Hybrid Mobile Programming

Introduction and Installation

WEEK 01

Informatics Engineering

Universitas Surabaya

Lecturers

KP A → Fikri Baharuddin, S.Kom., M.Kom.
Google Space : https://chat.google.com/room/AAAAzEwp27M?cls=4

Lesson Plan

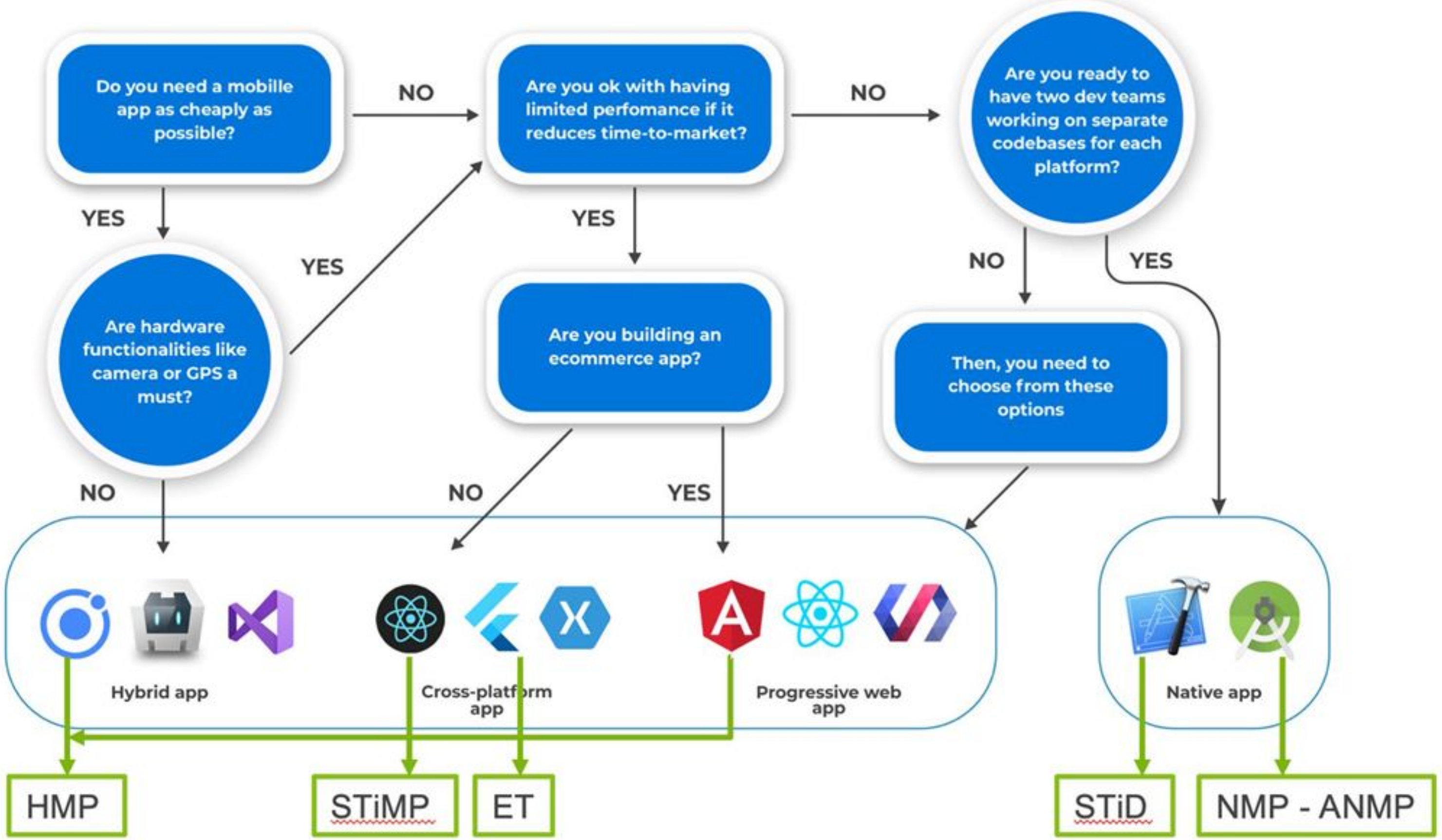
W	NTS	W	NAS
1	Introduction and Installation	8	Database – Part 1
2	Ionic Basic, Routing and Navigation	9	Database – Part 2
3	Angular Binding and Directive – Part 1	10	Local Storage
4	Angular Binding and Directive – Part 2	11	Camera Handling
5	Ionic UI – Layouting	12	Indexed DB
6	Ionic UI – Components	13	Quiz
7	Ionic Theme, Style and Animation	14	MAP

