Mobile Application Programming SCSJ3623

Semester 2, 2021/2022

Outline

- About the course:
    - Course Outline
    - Assessment
    - Group project
- Mobile App. Dev Technologies
- Installation: the framework and tools
- Test drive on emulator and phone
- A tour on VS Code, and Android Studio
- A tour on Git and Git Bash

Group Project >> The groups

- Group of 4.
- Form your own group members.
- From the same section

Group Project >> Requirements

Project Features:

- Authentication: login, logout, etc.
- Personalization: at least two types of users
- CRUD operations
- Push notification
- Reporting

Group Project >> Requirements (2)

Front-end:

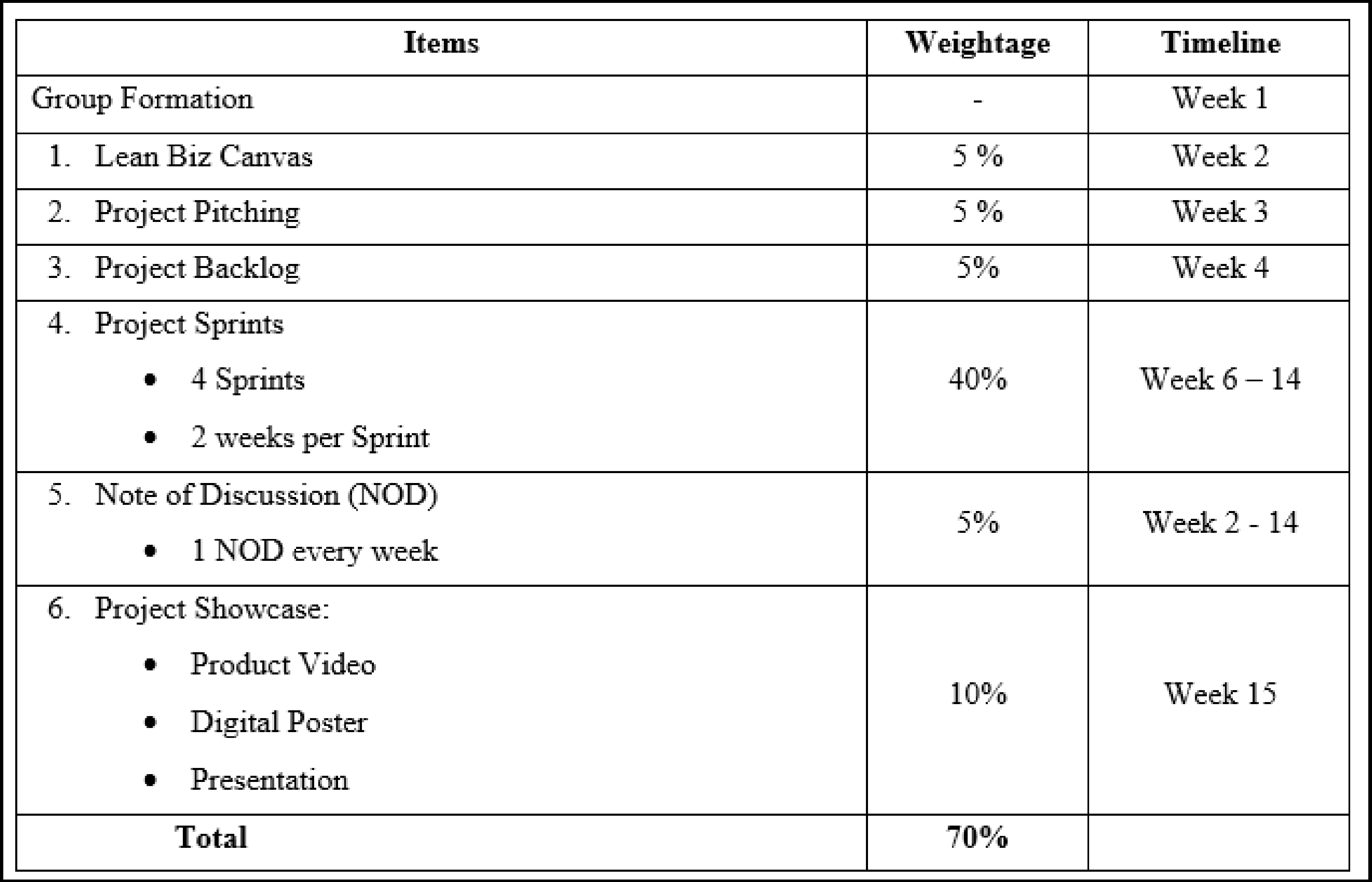
- Use Flutter Framework
- Adopt MVVM architecture

Backend-end:

- Use Firebase and use services:
  - Authentication
  - Database (e.g. Firestore)
  - Cloud Storage
  - Push Notification
  - Security Rules
- Server-side code is optional or minimal

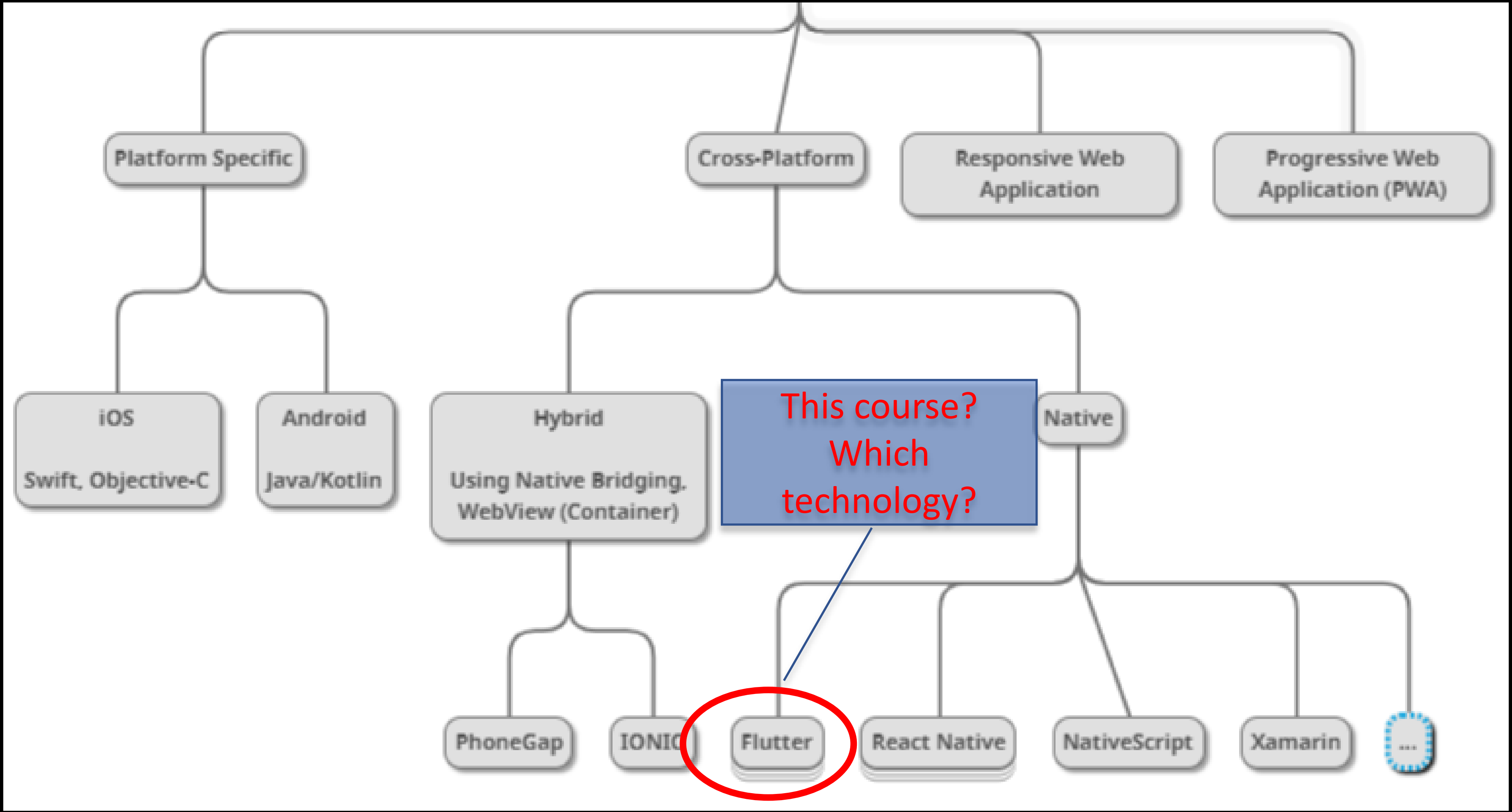
- Use Git and GitHub
  - Versioning and collaboration – Git and Github Repository
  - Project Management - Github Board

