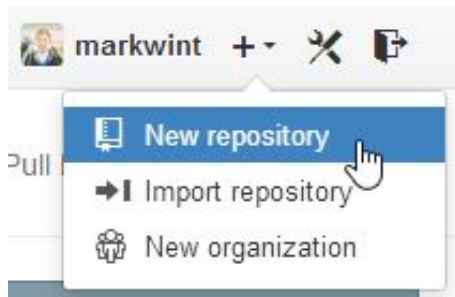


How to use GitHub with Android Studio


This article will explain how to use GitHub with Android Studio.

Firstly, let's login to github.com and create a new repository. From the top right corner of the screen, click the + sign and select *New repository*.




Now complete the form to create a new repository called *testproject*. Leave the *Initialize this repository with a README* unticked.

Owner

 markwint

Repository name

testproject 

Great repository names are short and memorable. Need inspiration? How about [fuzzy-computing-machine](#).

Description (optional)

☒ Public

Anyone can see this repository. You choose who can commit.


☐ Private

You choose who can see and commit to this repository.


☐ Initialize this repository with a README

This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

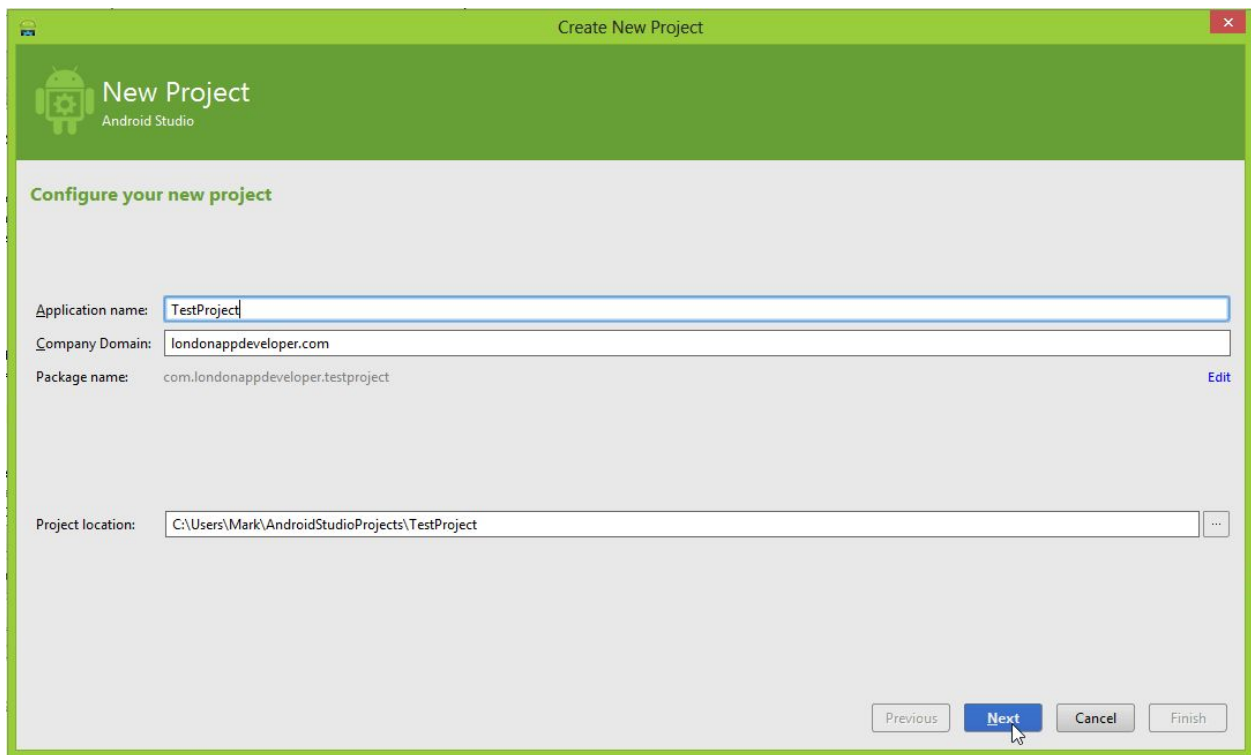