Grade (NTS and NAS)

1. Exercise
2. Quiz
3. Project

Introduction

CHOOSE A DEV APPROACH FOR YOUR MOBILE APP



Do you need a mobille app as cheaply as possible?
NO
Are you ok with having limited perfomance if it reduces time-to-market?
NO
Are you ready to have two dev teams working on separate codebases for each platform?
YES
YES
Are hardware functionalities like camera or GPS a must?
YES
Are you building an ecommerce app?
NO
YES
Then, you need to choose from these options
NO
NO
YES
Hybrid app
Cross-platform app
Progressive web app
Native app
HMP
STiMP
ET
STiD
NMP - ANMP

A Single codebase across various platforms

Ionic was built on Angular framework and Apache Cordova, as well as using HTML 5, CSS, and JavaScript as core technologies for app development..

Anyone acquainted with web technologies and Angular can use it, leveraging web skills to create fully functional applications. Forming just one codebase for all of your platforms, guarantees:

- Reduced costs on development, hiring native devs, codebase maintenance
- Faster time-to-market on both platforms
- Ease of maintenance via built-in browser instruments and debugging tools
- Availability of transforming your Ionic application into a desktop app or PWA

The economic purpose of utilizing Ionic is clear if you focus on the quick launch of the app in both application stores. Developing across Android, iOS, and maybe even Windows devices is a lot cheaper within a single codebase, compared to native development.

Popular Tech. and ease of learning

- It’s no secret that web technologies are the most widely spread, with JavaScript being the most popular programming language. According to the Stack Overflow survey 2019, frontend developers are the third largest group of all developer types. Having Ionic as your mobile application development tool will ensure that you will have no problem hiring developers for your project.

- Of course, having expertise in native development would only be a plus, as Ionic doesn’t compile the whole app into a native language. Instead, it compiles UI elements, using Cordova or Capacitor (a native-bridge platform for Ionic) plugins for the rest of the functionality. It’s easy to build and maintain an application with just the web technology stack.

Wide range of integration capabilities and plugins

If you feel that you are not gaining enough of your Ionic application, you can always integrate it with numerous tools. The official list of technologies to integrate with can be found on Ionic's website, providing easy access to analytical instruments, payment systems, security, and testing tools.

It also contains a number of plugins that help integrate with a device's hardware. But keep in mind that some plugins are available as a part of the Enterprise version of Ionic, which requires payments to use Premier plugins and tools.

For more plugins, you can also check the Cordova plugins list, which can be sorted by the platform availability. Or you can also use Capacitor plugins, downloading them from npm. A full procedure of using Capacitor plugins is described in the guide.

An extensive choice of UI elements and quick prototyping

Older versions of Ionic have already proven to be efficient in mimicking the look and feel of native applications thanks to its UI components library. These components can be used as ready-made elements to construct your graphic user interface (GUI) or utilize those elements for customizations. Paired with web components, Ionic is capable of speeding up the process of developing UI logic and retaining native look without additional costs.

Using ready-made UI elements helps to create prototypes of your future applications in a comparably short-time period. For that purpose, you are free to use a prototyping tool called Ionic Creator. It's maintained by the Ionic team, offering a drag & drop interface to construct interactive prototypes, but it can't be used for constructing the whole app

Testing Convenience

As long as Ionic apps work only via a webview, the device’s browser can be used for testing the app. It is much more convenient because you don’t even have to use a testing device to ensure that everything runs smoothly. The same concept is applicable to a variety of mobile devices in the modern world.

Browsers offer built-in testing and debugging tools that make the whole testing process convenient. To test Angular components used in older versions, Angular CLI can be used, while Ionic CLI is suitable for web components testing. So, a testing device or emulator might be needed to test some native functionality only.