Group Project >> Deliverables



Items	Weightage	Timeline
Group Formation	-	Week 1
1. Lean Biz Canvas	5 %	Week 2
2. Project Pitching	5 %	Week 3
3. Project Backlog	5%	Week 4
4. Project Sprints • 4 Sprints • 2 weeks per Sprint	40%	Week 6 – 14
5. Note of Discussion (NOD) • 1 NOD every week	5%	Week 2 - 14
6. Project Showcase: • Product Video • Digital Poster • Presentation	10%	Week 15
Total	**70%**	

Mobile App Dev. Technologies



What is Flutter?

- A UI Framework for building native application
- Cross-platform: Android, iOS, Desktop (Windows, MacOS, Linux), Web
- Made by Google
- Open-source
- Use Dart as the programming language
- More on flutter.dev

Install Flutter

Install Flutter on Windows

https://youtu.be/T9LdScRVhv8

Notes:
besides installing flutter, the video also shows you how to setup Flutter for Windows Desktop Development. You can skip this step as we are going to use Mobile Development.

Install Flutter on macOS

https://youtu.be/9GuzMsZQUYs

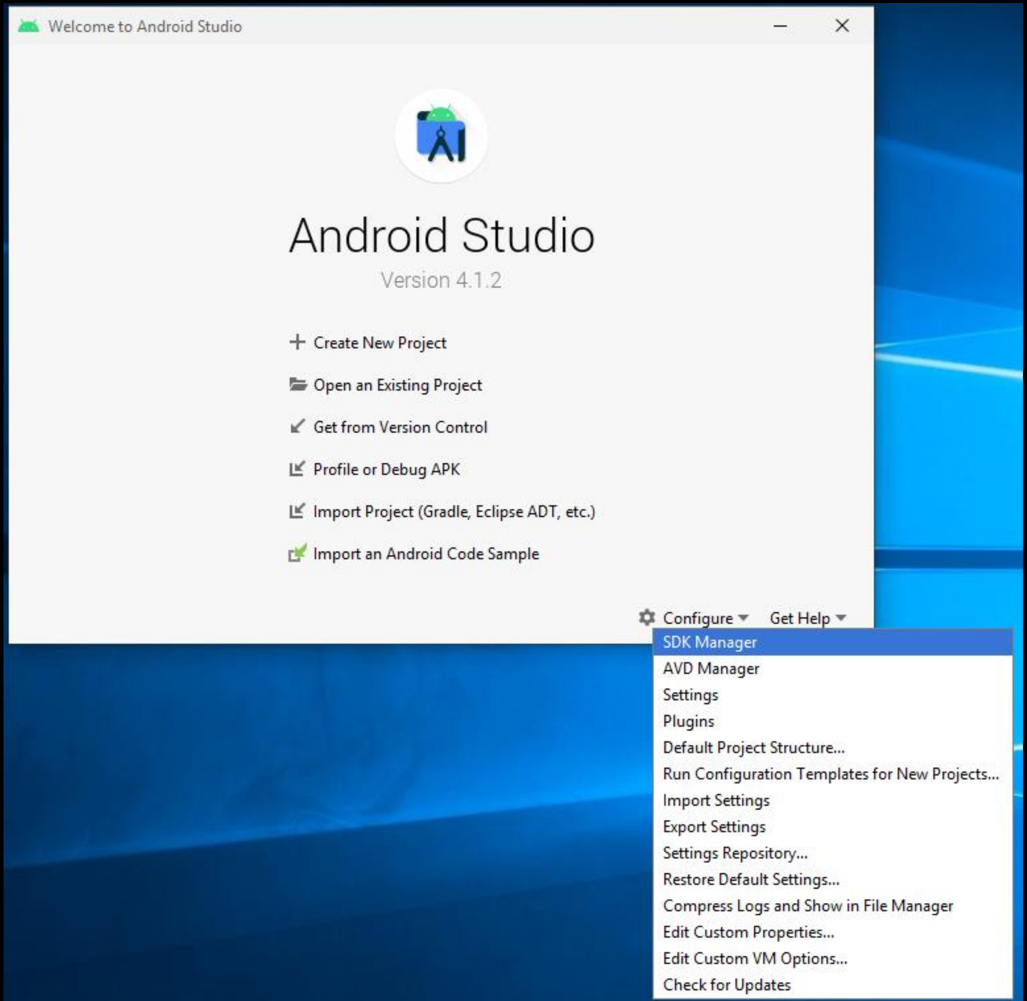
Notes:
besides installing flutter, the video also shows you how to setup Flutter for macOS Desktop Development. You can skip this step as we are going to use Mobile Development.

VS Code Extensions

	VS Code Extension	Extension ID
	Flutter	Dart-Code.flutter
	Live Share	ms-vsliveshare.vsliveshare
	Error Lens	PhilHindle.errorlens
	Pubspec Assist	jeroen-meijer.pubspec-assist
	Colonize	vmsynkov.colonize
	Better Comments	aaron-bond.better-comments
	Firebase	toba.vsfire
	REST Client	humao.rest-client
	Peacock	johnpapa.vscode-peacock

