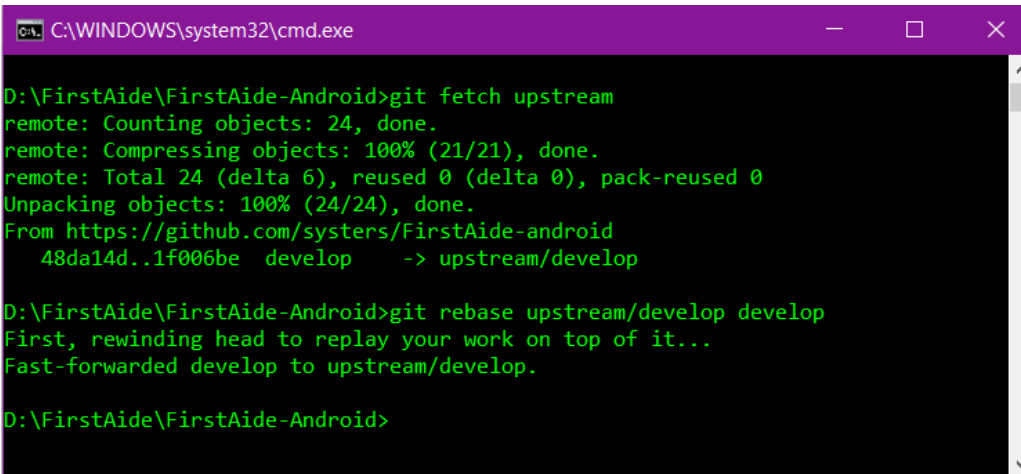
D:\FirstAide\FirstAide-Android>
```

- vi. If the local repo is old, and the upstream changed in the meanwhile, update the local repo from the upstream with the below commands –

**git fetch upstream**

**git rebase upstream/develop develop**

In my case, my branch was already updated to the latest commit, so it shows that my branch was up to date. Use this every time before making a change to ensure that the local repo is updated and there are no problems while creating a Pull Request to submit the changes.



```
C:\WINDOWS\system32\cmd.exe

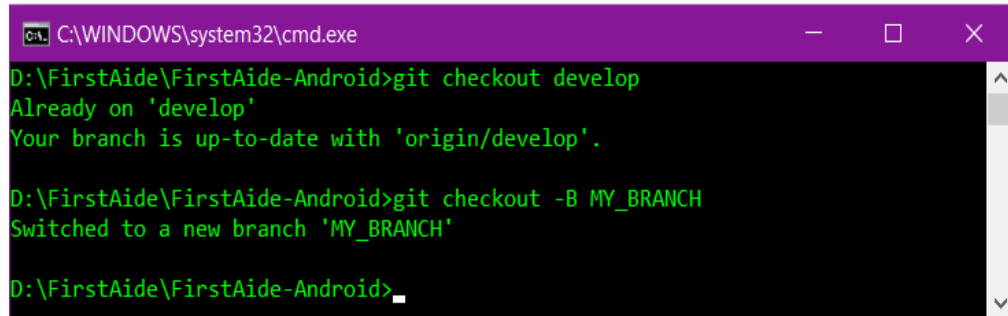
D:\FirstAide\FirstAide-Android>git fetch upstream
remote: Counting objects: 24, done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 24 (delta 6), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (24/24), done.
From https://github.com/systems/FirstAide-android
   48da14d..1f006be  develop    -> upstream/develop

D:\FirstAide\FirstAide-Android>git rebase upstream/develop develop
First, rewinding head to replay your work on top of it...
Fast-forwarded develop to upstream/develop.

D:\FirstAide\FirstAide-Android>
```

## V. Making new changes

- i. Be sure to be in the develop branch using this command '`git checkout develop`'.
- ii. Create a new branch with this command '`git checkout -B BRANCH_NAME`'.

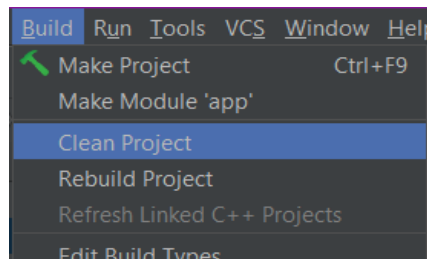


```
C:\WINDOWS\system32\cmd.exe
D:\FirstAide\FirstAide-Android>git checkout develop
Already on 'develop'
Your branch is up-to-date with 'origin/develop'.

