Summer 2, 2019 - CS 4520/CS5520 – Mobile Application Development

Pratheep Kumar Paranthaman, Ph.D.,

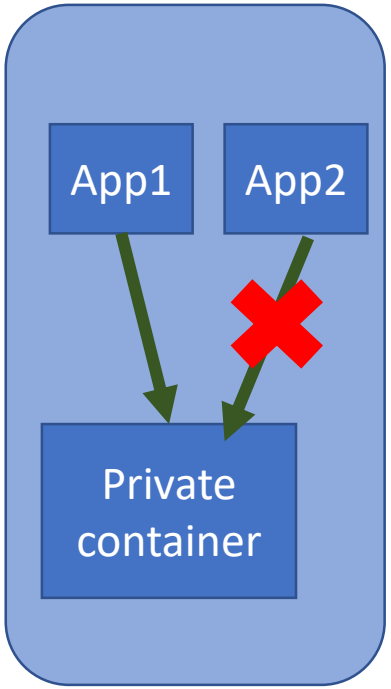
Announcements

- Checkpoint 2
- Portfolio page
- Supplementary topics

Data storage

Data Storage in Android

- **Internal Storage –** Stores in app-private file system on the device (Private)



Data Storage in Android - Internal

- Creating a File in internal Storage

```
File file = new File(context.getFilesDir(), filename);
```

- Writing to a File (**FileOutputStream**)

```
String filename = "myfile";
String fileContents = "Hello world!";
FileOutputStream outputStream

try {
    outputStream = openFileOutput(filename, Context.MODE_PRIVATE);


} catch (Exception e) {
    e.printStackTrace();
}
```

Data Storage in Android - Internal

- Creating a File in internal Storage

```
File file = new File(context.getFilesDir(), filename);
```

- Writing to a File (**FileOutputStream**)

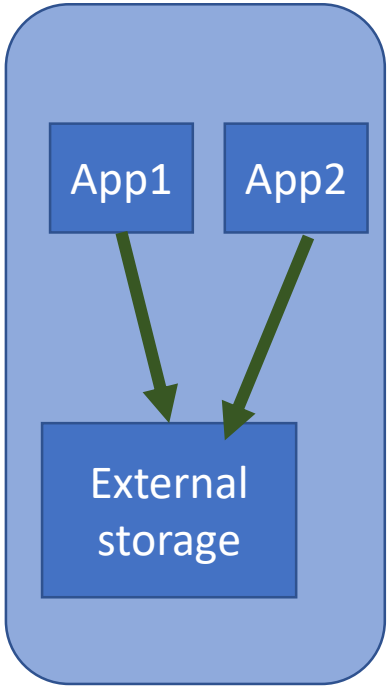
```
String filename = "myfile";
String fileContents = "Hello world!";
FileOutputStream outputStream

try {
    outputStream = openFileOutput(filename, Context.MODE_PRIVATE);
    outputStream.write(fileContents.getBytes());
    outputStream.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

Write data

Data Storage in Android - External

- **External Storage –** Stores in external location on the device
  - Readable by other apps
  - Not guaranteed to be accessible



Data Storage in Android - External

- **Save files to external storage**
  - getExternalStoragePublicDirectory()
  - Save location -> DIRECTORY_PICTURES or DIRECTORY_MUSIC

```
File file = new File(Environment.getExternalStoragePublicDirectory(
        Environment.DIRECTORY_PICTURES), albumName);
```

Manifest permissions??

App's Manifest file (AndroidManifest.xml)

- Must have for an app

- ***Describes essential details about your app:***

  - Package Name (used by Build tools for building your app)
  - Application components
    - Activity (data about each activity)
    - Services
    - Broadcast receivers
    - Content Providers
  - Device Configurations and Compatibility
  - **Permissions**

App's Manifest file (AndroidManifest.xml)

Package Name

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.myapp"
   android:versionCode="1"
   android:versionName="1.0" >
   ...
</manifest>
```

App's Manifest file (AndroidManifest.xml)

- **App Components - Activity**

```
<manifest package="com.example.myapp" ... >
   <application ... >
      <activity android:name=".MainActivity" ... >
         ...
      </activity>
   </application>
</manifest>
```

Activity definition

App’s Manifest file (AndroidManifest.xml)

Intent filters - define the behaviour of an activity

```
<activity android:name="MainActivity">



</activity>
```

App's Manifest file (AndroidManifest.xml)