Configure Android SDK

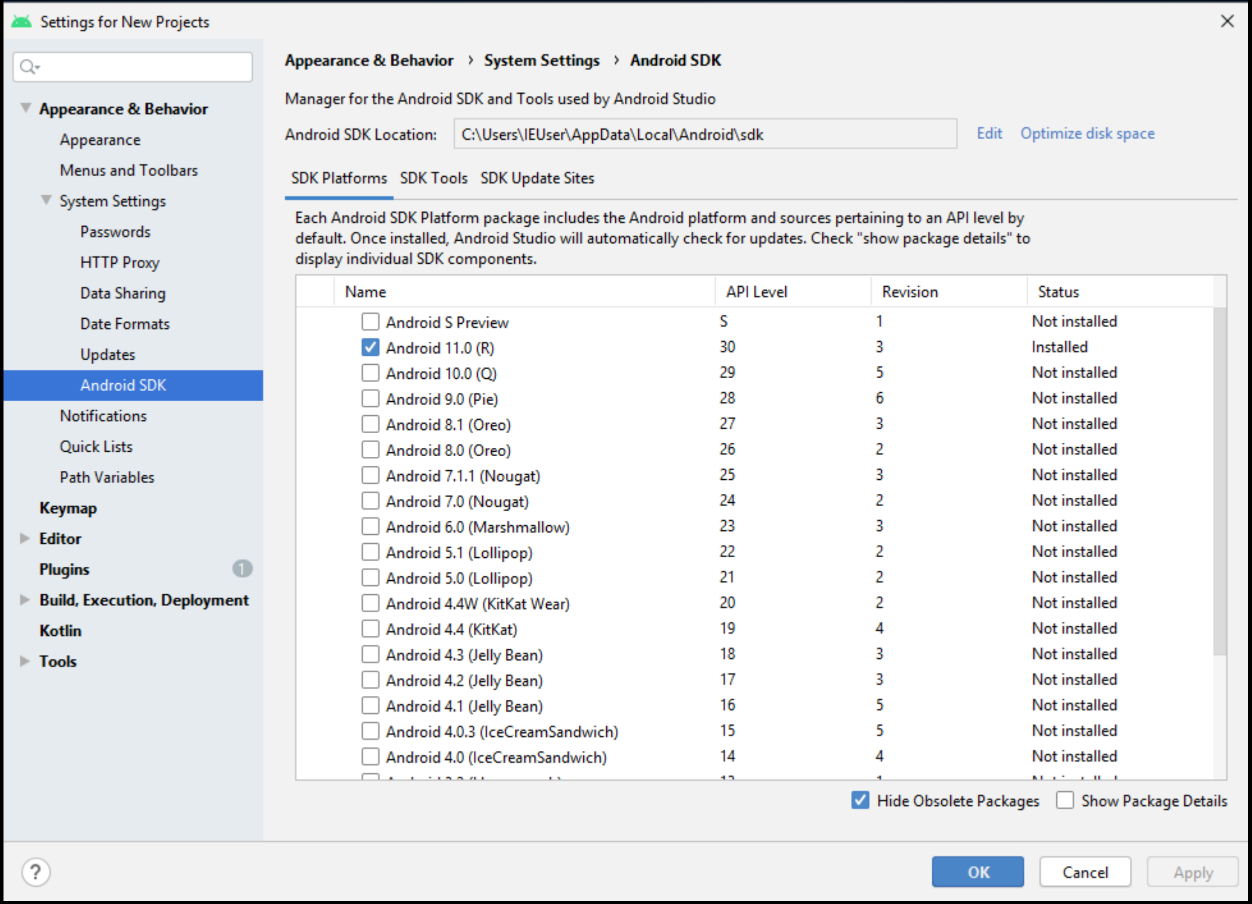
Run Android Studio (AS)
Inside AS, run SDK Manager



Welcome to Android Studio
Android Studio
Version 4.1.2
Create New Project
Open an Existing Project
Get from Version Control
Profile or Debug APK
Import Project (Gradle, Eclipse ADT, etc.)
Import an Android Code Sample
Configure
Get Help
SDK Manager
AVD Manager
Settings
Plugins
Default Project Structure...
Run Configuration Templates for New Projects...
Import Settings
Export Settings
Settings Repository...
Restore Default Settings...
Compress Logs and Show in File Manager
Edit Custom Properties...
Edit Custom VM Options...
Check for Updates

Configure Android SDK

SDK Platform



Settings for New Projects

- Appearance & Behavior
  - Appearance
  - Menus and Toolbars
  - System Settings
    - Passwords
    - HTTP Proxy
    - Data Sharing
    - Date Formats
    - Updates
    - Android SDK
  - Notifications
  - Quick Lists
  - Path Variables
- Keymap
- Editor
- Plugins
- Build, Execution, Deployment
- Kotlin
- Tools

Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\IEUser\AppData\Local\Android\sdk   Edit   Optimize disk space

SDK Platforms | SDK Tools | SDK Update Sites

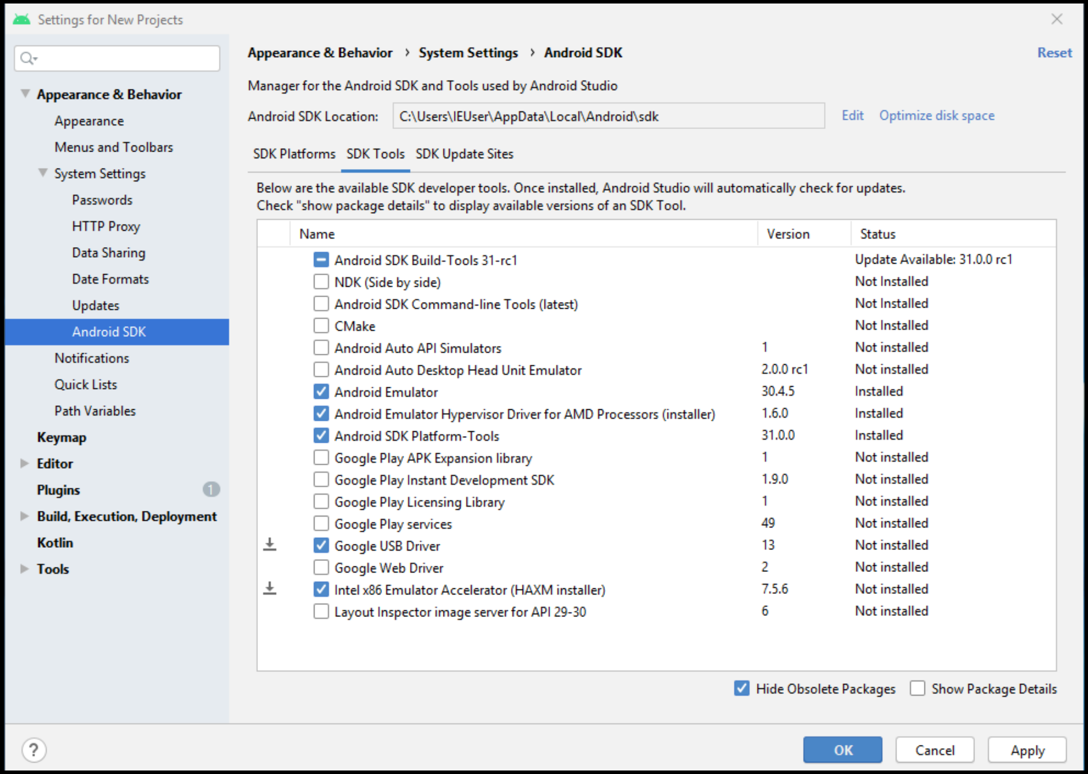
Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

Name	API Level	Revision	Status
☐ Android S Preview	S	1	Not installed
☑ Android 11.0 (R)	30	3	Installed
☐ Android 10.0 (Q)	29	5	Not installed
☐ Android 9.0 (Pie)	28	6	Not installed
☐ Android 8.1 (Oreo)	27	3	Not installed
☐ Android 8.0 (Oreo)	26	2	Not installed
☐ Android 7.1.1 (Nougat)	25	3	Not installed
☐ Android 7.0 (Nougat)	24	2	Not installed
☐ Android 6.0 (Marshmallow)	23	3	Not installed
☐ Android 5.1 (Lollipop)	22	2	Not installed
☐ Android 5.0 (Lollipop)	21	2	Not installed
☐ Android 4.4W (KitKat Wear)	20	2	Not installed
☐ Android 4.4 (KitKat)	19	4	Not installed
☐ Android 4.3 (Jelly Bean)	18	3	Not installed
☐ Android 4.2 (Jelly Bean)	17	3	Not installed
☐ Android 4.1 (Jelly Bean)	16	5	Not installed
☐ Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
☐ Android 4.0 (IceCreamSandwich)	14	4	Not installed

☑ Hide Obsolete Packages   ☐ Show Package Details

Configure Android SDK

SDK Tools



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\IEUser\AppData\Local\Android\sdk

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates.
Check "show package details" to display available versions of an SDK Tool.