Concise Docs & Strong Community

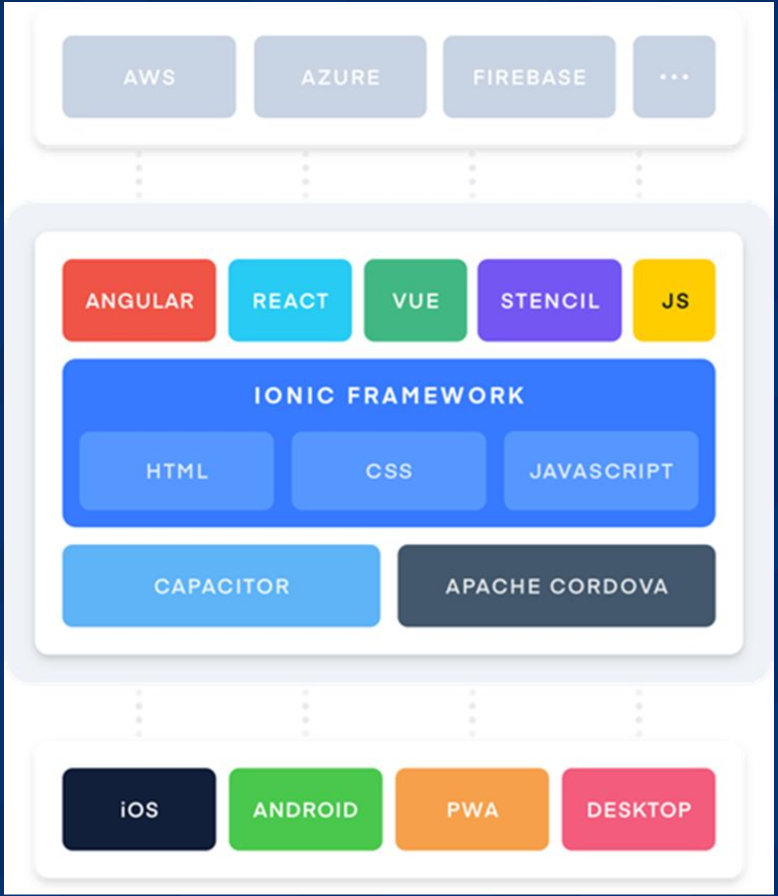
In the case of Ionic, everything is grouped on their website. The documentation can only be described as exhaustive, covering each and every useful topic on what the components of Ionic are, how to use them, and how they interrelate. In the documentation, you also can find guides for various tasks on installing, configuring, launching, and fine-tuning various instruments used with Ionic

As long as creators of Ionic take care of accessibility of their tool for the users, the community will be growing. With more than 5 million developers and constant activity on the forum, you will be able to find the answer to any question, if it wasn't covered in the documentation.

External Source

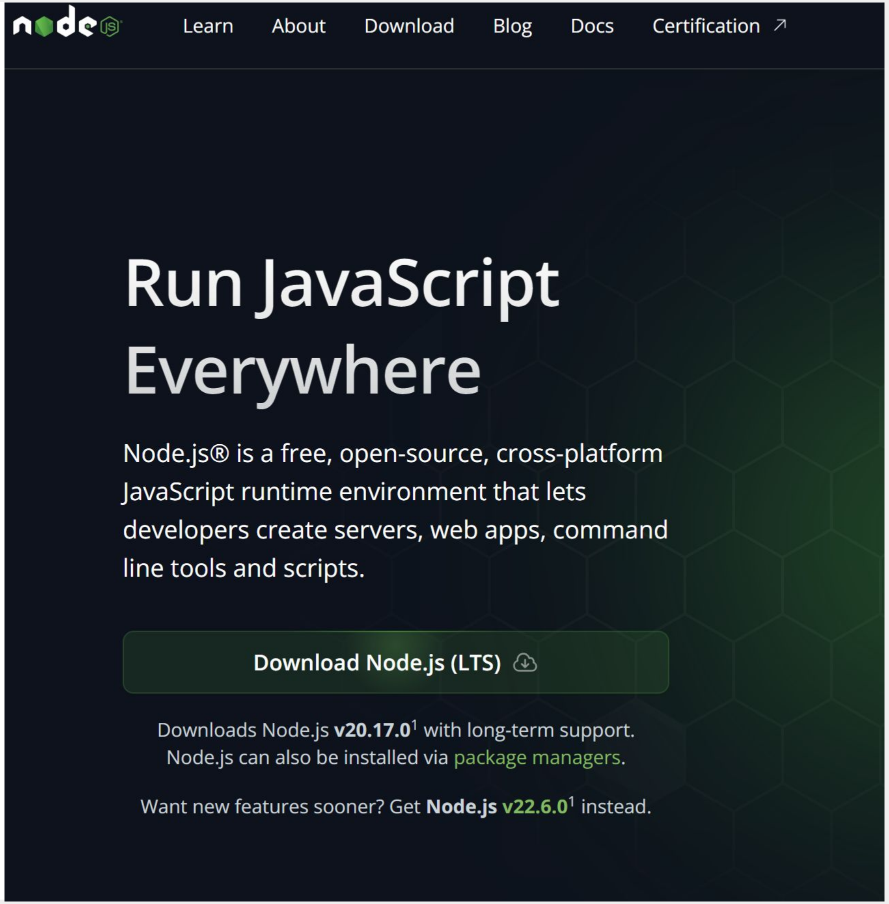
More about ionic can be found at:

www.ionicframework.com



Installation

Download :
https://nodejs.org/en/



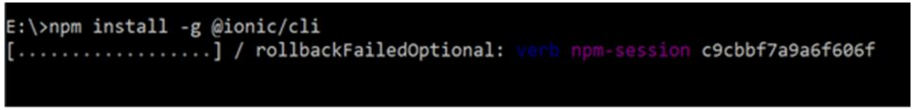
Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