D:\FirstAide\FirstAide-Android>git checkout -B MY_BRANCH
Switched to a new branch 'MY_BRANCH'

D:\FirstAide\FirstAide-Android>
```

- iii. Import the project as a 'New Project' in Android Studio, and then make the required changes.
- iv. After adding and testing the changes, clean the Project in Android Studio. See below picture for reference.



- v. Add the changes you have made with the below command. For example, I edited the AndroidManifest.xml file, so I'll use this –

```
git add
```

```
D:\FirstAide\FirstAide-Android\app\src\main\AndroidManifest.xml
```

*Avoid using '`git add *`' as it adds unnecessary files.*

- vi. Commit the changes by using the below command.

```
git commit -m "YOUR MESSAGE"
```

- vii. Push the changes to your remote repo

([https://github.com/{YOUR\\_USERNAME}/FirstAide-Android](https://github.com/{YOUR_USERNAME}/FirstAide-Android)) with this command-

```
git push --set-upstream origin [YOUR_BRANCH_NAME]
```

```
C:\WINDOWS\system32\cmd.exe

D:\FirstAide\FirstAide-Android>git add D:\FirstAide\FirstAide-Android\app\src\main\AndroidManifest.xml

D:\FirstAide\FirstAide-Android>git commit -m "AndroidManifest.xml formatted"
[MY_BRANCH c67d67a] AndroidManifest.xml formatted
1 file changed, 2 insertions(+), 3 deletions(-)

D:\FirstAide\FirstAide-Android>git push
fatal: The current branch MY_BRANCH has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin MY_BRANCH

D:\FirstAide\FirstAide-Android>git push --set-upstream origin MY_BRANCH
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 520 bytes | 0 bytes/s, done.
Total 6 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/InfernoCoder/FirstAide-Android.git
 * [new branch]      MY_BRANCH -> MY_BRANCH
Branch MY_BRANCH set up to track remote branch MY_BRANCH from origin.