Add .gitignore: **None**

Add a license: **None** 

Create repository

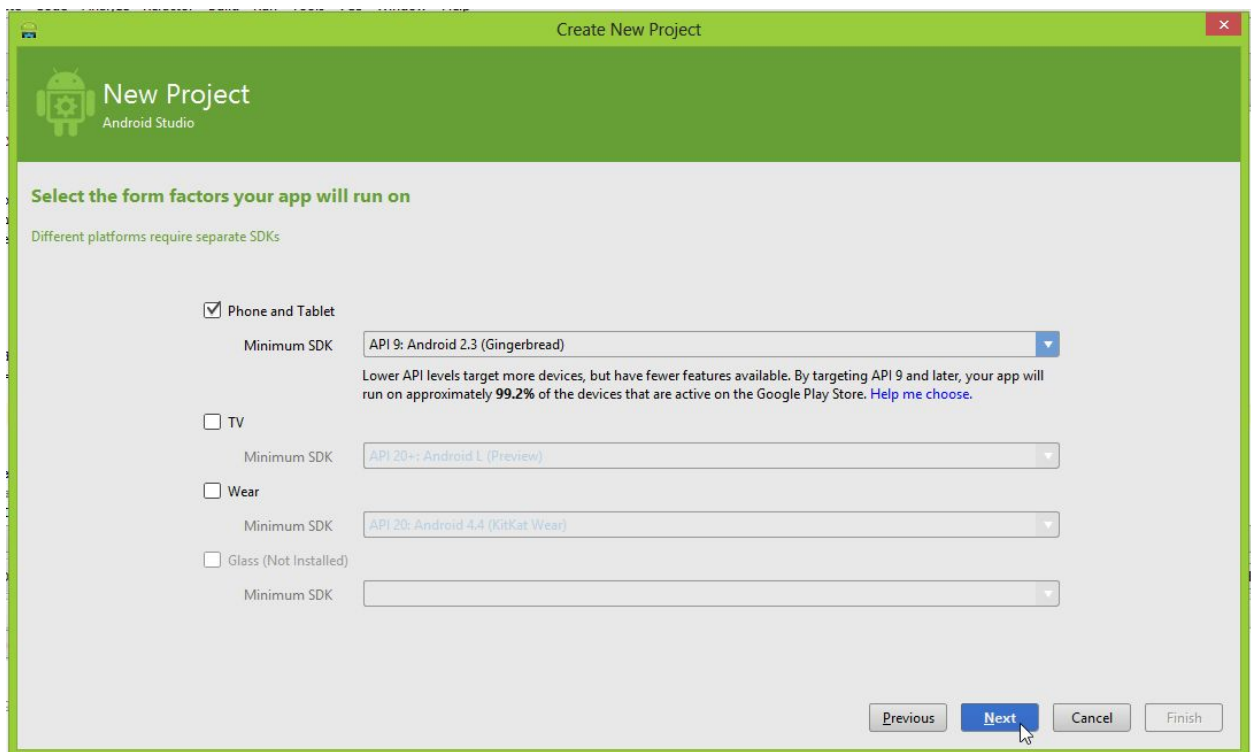


Next, open *Android Studio* and create a new project. Call your new application *TestProject*. Click *Next* to continue.



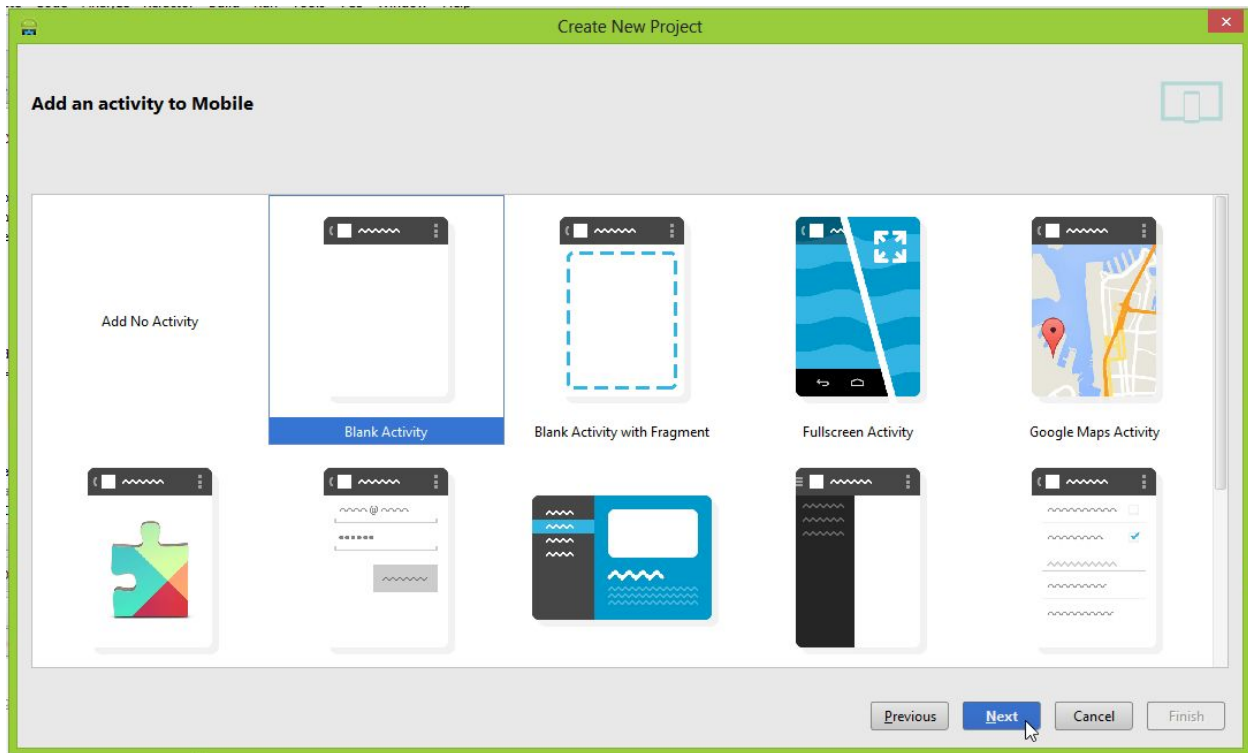
The screenshot shows the 'Create New Project' dialog in Android Studio. The title bar says 'Create New Project'. The main header is 'New Project' with the Android Studio logo. Below the header, the section is 'Configure your new project'. There are four input fields: 'Application name' with 'TestProject', 'Company Domain' with 'londonappdeveloper.com', 'Package name' with 'com.londonappdeveloper.testproject', and 'Project location' with 'C:\Users\Mark\AndroidStudioProjects\TestProject'. At the bottom right, there are four buttons: 'Previous', 'Next' (highlighted with a mouse cursor), 'Cancel', and 'Finish'.