Intent filters - defines the behaviour of an activity

```
<activity android:name="MainActivity">
  <!-- This activity is the main entry, should appear in app launcher -->
  <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Main Entry point

App's Manifest file (AndroidManifest.xml)

- **Device Configurations and Compatibility (Need specification)**
  - Declare the hardware /software features that your app would require.
  - Google PlayStore doesn't allow your app to be installed on devices that lack the hardware /software specifications

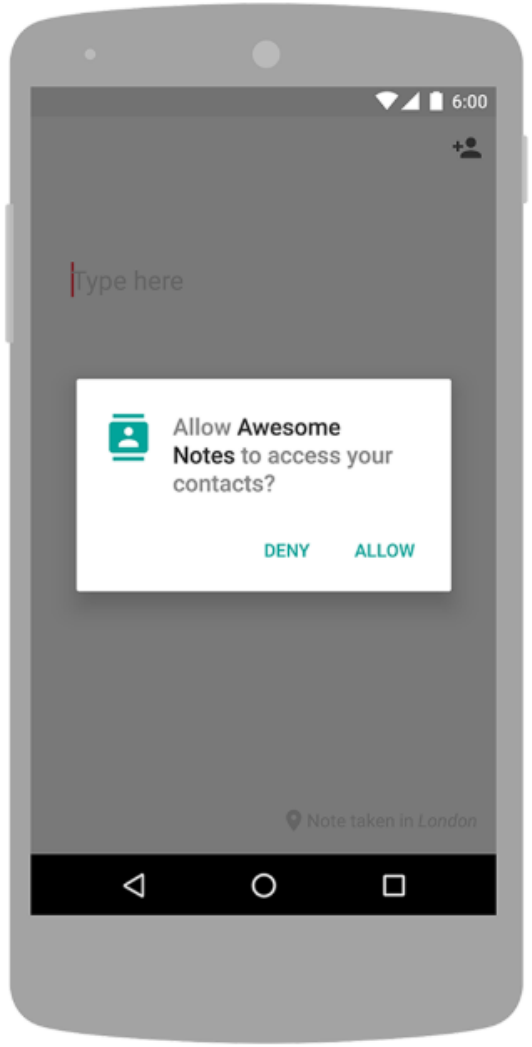
<uses-feature> -> specify the needs of your app

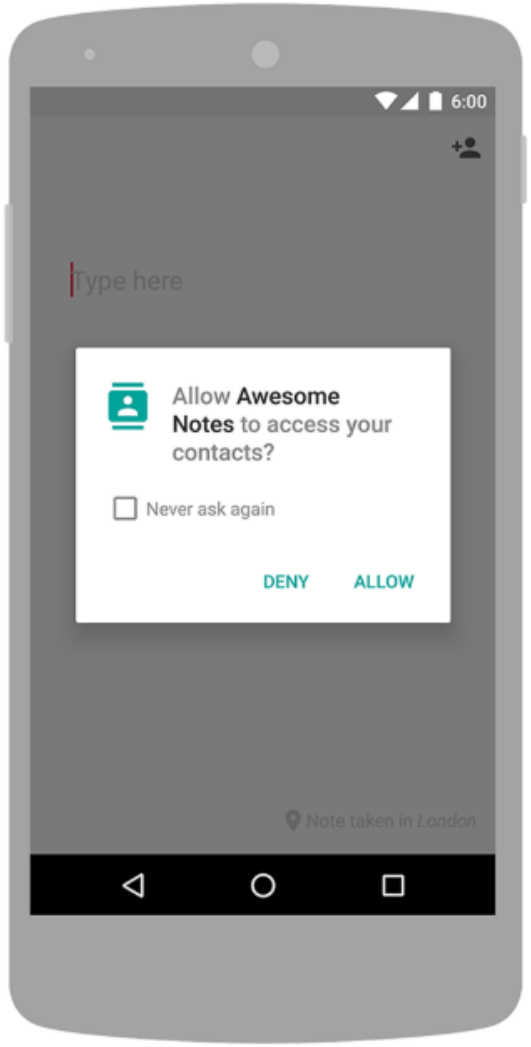
```
<manifest ... >
   <uses-feature android:name="android.hardware.sensor.compass"
                 android:required="true" />
   ...
</manifest>
```

App's Manifest file (AndroidManifest.xml)

- Permissions – Apps must request permission for accessing sensitive components of the device.
  - Example components: Contacts, camera, sensors, etc.
  - ***Tag(<uses-permission>)*** and ***Value(WRITE_EXTERNAL_STORAGE)*** to specify the permission