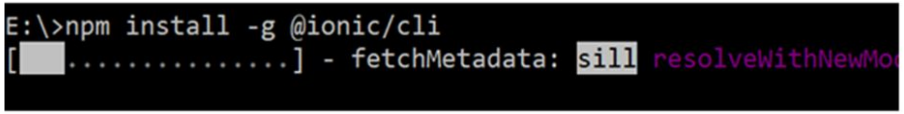
Downloads Node.js **v20.17.0**[1] with long-term support.
Node.js can also be installed via package managers.

Want new features sooner? Get **Node.js** **v22.6.0**[1] instead.

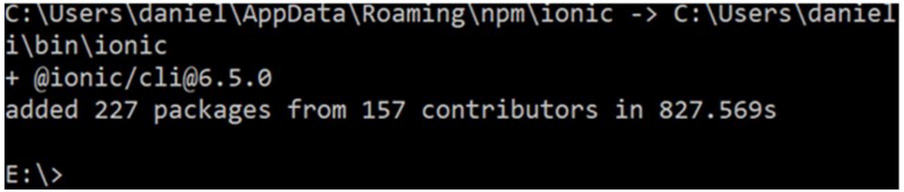
Ionic Installation



```
E:\>npm install -g @ionic/cli
[.................] / rollbackFailedOptional: verb npm-session c9cbbf7a9a6f606f
```



```
E:\>npm install -g @ionic/cli
[  ...............] - fetchMetadata: sill resolveWithNewMo
```



```
C:\Users\daniel\AppData\Roaming\npm\ionic -> C:\Users\daniel
i\bin\ionic
+ @ionic/cli@6.5.0
added 227 packages from 157 contributors in 827.569s

E:\>
```

Syntax Command :

npm install -g @ionic/cli

In HMP class, we will use **Ionic** as environment and UI engine. using **Angular** as framework**,** and new scripting language : **TypeScript.**

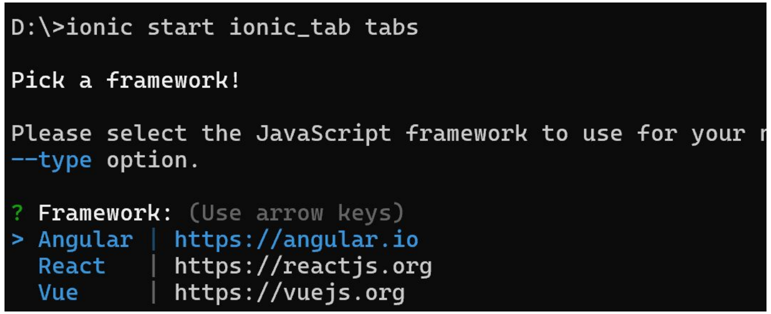
*You have to install NodeJS first, if npm command is not recognized.

Ionic Installation

Type :

1. ionic start ionic_tab tabs
2. Choose Angular
3. Choose NgModule
4. Choose *No* (Create free acc)

- After finish the installation, we will create a new project
- Ionic provide many template starter . Some are free and others are commercial. https://market.ionicframework.com/starters/
- Above starter template can be use when we create the new project. For the first time, we will use the "Tabs" template in order to see one of above different templates.

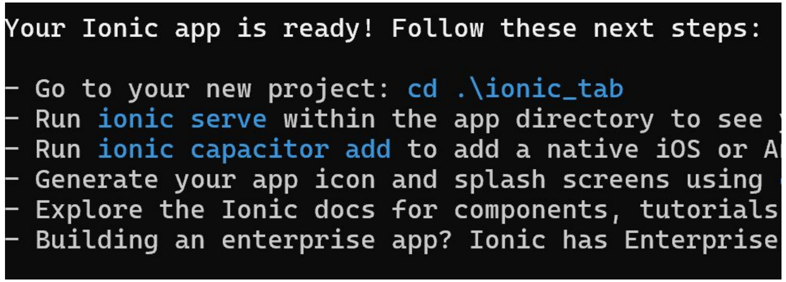


```
D:\>ionic start ionic_tab tabs

Pick a framework!

Please select the JavaScript framework to use for your r
--type option.

? Framework: (Use arrow keys)
> Angular | https://angular.io
  React   | https://reactjs.org
  Vue     | https://vuejs.org
```



```
Your Ionic app is ready! Follow these next steps:

- Go to your new project: cd .\ionic_tab
- Run ionic serve within the app directory to see y
- Run ionic capacitor add to add a native iOS or An
- Generate your app icon and splash screens using c
- Explore the Ionic docs for components, tutorials
- Building an enterprise app? Ionic has Enterprise
```