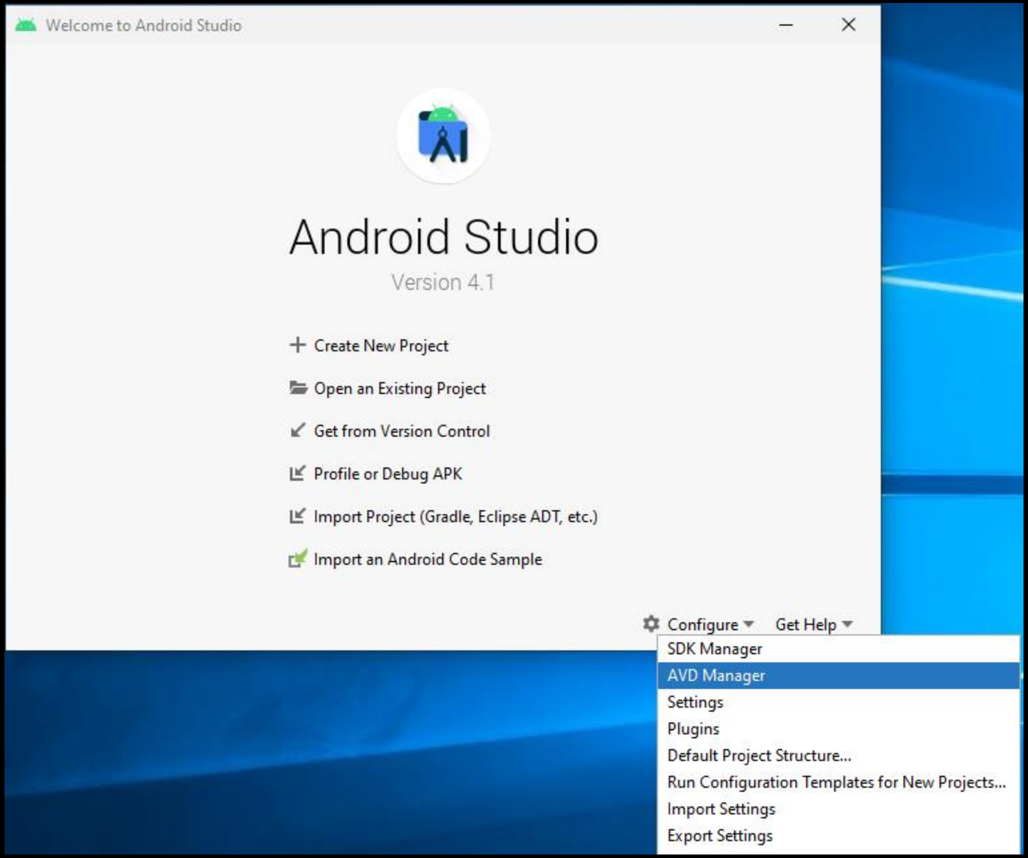
	Name	Version	Status
	▬ Android SDK Build-Tools 31-rc1		Update Available: 31.0.0 rc1
	☐ NDK (Side by side)		Not Installed
	☐ Android SDK Command-line Tools (latest)		Not Installed
	☐ CMake		Not Installed
	☐ Android Auto API Simulators	1	Not installed
	☐ Android Auto Desktop Head Unit Emulator	2.0.0 rc1	Not installed
	✓ Android Emulator	30.4.5	Installed
	✓ Android Emulator Hypervisor Driver for AMD Processors (installer)	1.6.0	Installed
	✓ Android SDK Platform-Tools	31.0.0	Installed
	☐ Google Play APK Expansion library	1	Not installed
	☐ Google Play Instant Development SDK	1.9.0	Not installed
	☐ Google Play Licensing Library	1	Not installed
	☐ Google Play services	49	Not installed
⤓	✓ Google USB Driver	13	Not installed
	☐ Google Web Driver	2	Not installed
⤓	✓ Intel x86 Emulator Accelerator (HAXM installer)	7.5.6	Not installed
	☐ Layout Inspector image server for API 29-30	6	Not installed

✓ Hide Obsolete Packages ☐ Show Package Details

Tick Intel x86 Emulator Accelerator, for Intel machine,
Tick Emulator Hypervisor Driver for AMD Processors, for AMD machine

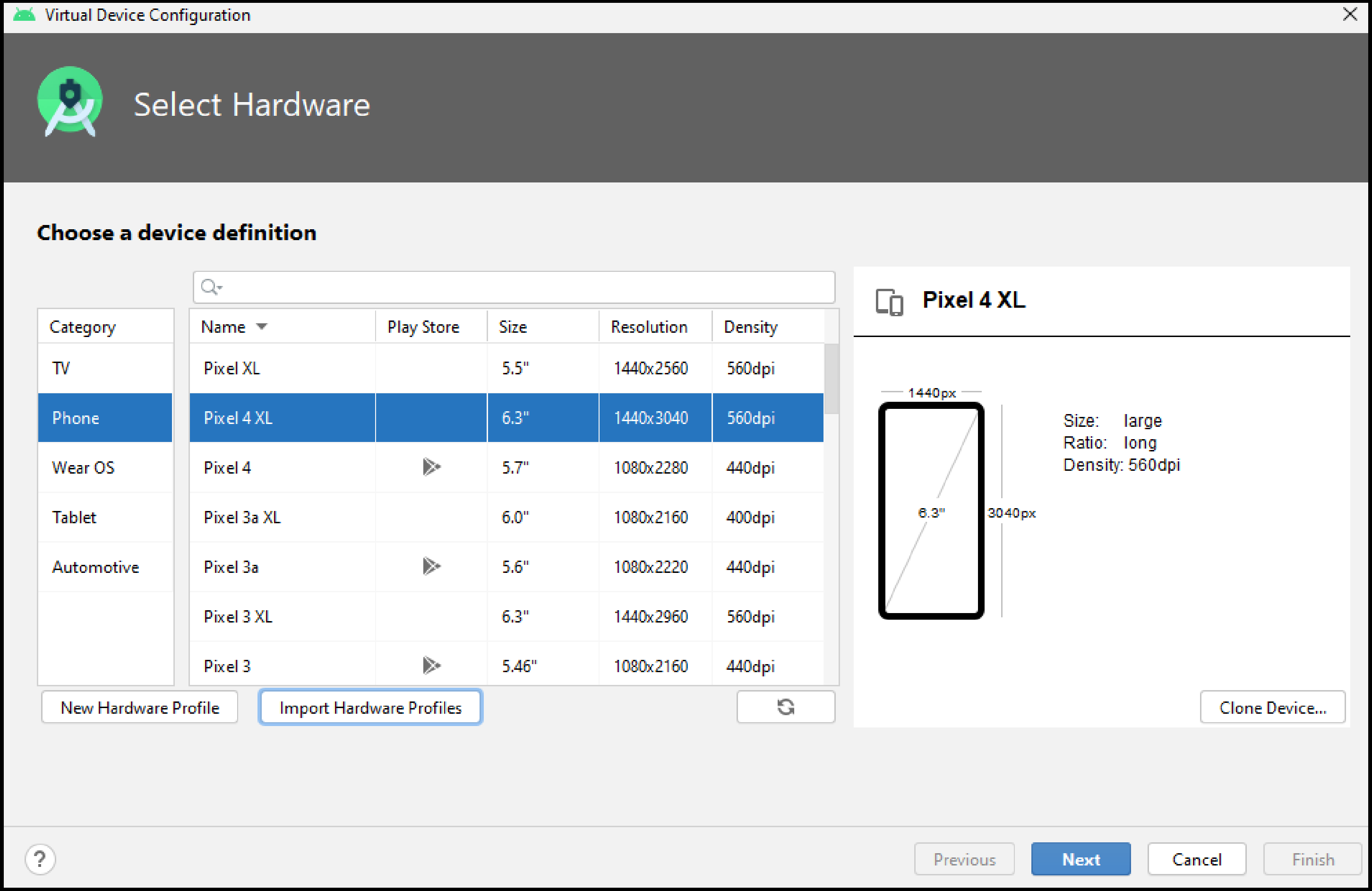
Create Android Emulator

Inside AS, run AVD Manager



Create Android Emulator

+ Create Virtual Device ...