Leave the next page as default and click *Next*.

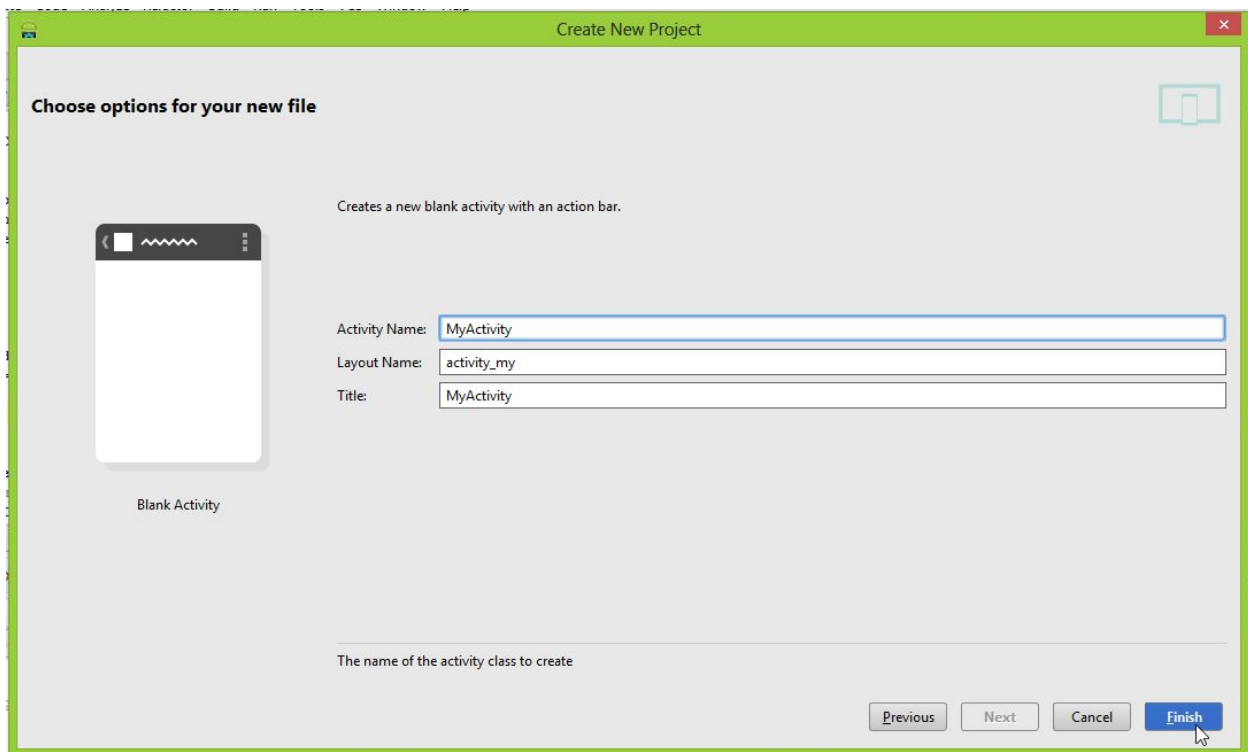


The screenshot shows the 'Create New Project' dialog in Android Studio, Step 2: 'Select the form factors your app will run on'. The title bar says 'Create New Project'. The main header is 'New Project' with the Android Studio logo. Below the header, the section is 'Select the form factors your app will run on'. There is a sub-header 'Different platforms require separate SDKs'. There are four options: 'Phone and Tablet' (checked), 'TV', 'Wear', and 'Glass (Not Installed)'. Each option has a 'Minimum SDK' dropdown menu. For 'Phone and Tablet', the dropdown shows 'API 9: Android 2.3 (Gingerbread)'. Below the dropdowns, there is a note: 'Lower API levels target more devices, but have fewer features available. By targeting API 9 and later, your app will run on approximately 99.2% of the devices that are active on the Google Play Store. [Help me choose.](#)'. At the bottom right, there are four buttons: 'Previous', 'Next' (highlighted with a mouse cursor), 'Cancel', and 'Finish'.