D:\FirstAide\FirstAide-Android>_
```

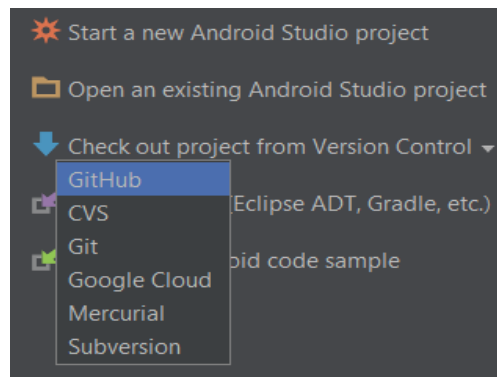
*Notice that I first typed just “git push” to get the right command*

### Step VI is at the end of Approach (B).

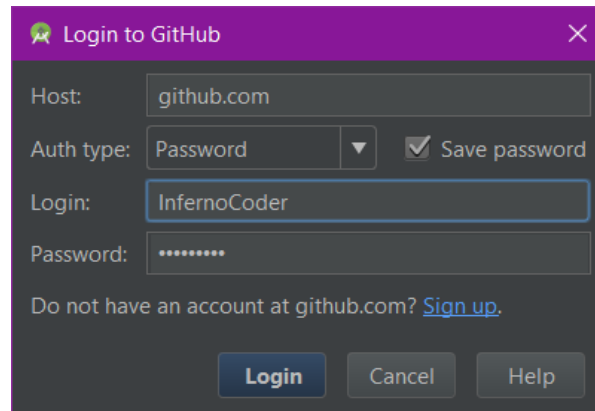
## **(B) Android Studio based approach -**

### **III. Clone the project to your local computer.**

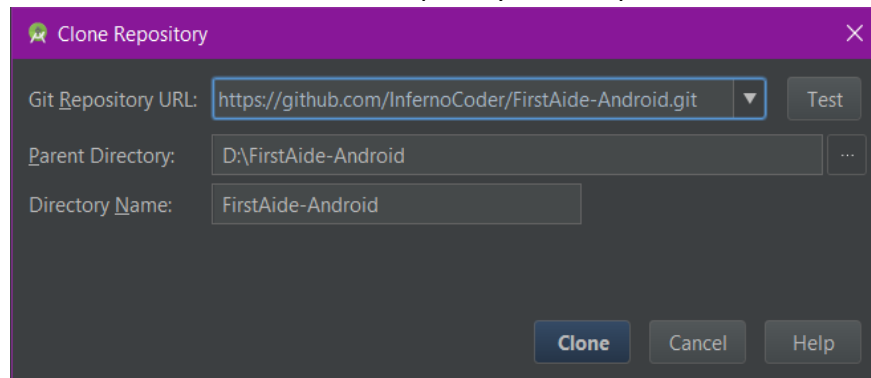
- i. Open Android Studio and close any existing projects through **File -> Close Project**. You will be shown the Android Studio Welcome screen.
- ii. Select the **“Check out project from Version Control option”** and choose **GitHub**.



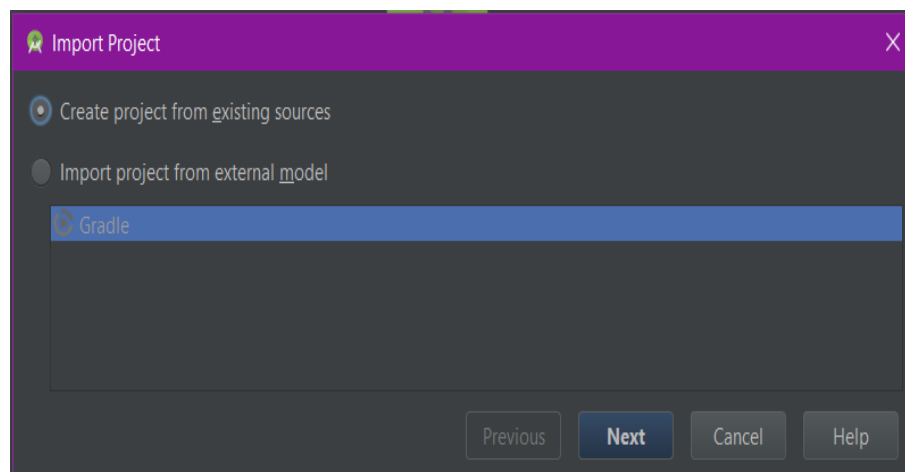
- iii. If you haven't signed in already, you will be asked to sign in to GitHub from Android Studio. Fill out your username and password and click on **“Login”**.



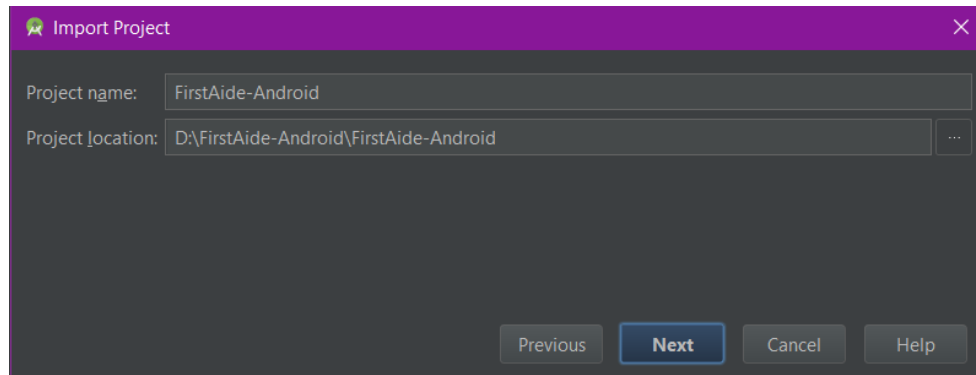
- iv. After signing in, you will be given a dialog to clone your preferred directory. Enter your fork in the **“Git Repository URL”** ([https://github.com/{YOUR\\_USERNAME}/FirstAide-Android.git](https://github.com/{YOUR_USERNAME}/FirstAide-Android.git)) and the directory in which you want to clone the repo in **“Parent Directory”**.