Virtual Device Configuration

Select Hardware

Choose a device definition

Category
TV
Phone
Wear OS
Tablet
Automotive

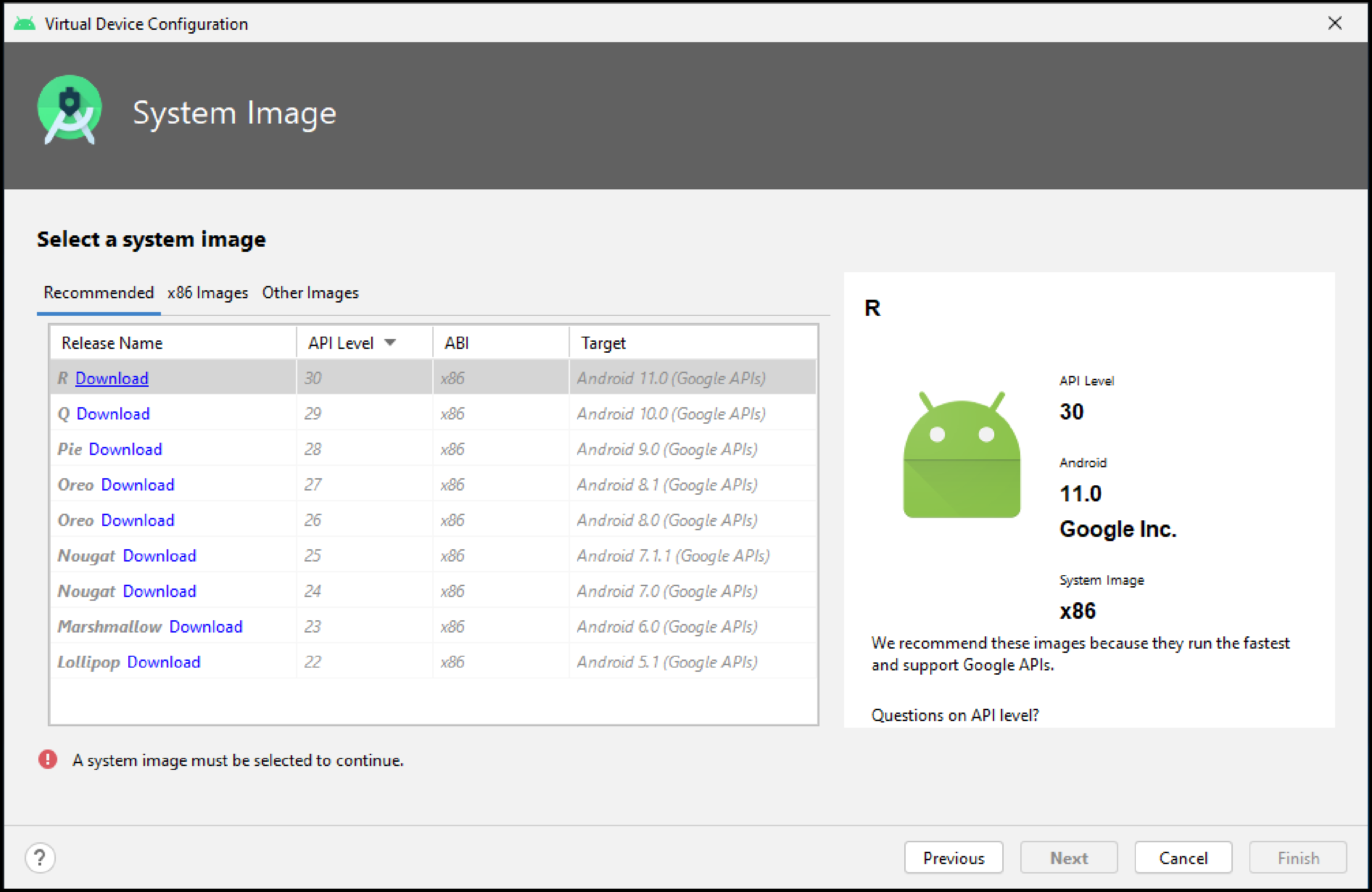
Name	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2560	560dpi
Pixel 4 XL		6.3"	1440x3040	560dpi
Pixel 4	▶	5.7"	1080x2280	440dpi
Pixel 3a XL		6.0"	1080x2160	400dpi
Pixel 3a	▶	5.6"	1080x2220	440dpi
Pixel 3 XL		6.3"	1440x2960	560dpi
Pixel 3	▶	5.46"	1080x2160	440dpi

Pixel 4 XL

Size: large
Ratio: long
Density: 560dpi

Create Android Emulator

Download system image



Virtual Device Configuration

System Image

Select a system image

Recommended x86 Images Other Images

Release Name	API Level	ABI	Target
R Download	30	x86	Android 11.0 (Google APIs)
Q Download	29	x86	Android 10.0 (Google APIs)
Pie Download	28	x86	Android 9.0 (Google APIs)
Oreo Download	27	x86	Android 8.1 (Google APIs)
Oreo Download	26	x86	Android 8.0 (Google APIs)
Nougat Download	25	x86	Android 7.1.1 (Google APIs)
Nougat Download	24	x86	Android 7.0 (Google APIs)
Marshmallow Download	23	x86	Android 6.0 (Google APIs)
Lollipop Download	22	x86	Android 5.1 (Google APIs)

R

API Level
30

Android
11.0
Google Inc.

System Image
x86

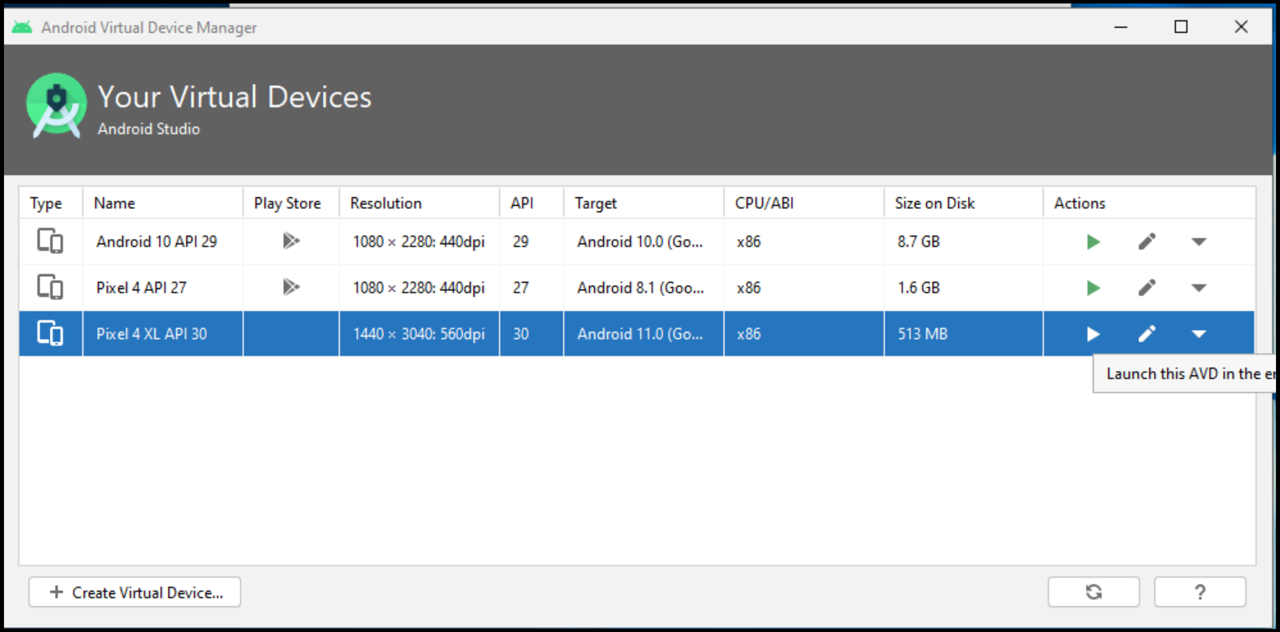
We recommend these images because they run the fastest and support Google APIs.