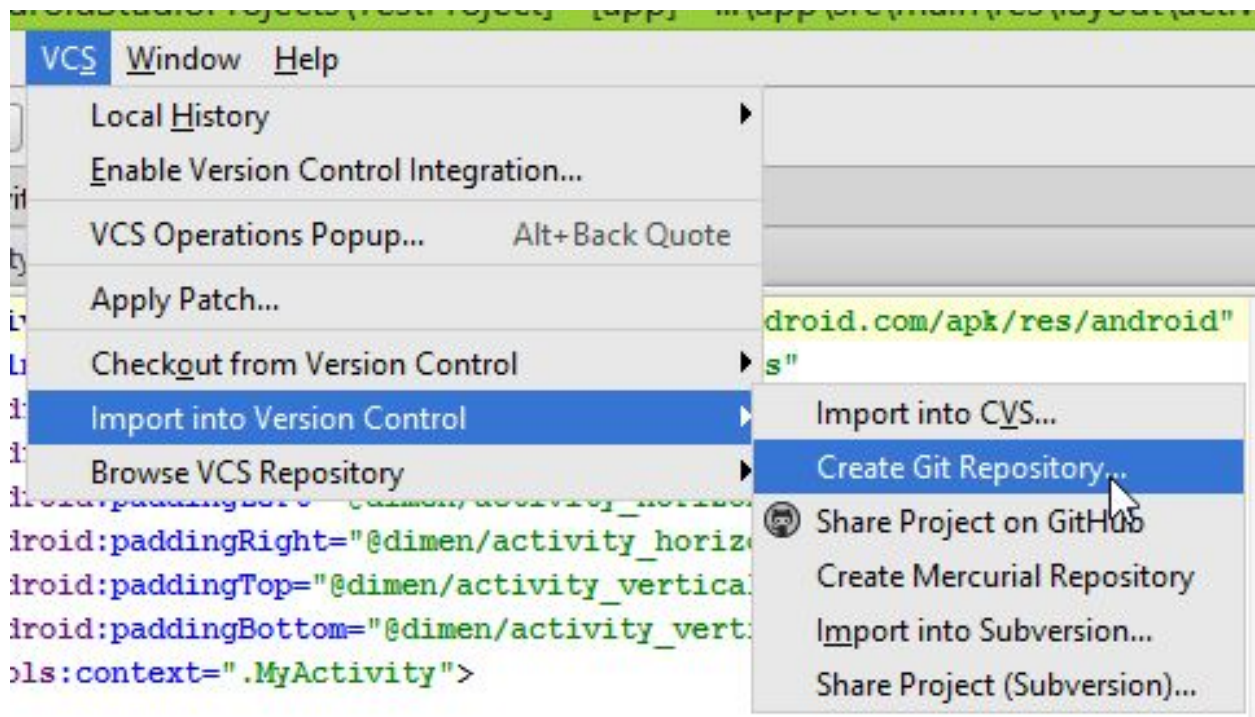
On the *Add an activity to Mobile* screen select *Blank Activity* and click *Next*.



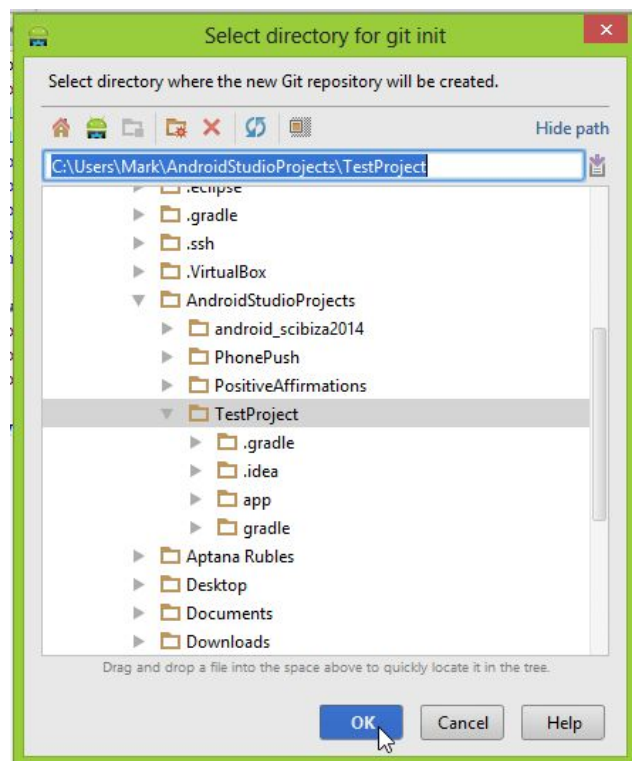
In the next screen, leave the default activity name of *MyActivity* and click *Finish*.