- v. It will take a while to clone the repo to your computer.



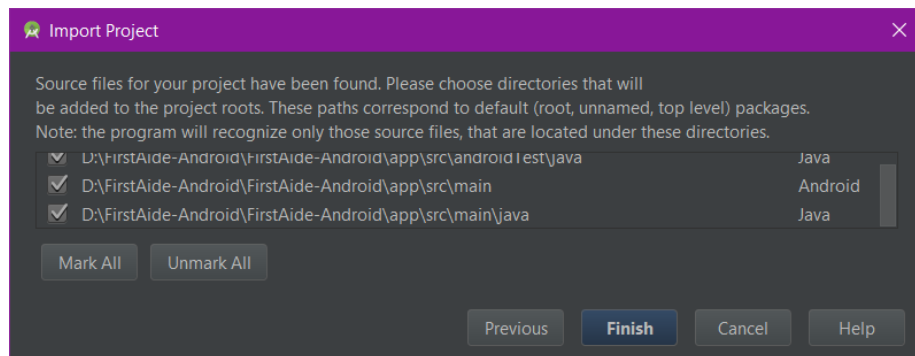
- vi. After cloning, it will ask how to import project. Choose **“Create project from existing sources”** and click **“Next”**.



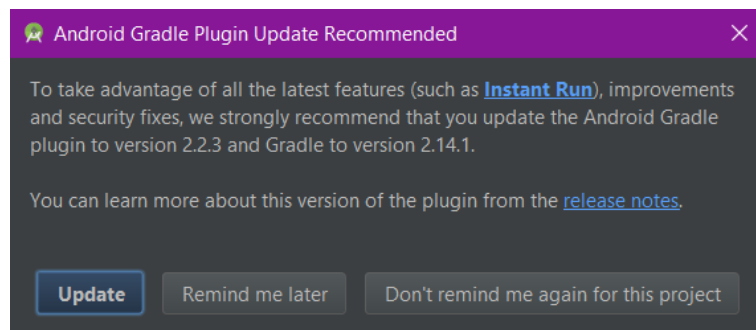
- vii. Don't change the project name as it may cause problems when uploading the changes to GitHub. So, click **"Next"** again.



- viii. In the final step, click on **"Mark all"** and then click on **"Finish"**.



**NOTE:** In case the below dialog is shown, click on **"Don't remind me again for this project"** as this will prevent from changing the project's Gradle version which may cause errors while uploading your changes to GitHub.



#### IV. Making the project ready for development

- Unfortunately, Android Studio does not provide a way to change the remote location. So, the below steps have to be carried out through command prompt.
- Start a command prompt in the folder where the project was cloned. To do that, refer to **Step III. -> iv. of Approach (A)**.



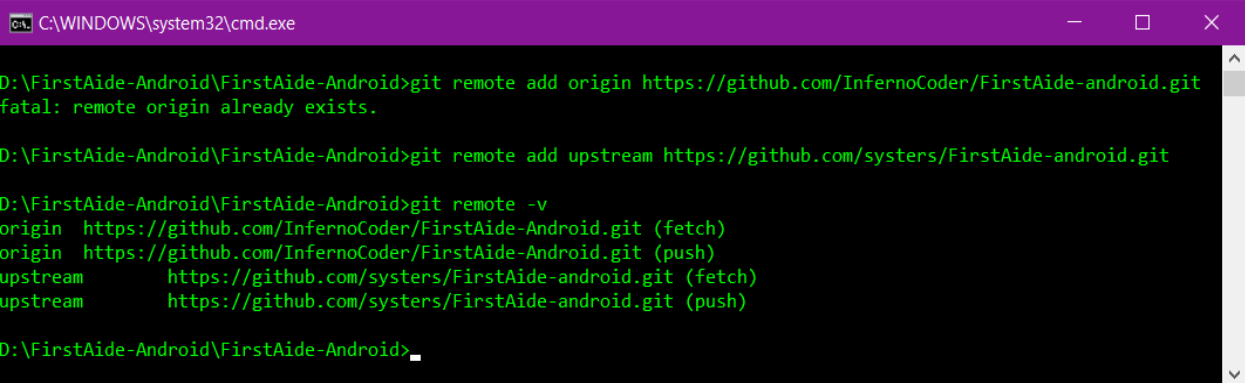
- iii. Set up the origin:

```
git remote add origin  
https://github.com/{YOUR_USERNAME}/FirstAide-android.git
```

- iv. Set the upstream:

```
git remote add upstream  
https://github.com/systers/FirstAide-android.git
```

- v. Run `git remote -v` to check the status. It should be something similar to the below picture.

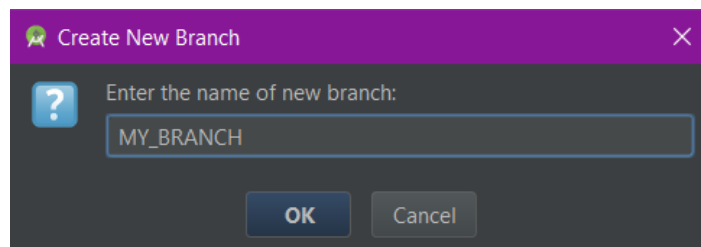


```
C:\WINDOWS\system32\cmd.exe  
D:\FirstAide-Android\FirstAide-Android>git remote add origin https://github.com/InfernoCoder/FirstAide-android.git  
fatal: remote origin already exists.  
D:\FirstAide-Android\FirstAide-Android>git remote add upstream https://github.com/systers/FirstAide-android.git  
D:\FirstAide-Android\FirstAide-Android>git remote -v  
origin https://github.com/InfernoCoder/FirstAide-Android.git (fetch)  
origin https://github.com/InfernoCoder/FirstAide-Android.git (push)  
upstream https://github.com/systers/FirstAide-android.git (fetch)  
upstream https://github.com/systers/FirstAide-android.git (push)  
D:\FirstAide-Android\FirstAide-Android>
```

- vi. Close the command prompt, and then open Android Studio.
- vii. Fetch the upstream in Android Studio through **VCS -> Git -> Fetch**.
- viii. Rebase your branch through **VCS -> Git -> Rebase**.

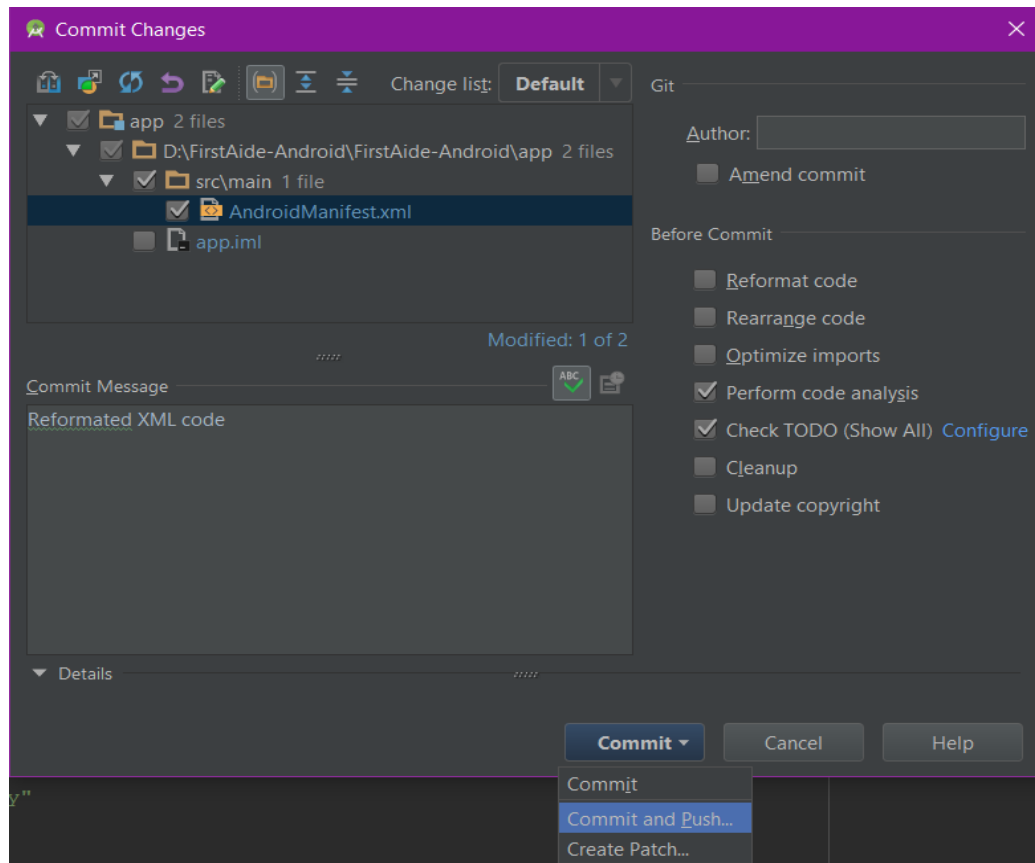
## V. Making new changes

- i. Checkout to a new branch in Android Studio through **VCS -> Git -> Branches -> New Branch** and then enter the name of a new branch.

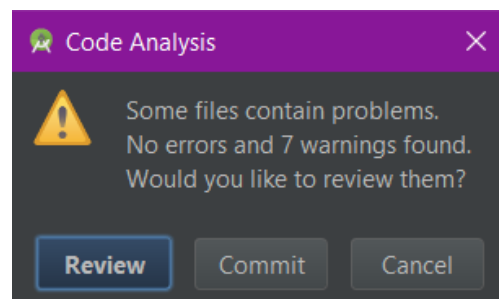


- ii. Make your changes as required in Android Studio.
- iii. After adding and testing the changes, clean the Project in Android Studio. See below picture for reference.

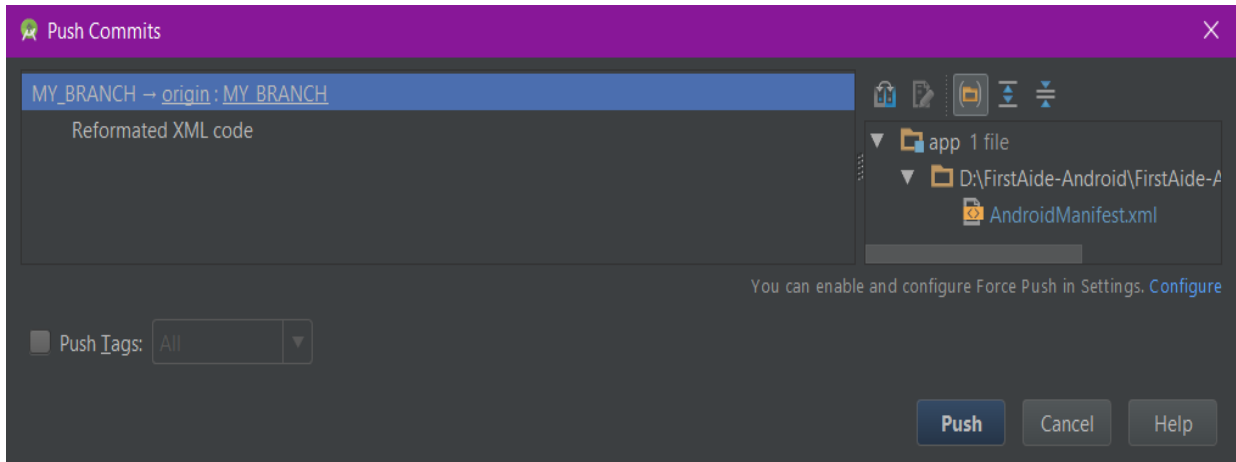
- iv. To add the changes, use **VCS -> Git -> Add (Shortcut– Ctrl+Alt+A)**.