However, in following weeks we will use blank template for activities

If there is some error..and it raises the same error when you repeat the process, you may need to execute this command first: **npm cache clear –force**

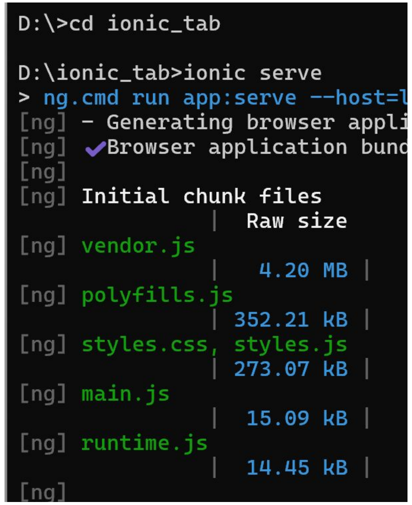
Ionic Serve

After finish creating project, you will have new folder with the same name with project name. go to that folder with cd command. and do serve command.

You can browse with address localhost:8100

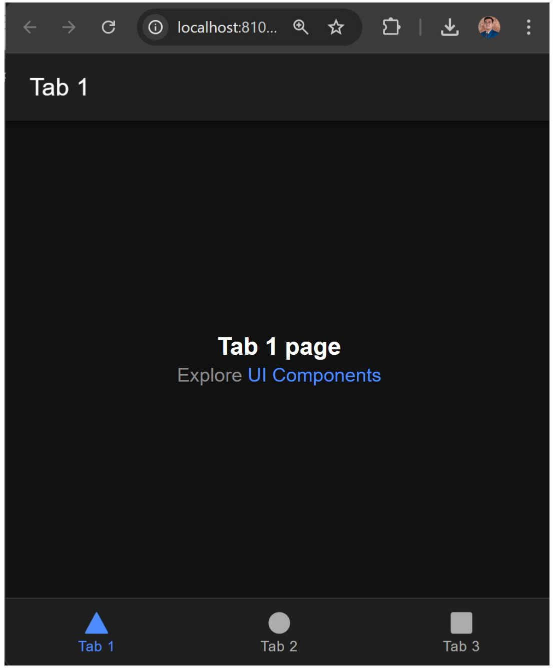
Type :

1. cd ionic_tab
2. ionic serve



```
D:\>cd ionic_tab

D:\ionic_tab>ionic serve
> ng.cmd run app:serve --host=l
[ng] - Generating browser appli
[ng] ✔Browser application bund
[ng]
[ng] Initial chunk files
                    | Raw size
[ng] vendor.js
                    |   4.20 MB |
[ng] polyfills.js
                    | 352.21 kB |
[ng] styles.css, styles.js
                    | 273.07 kB |
[ng] main.js
                    |  15.09 kB |
[ng] runtime.js
                    |  14.45 kB |
[ng]
```



capacitor

- Capacitor is a cross-platform app runtime that makes it easy to build web apps that run natively on iOS, Android, Electron, *and* the web.