Your new project will open in Android Studio. On the top menu, select **VCS > Import into Version Control > Create Git Repository**.



On the next screen, leave it all as default and click OK.

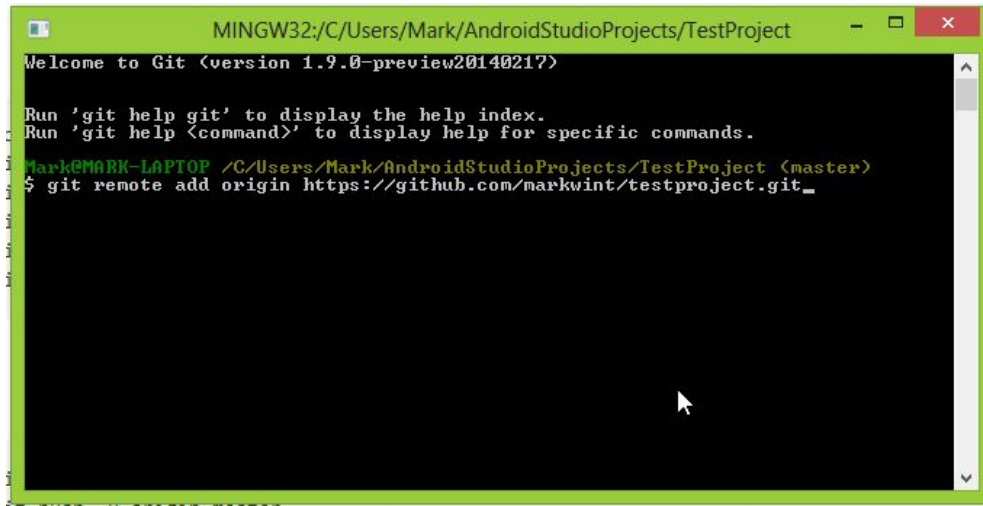


Now use Windows Explorer and navigate to the root of your projects folder. Right click and select *Git Bash* (If you do not see this option, then first install [Git for Windows](#)).

When the Git bash screen appears, type:

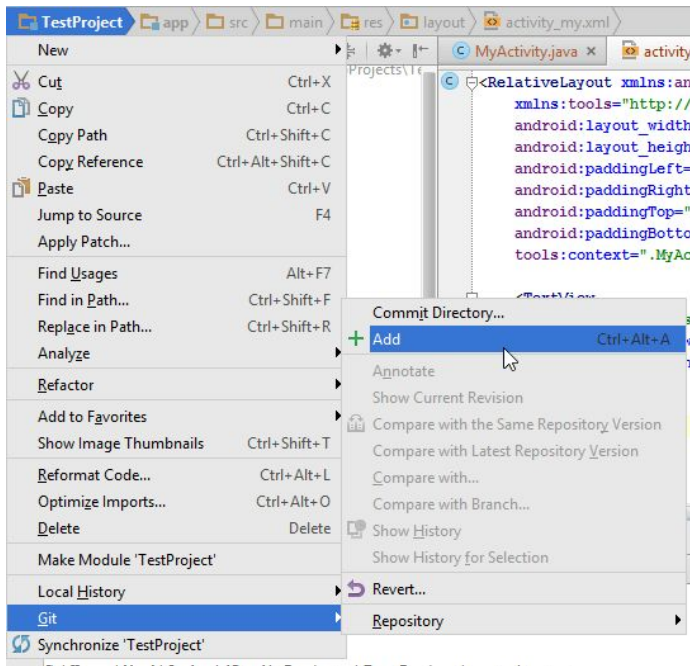
```
1 git remote add origin https://github.com/[username]/[project_name].git
```

An example of a Git repository URL is: <https://github.com/markwint/testproject.git>