  - v. To commit and push changes, use **VCS -> Commit Changes (Shortcut– Ctrl+K)**.
- Tick all the files to be changed and write a commit message. After doing this, **hover** on the “**Commit**” to get a drop-down menu and choose “**Commit and Push**” from the menu to push the changes to GitHub.



**NOTE:** If you get a Code Analysis menu like below, if warnings are found in previous commits (the changes NOT made by you), then ignore them and press “Commit” as uploading changes to fix those warnings in the same commit will create problems for reviewers to review your changes.



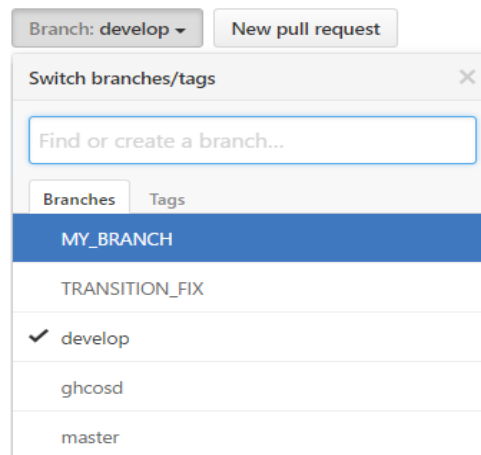
- vi. You will get a Push Commits window, click on **“Push”** button to continue.



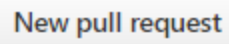
**Below is Step VI which is common for both approaches (A) and (B)**

**VI. Creating a Pull Request(PR) to submit the task.**

- i. Go to your forked repository on GitHub  
-[https://github.com/{YOUR\\_USERNAME}/FirstAide-Android](https://github.com/{YOUR_USERNAME}/FirstAide-Android).
- ii. Switch the branch from **‘develop’** to your pushed branch. Here, I am switching to **‘MY\_BRANCH’**.



- iii. Create pull request by clicking on the “New pull request” button (see below) (next to Branch Selection button, as seen in the step ii. picture).

A rectangular button with a light gray background and a thin border. The text "New pull request" is centered in a dark blue font.

- iv. Write the title and message of the pull request and then click on the “Create Pull Request button” (see below) to submit your change!

A rectangular button with a solid green background. The text "Create pull request" is centered in a white font.