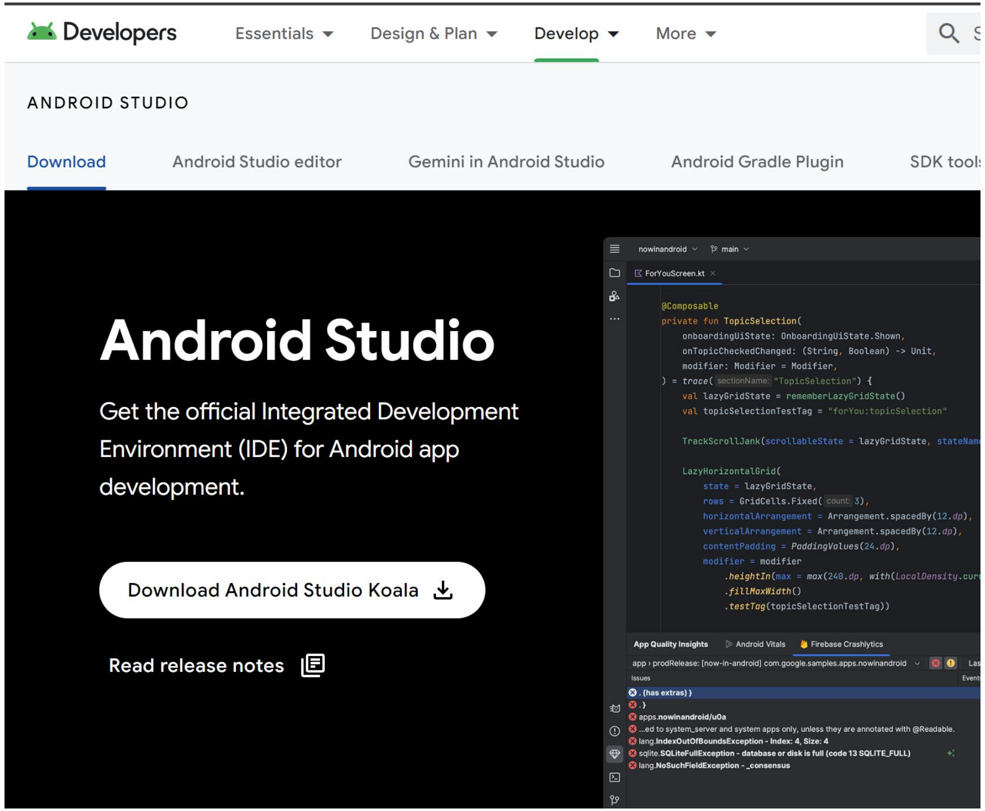
- Capacitor provides a consistent, web-focused set of APIs that enable an app to stay as close to web-standards as possible, while accessing rich native device features on platforms that support them. Adding native functionality is easy with a simple Plugin API for Swift on iOS, Java on Android, and JavaScript for the web.

capacitor

You have to install android studio first before using capacitor.

Download :

https://developer.android.com/studio



Android Studio

Get the official Integrated Development Environment (IDE) for Android app development.

```
@Composable
private fun TopicSelection(
    onboardingUiState: OnboardingUiState.Shown,
    onTopicCheckedChanged: (String, Boolean) -> Unit,
    modifier: Modifier = Modifier,
) = trace( sectionName: "TopicSelection") {
    val lazyGridState = rememberLazyGridState()
    val topicSelectionTestTag = "forYou:topicSelection"

    TrackScrollJank(scrollableState = lazyGridState, stateNam

    LazyHorizontalGrid(
        state = lazyGridState,
        rows = GridCells.Fixed( count: 3),
        horizontalArrangement = Arrangement.spacedBy(12.dp),
        verticalArrangement = Arrangement.spacedBy(12.dp),
        contentPadding = PaddingValues(24.dp),
        modifier = modifier
            .heightIn(max = max(240.dp, with(LocalDensity.cur
            .fillMaxWidth()
            .testTag(topicSelectionTestTag))
```

```
app › prodRelease: [now-in-android] com.google.samples.apps.nowinandroid
. (has extras) }
.}
apps.nowinandroid/u0a
...ed to system_server and system apps only, unless they are annotated with @Readable.
lang.IndexOutOfBoundsException - Index: 4, Size: 4
sqlite.SQLiteFullException - database or disk is full (code 13 SQLITE_FULL)
lang.NoSuchFieldException - _consensus
```

capacitor

Next commands to run is **ionic build** for compiling the ionic project, then we have to install android platform by using **npm install @capacitor/android**. Following with **npx cap add android** to add android native platform.

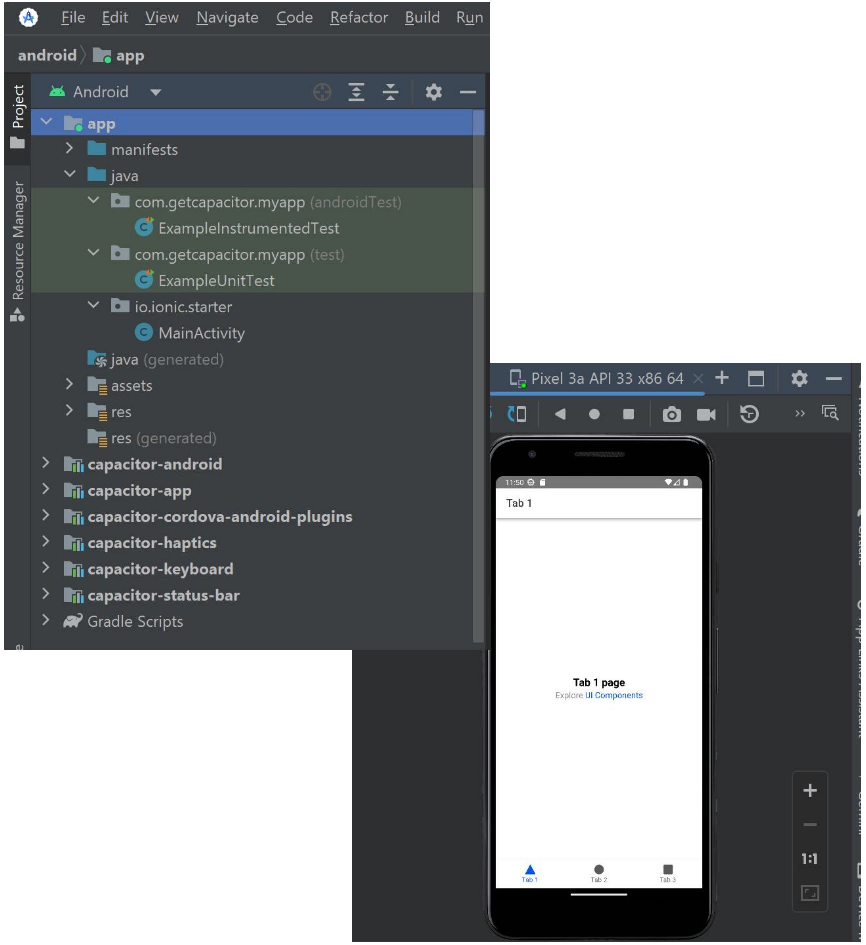
At last use **npx cap open android** to open our project in android studio.