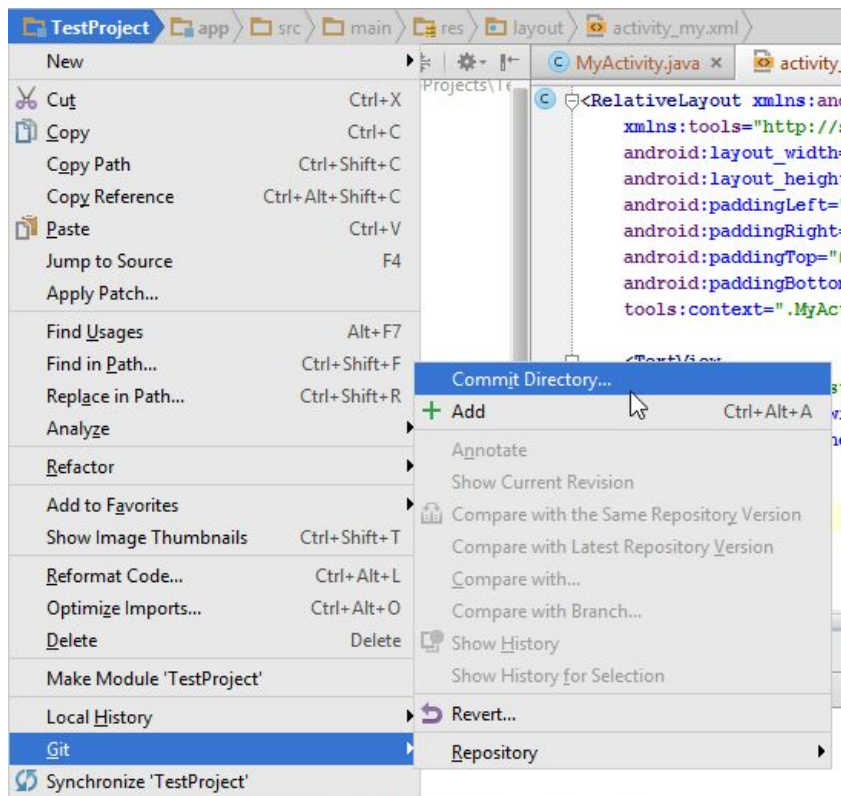
Then press enter. The GitHub remote will be added to your Git repository.

Next, jump back into Android Studio, right click your projects root directory and select *Git > Add*. This will add all your project files to your Git repository.

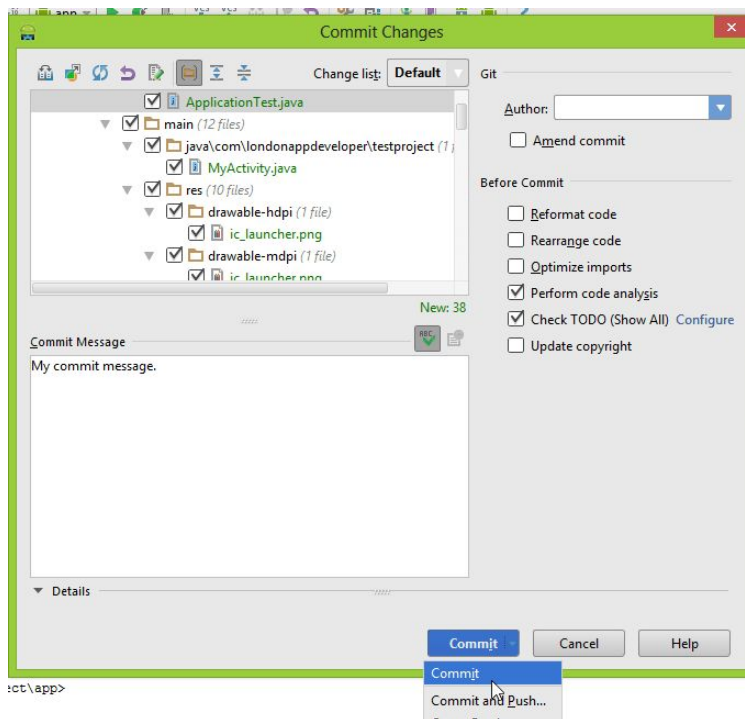


It will seem like nothing has happened, but trust me, the project files are added.

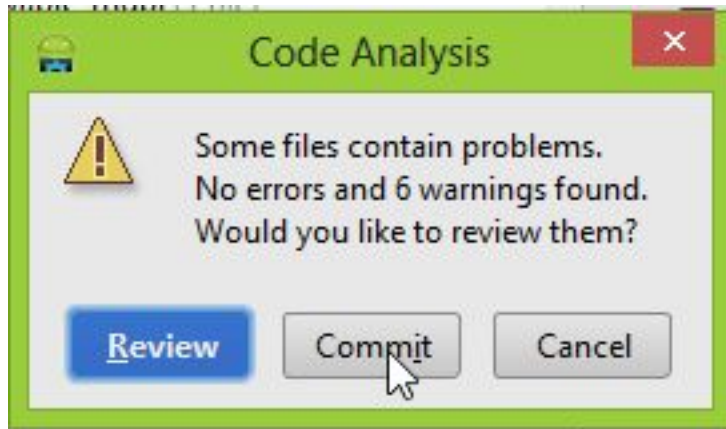
Now right click the project name again and this time select *Git > Commit Directory*.