Questions on API level?

A system image must be selected to continue.

Test the Emulator

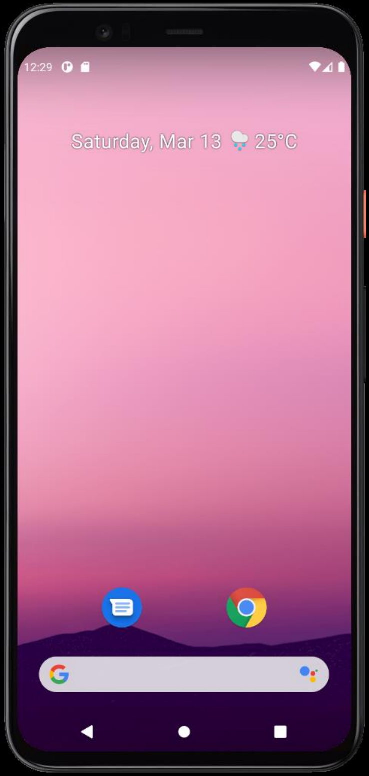
Run the Emulator from AVD Manager


Android Virtual Device Manager

Your Virtual Devices
Android Studio

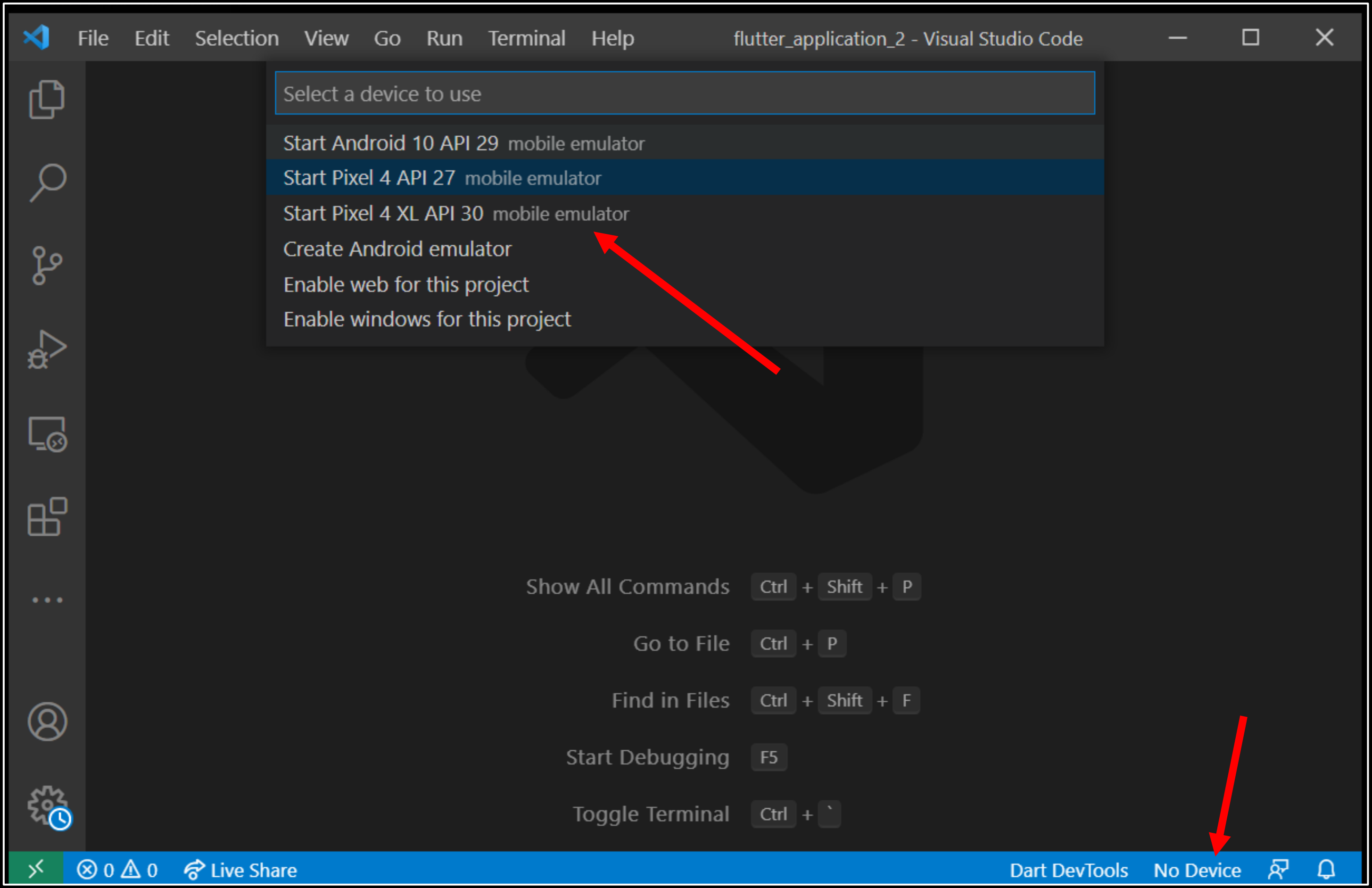
Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Android 10 API 29	▶	1080 × 2280: 440dpi	29	Android 10.0 (Go...	x86	8.7 GB	
	Pixel 4 API 27	▶	1080 × 2280: 440dpi	27	Android 8.1 (Goo...	x86	1.6 GB	
	Pixel 4 XL API 30		1440 × 3040: 560dpi	30	Android 11.0 (Go...	x86	513 MB	

Launch this AVD in the er



Test the Emulator

Run the Emulator directly from VS Code



flutter_application_2 - Visual Studio Code

Select a device to use

Start Android 10 API 29 mobile emulator
Start Pixel 4 API 27 mobile emulator
Start Pixel 4 XL API 30 mobile emulator
Create Android emulator
Enable web for this project
Enable windows for this project

A Tour on VS Code

- Command Pallet: Ctrl Shift P
- Copy code: Shift Alt Arrow (up or down)
- Move code: Alt Arrow (up or down)
- Comment: Ctrl /
- Split Code Editor: Ctrl Alt Left / Right