1. ionic build
2. npm install @capacitor/android
3. npx cap add android
4. npx cap open android

capacitor

After our project opened in android studio, we can run it in emulator or in device that connect without computer (remember your Native Mobile Programming class).

Also we can get our apk or aab file.(aab file is used for delivering application to google play store)

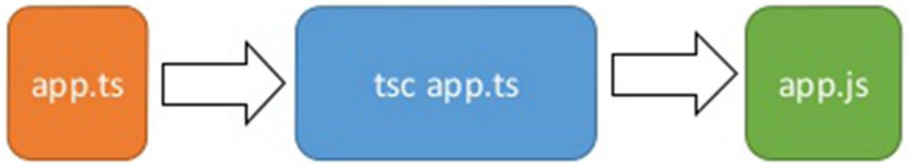


ANGULARJS by Google

- Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps.
- Angular is a development platform, built on TypeScript. As a platform, Angular includes:
  - A component-based framework for building scalable web applications
  - A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more
  - A suite of developer tools to help you develop, build, test, and update your code
- Maintained by Google
- It adopt MVW (Model - View - Whatever)

- TLDR: because Angular JS use typescript
- It is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for the development of large applications and transcompiles to JavaScript.
- Typescript supports OOP, strongly typed, and compiled language.
- Extensions of Typescript is .ts and already supported by all browser

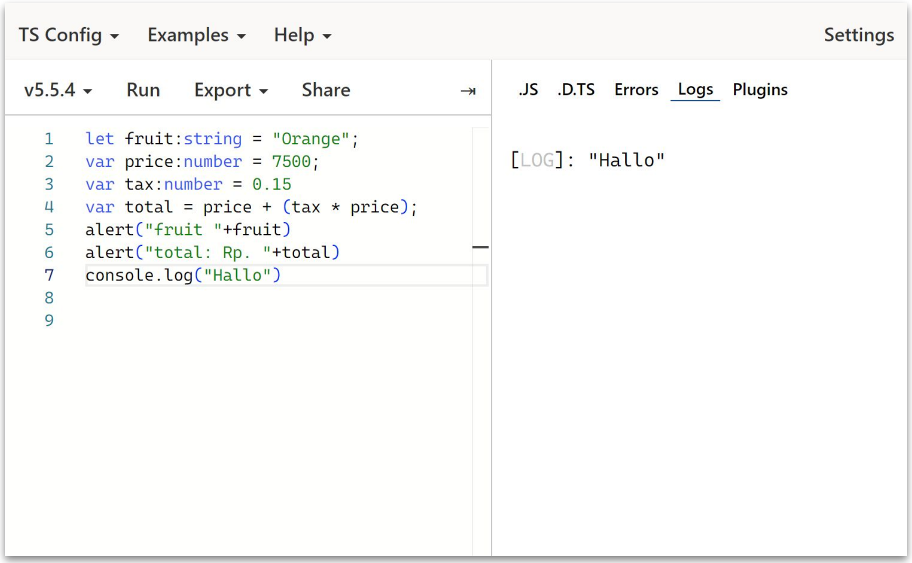
- Typescript is a compiler language. It mean Typescript have to compile by a compiler first. TS can not run directly.
- One of typescript compiler is nodejs server.
- File .ts will be compiled as JavaScript (.js) file. Command to compile is **tsc.**