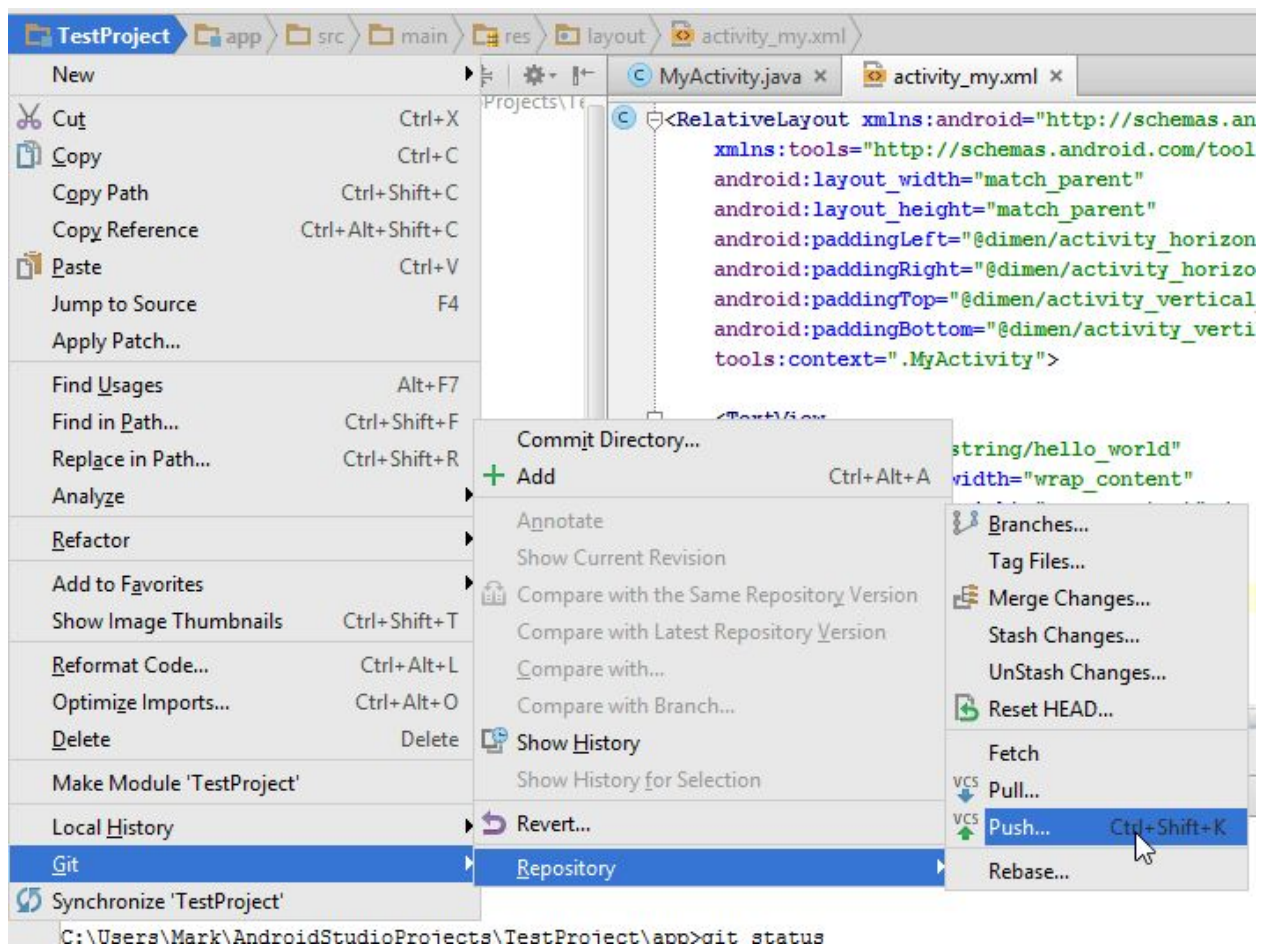
In the next screen, type a *Commit Message* and select *Commit*.