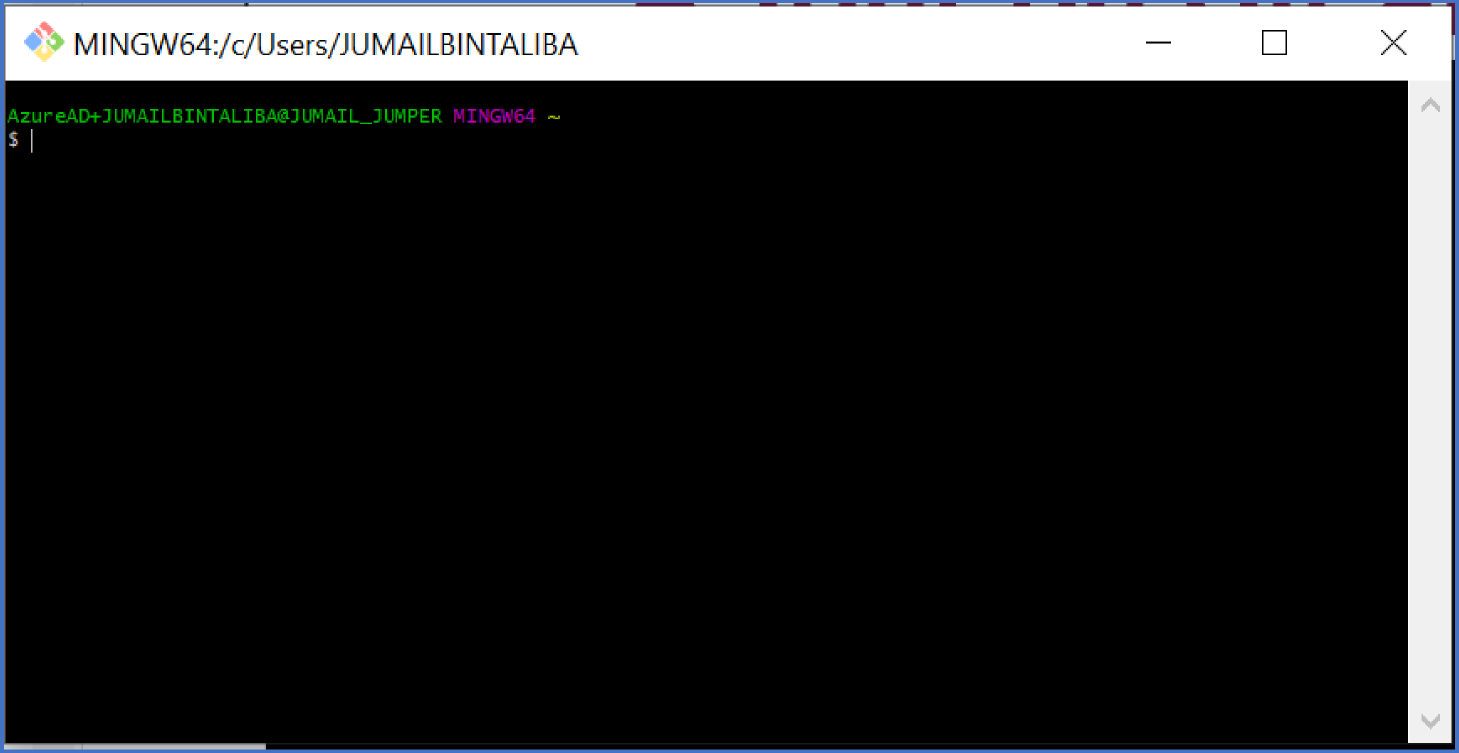
Test the Installation

- Open VS Code
- Go to Command Pallete.. Ctrl Shift P
- Type: flutter New Project
- Run / Start an emulator (from VS Code) or (from Android Studio (AS) Avd Manager)
- Run your first flutter program. Press F5

Test the Installation

- Next, to test running on a real device
- Configure your phone to "Developer Mode"
  https://developer.android.com/studio/debug/dev-options
- Connect your phone to the PC
- Choose your phone on VS Code
- Run your flutter project. Press F5
- To cast your phone to PC, this is an example app you can use: Letsview
  https://youtu.be/HPFhFbw4J-c

A Tour on Git and Git Bash



```
AzureAD+JUMAILBINTALIBA@JUMAIL_JUMPER MINGW64 ~
$ |
```

Common Unix Commands

Move to a directory and check out the content

```
$ cd c:/
$ ls
```

Create a new directory

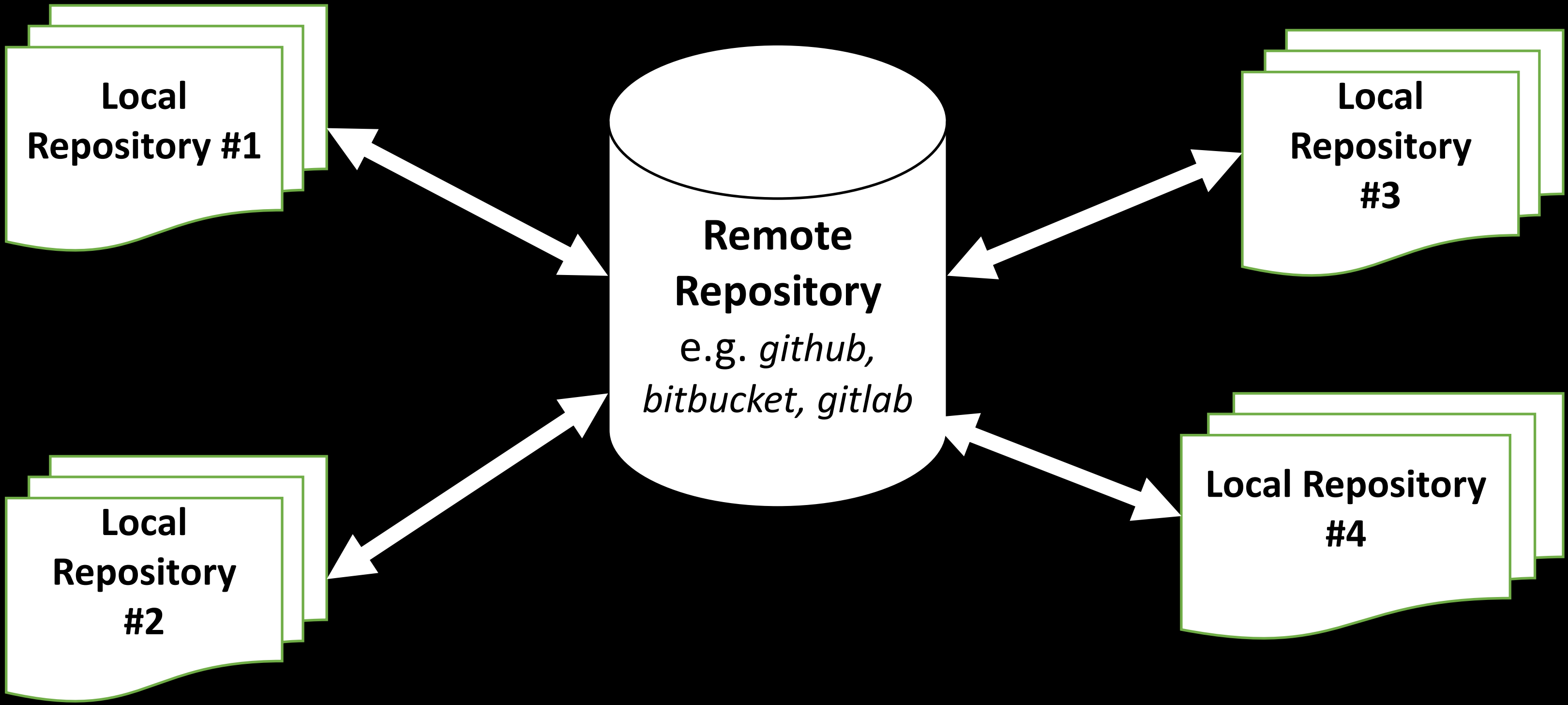
```
$ mkdir c:/code
$ mkdir c:/code/flutter
$ cd c:/code/flutter
```

Create a new file

```
$ touch readme.txt
```

What is GIT ?