Learn TypeScript

TypeScript Playground

- The purpose of this chapter is to make familiar with typescript
- The simple way to try typescript is by using online tool: http://www.typescriptlang.org/play/
- You can also use nodejs and VisualStudio Code to code and compile Typescript. You can follow this tutorial: https://code.visualstudio.com/docs/languages/typescrip



```
let fruit:string = "Orange";
var price:number = 7500;
var tax:number = 0.15
var total = price + (tax * price);
alert("fruit "+fruit)
alert("total: Rp. "+total)
console.log("Hallo")
```

```
[LOG]: "Hallo"
```

Variable

Typescript support data type: number, string, boolean, void, undefined, null

Example declaration : (please try to put and run it in playground) :

```
let fruit:string = "Orange";
var price:number = 7500;
var tax:number = 0.15
var total = price + (tax * price);
alert("fruit "+fruit)
alert("total: Rp. "+total)
```

Variable Scope

```
var global_num = 12                 //global variable
class Numbers {
   num_val = 13;                    //class variable
   static sval = 10;                //static field

   storeNum():void {
         var local_num = 14;        //local variable
   }
}
```

Operator

Typescript support several operator:

- Arithmetic: +, -, *, /, ++, --, %
- Relational: <, >, <=, >=, ==, !=
- Logical: &&, ||, !
- Bitwise: &, |, <<, >>, <<<, ~
- Assignment: +=, -=, *=, /=, =
- typeof: operator that return the type of variable
- instanceof: operator to check whether an object is created from certain class

Example

```
let nama = 1235
var price:number = 7500
var qty:number = 10
var member: boolean = true
if (typeof (nama) != 'string') {
        alert("Invalid name")
} else if(!member) {
        alert("Price: Rp. " + (price*qty))
} else if(price*qty > 70000) {
        alert("Discounted Price: Rp. " + (price*qty) * 0.9)
}
```

Loops

- Typescript support 3 type of loop: for, while, and do while
- Typescript also support break and continue statement
- The Syntax is similar with another programming language

Example :

```
var budget: number = 100000
var cart: number = 0
while (cart < budget) {
    var price = Math.random() * 10000
    cart += price
    var body = document.body.innerHTML
    body += "Rp." + price.toFixed(0) + "<br/>"
    document.body.innerHTML = body
}
```

Defining & Calling a Function

To define function:

```
function  function_name() {
  // function body
}
```

To call function:

```
function_name()
```

Function with Return Value

To define function with return value :

```
function greet():string {
   return "Hello World"
}
var msg = greet()
alert(msg)
```

Parameterized Function

```
function test_param(n1:number,s1:string ="oke" ) {
   alert(n1)
   alert(s1)
}

test_param(123,"this is a string")
```

Array

Example of array declaration and initiation :

```
var vocals:string[];
vocals = ["a","i","u","e","o"]
alert(vocals[0]);
var consonants = ["b","c","d","f","g"]
```

Some array method can be used are join(), concat(), pop(), push(), toString(), sort(), etc

Example Loop in Array (for..in)

```
var j;
var nums:number[] = [1001,1002,1003,1004]
for(j in nums) {
   alert(nums[j])
}
```

Class Definition

```
class Car {
   //field
   engine:string;

   //constructor
   constructor(engine:string) {
    this.engine = engine
   }
   //function
   disp():void {
    alert("Engine is  :   "+this.engine)
   }
}
```

Create Instance Object

It is no different from other programming languages:

```
var obj = new Car("SUPER SPEED")
obj.disp()
```

Objects

- Object have key – value pair, and it is not an instance of a class.
- Object is independent and create to handle something that not have to much data (so we not need to make the class)

Example :

```
var person = {
  firstname:"Tom",
  lastname:"Hanks"
};
alert(person.firstname + " " + person.lastname)
```

Export

- Typescript is a modular language. It mean in some part function and variable not globally provided and we have to imported first when we need it.

- The export keyword is used in order to make the interface of the class can be imported by another class

Export used to allow the person class to be used elsewhere

- File Person.ts

```
export class Person {
  . . .
}
```

Import command specifies the class name within brackets {}

- Pada file test.ts

```
import { Person } from "./Person"
var saya = new Person()
```

Decorator

- Decorators provide a very simple syntax for calling higher-order functions
- Decorators are a design pattern that is used to separate modification or decoration of a class without modifying the original source code.
- In AngularJS, decorators are functions that allow a service, directive or filter to be modified prior to its usage

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
```

@Component is an example of a decorator. It is initially imported at the top, then this component class is modified by some of its properties (inside curly braces). A decorator is different from inheritance (subclassing)

TASK

Project UTS

- Make a group with maximum of three students.
- Register your group in → Link Pendaftaran Kelompok HMP Genap 24/25
- The Project Topic will be announced in week 2.

Thanks.