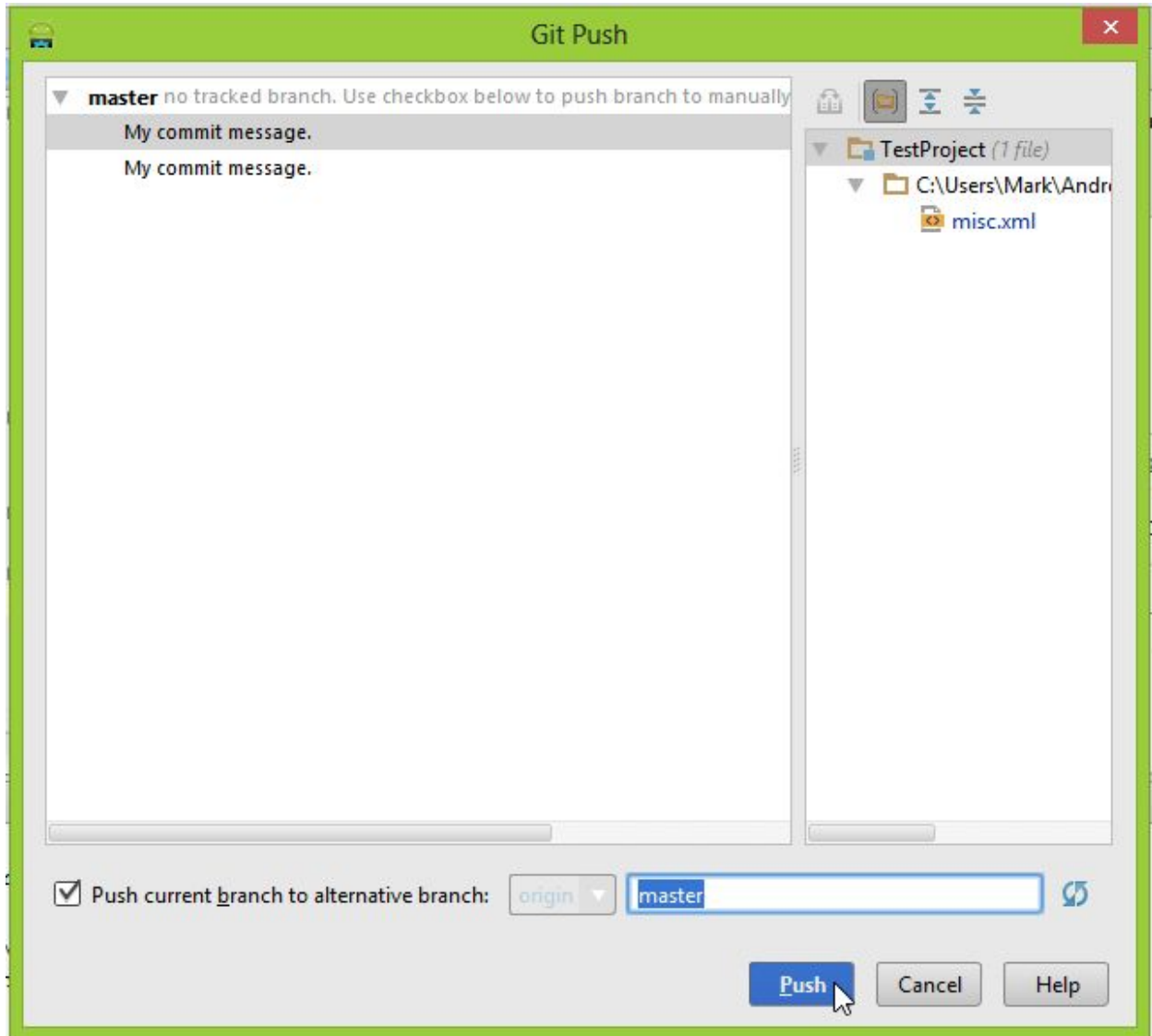
If a Code Analysis warning appears, click *Commit*. (Unless it's a real project, in which case review and fix the issue before committing!)



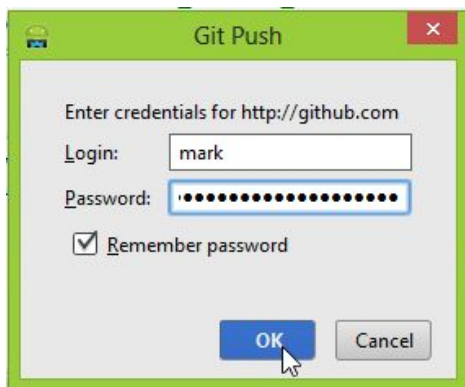
Now, right click the project name, select *Git > Repository > Push*.



Check the box *Push current branch to alternative branch* and leave the branch name as *master*. Then select *push*.



Now enter your GitHub *Login* (email address) and *Password*. Then click *OK*.



If it's all good, you will see this message.

Push successful
master pushed to new branch origin/master

Now your code is pushed to your GitHub repository. Don't believe me? Logon and check for yourself.

My commit message.

markwint authored 3 minutes ago latest commit d744568c26

.idea	My commit message.	3 minutes ago
app	My commit message.	6 minutes ago
gradle	My commit message.	6 minutes ago
.gitignore	My commit message.	6 minutes ago
TestProject.iml	My commit message.	6 minutes ago
build.gradle	My commit message.	6 minutes ago
gradle.properties	My commit message.	6 minutes ago
gradlew	My commit message.	6 minutes ago
gradlew.bat	My commit message.	6 minutes ago
settings.gradle	My commit message.	6 minutes ago

We recommend [adding a README](#) to this repository to help give people an overview of your project. [Add a README](#)

And that's it! Thanks for reading. I hope you found this useful. As always, if you have any questions please drop me a line at hello@londonappdeveloper.com

AND WHY NOT:

Follow me on twitter [@londonappdev](https://twitter.com/londonappdev)

Subscribe to me on [Facebook](#)

Check out my videos on [YouTube](#)

See my work on [GitHub](#)