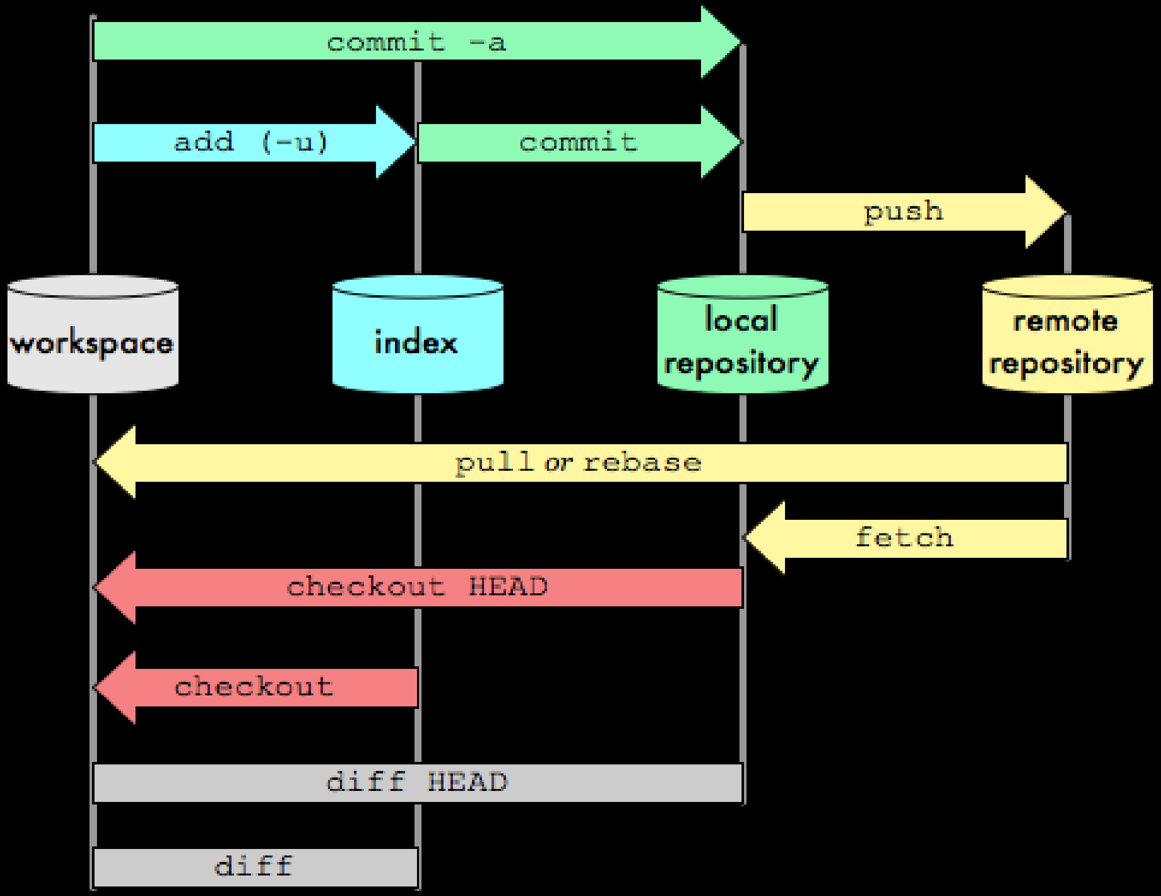
- A distributed Versioning Control System (VCS)



- Git provides a collection of tools to manage versioning of your project

What is GIT ?

- From a local repo’s perspective



Example Use Case

Clone the starter project .

```
$  git clone http://github.com/jumail-utm/flutter_logo.git   flutter_logo
```

This command will download my repo on github to your local machine

See what’s inside the repo

```
$ cd flutter_logo
$ git log
$ git log --oneline
```

git log will show a list of commits have been done on the repo

Open and run the program in vs code

```
$ code .
```

Checking out snapshot

```
$ git log --oneline
$ git checkout <commit_hash>
$ git checkout master
```

Modifying for your own work in a new branch

```
$ git checkout <commit_hash> -b my_branch
```

Git by examples»

Create a readme.txt file.

```
$ touch readme.txt
$ code readme.txt
```

Check your repo’s status

```
$ git status
```

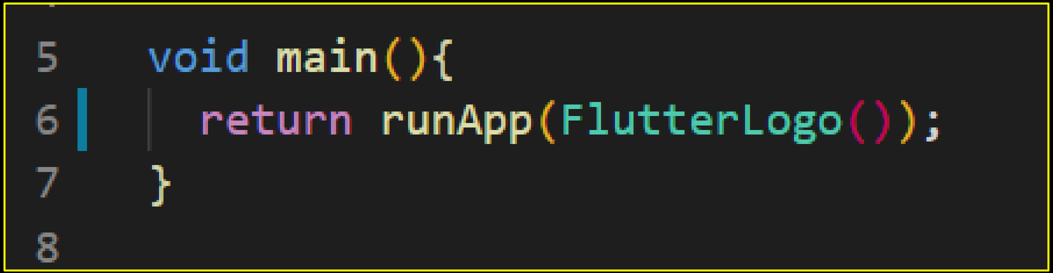
Set the newly created readme.txt file so that it is traceable

```
$ git add readme.txt
```

Commit your update to your local repo

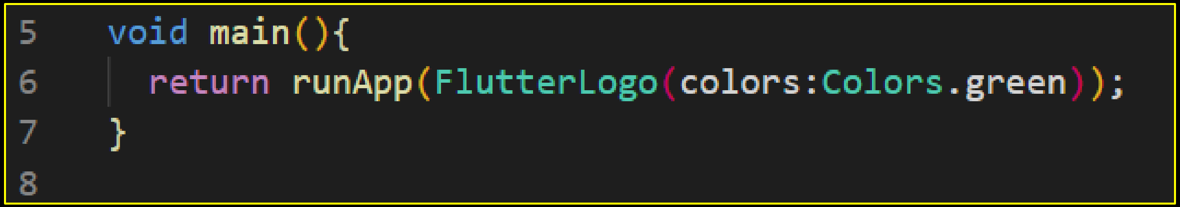
```
$ git commit -a -m “Update 1: Add a readme.txt”
```

Continue other tasks and perform a commit for each task, e.g.



```
void main(){
  return runApp(FlutterLogo());
}
```

```
$ git commit -a -m “Task 1: Add a flutter logo”
```



```
void main(){
  return runApp(FlutterLogo(colors:Colors.green));
}
```

```
$ git commit -a -m “Task 2: Change color”
```

List all branches

```
$ git branch -a
```

Go back to master branch

```
$ git checkout master
```

Creating Your own Repository

Create a starter project using the flutter template project

```
$ cd c:/code/flutter
$ flutter create flutter_counter
$ cd flutter_counter
```

Create a repo for this project

```
$ git init
$ git status
$ git add .
$ git commit –a -m "My first commit"
```

To make changes to the last commit

```
$ git add .
$ git commit --amend
```

Pushing to Remote Repo

- Login to github.com with your own account
- Create a new public repository on github.com, named flutter_counter
- Back to your git bash (command line)

```
$ git remote add origin https://github/your-username/flutter_counter.git

$ git push -u origin master
```

Sharing Offline

To share a git repo without going through a remote repo, use git bundle.

To create a bundle (e.g., in user1's PC):

```
user1$ git bundle build/flutter_counter.git HEAD master
```

Then share the bundle file by any mean, e.g. copying it to a pen-drive. The file should be inside the build directory

In another pc (e.g. user2's) copy the the bundle file and create a clone from it

```
user2$ git clone flutter_counter.git  flutter_counter
```

Git Resources

https://app.pluralsight.com/course-player?clipId=139ae6dd-af56-45a5-aa4f-9924129ef340

https://www.tutorialspoint.com/git