```
<manifest ...>
   <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
   <uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.SEND_SMS" />
</manifest>
```





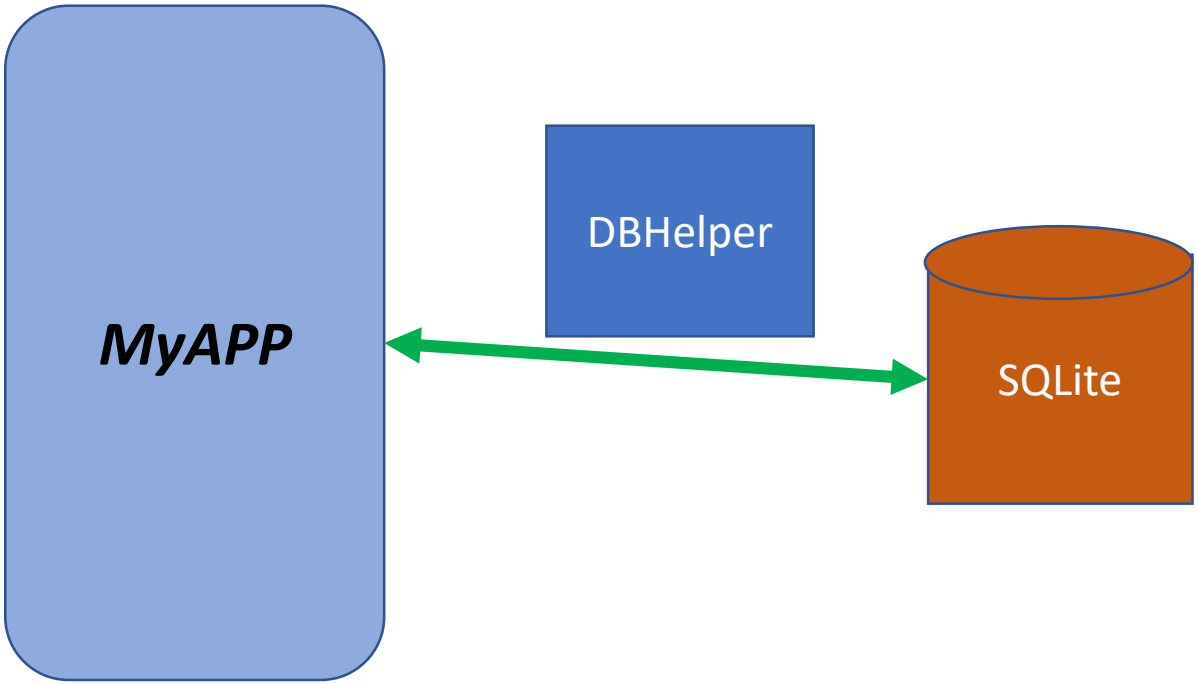
[developer.android.com/guide/topics/permissions/overview]

Permission Group	Permissions
CALENDAR	• READ_CALENDAR • WRITE_CALENDAR
CALL_LOG	• READ_CALL_LOG • WRITE_CALL_LOG • PROCESS_OUTGOING_CALLS
CAMERA	• CAMERA
CONTACTS	• READ_CONTACTS • WRITE_CONTACTS • GET_ACCOUNTS
LOCATION	• ACCESS_FINE_LOCATION • ACCESS_COARSE_LOCATION
MICROPHONE	• RECORD_AUDIO
PHONE	• READ_PHONE_STATE • READ_PHONE_NUMBERS • CALL_PHONE • ANSWER_PHONE_CALLS • ADD_VOICEMAIL • USE_SIP
SENSORS	• BODY_SENSORS
SMS	• SEND_SMS • RECEIVE_SMS • READ_SMS • RECEIVE_WAP_PUSH • RECEIVE_MMS
STORAGE	• READ_EXTERNAL_STORAGE • WRITE_EXTERNAL_STORAGE

Shared Preferences

Database

- Database(SQLite) – stores structured data
  - Private to your app



Database types - SQL

- SQL – RDBMS(Relational database management system)
  - Table based structure – the data will be arranged in tables with links to each other.
  - Pre-defined schema for the structure
  - Examples:
    - MySQL
    - PostgreSQL
    - Microsoft SQL Server
    - Oracle
    - And the list continues….

SQL

Employee table

Empno (PK)	Ename	Job	Deptno (FK)
101	A	Salesman	10
102	B	Manager	10
103	c	Manager	20

Department table

Deptno (PK)	dname	loc
10	Sales	Chicago
20	Sales	Chicago
30	Finance	New York

Primary Key

Foreign Key

Primary Key

SQL - Advantages and Disadvantages

- **Well-documented schemas**
- **Strict to ensure integrity**

- **Flexibility issues - Won't be able to adapt for the changes in data structure**
- **Quantity of data**
  - **Ill suited for large analytics and IoT based apps**

NoSQL

- **Non-relational databases**
  - Ideal for semi-structured and unstructured
  - Key-value pairs -> most commonly used
  - Applications: Games, IoT, rapid analytics

  - Examples:
    - Amazon DynamoDB
    - Mongo DB
    - CouchDB
    - Hbase
    - Redis
    - And again the list continues..

```
{
  "results": [
    {
      "gender": "male",
      "name": {
        "title": "mr",
        "first": "rolf",
        "last": "hegdal"
      },
      "location": {
        "street": "ljan terrasse 346",
        "city": "vear",
        "state": "rogaland",
        "postcode": "3095",
        "coordinates": {
          "latitude": "54.8646",
          "longitude": "-97.3136"
        },
        "timezone": {
          "offset": "-10:00",
          "description": "Hawaii"
        }
      },
      "email": "rolf.hegdal@example.com",
      "login": {
        "uuid": "c4168eac-84b8-46ea-b735-c9da9bfb97fd",
        "username": "bluefrog786",
        "password": "ingrid",
        "salt": "GtRFz4NE",
        "md5": "5c581c5748fc8c35bd7f16eac9efbb55",
        "sha1": "c3feb8887abed9ec1561b9aa2c9f58de21d1d3d9",
        "sha256": "684c478a98b43f1ef1703b35b8bbf61b27dbc93d52acd515e141e97
```


[includehelp.com/code-snippets/json-parse-and-stringify.aspx]

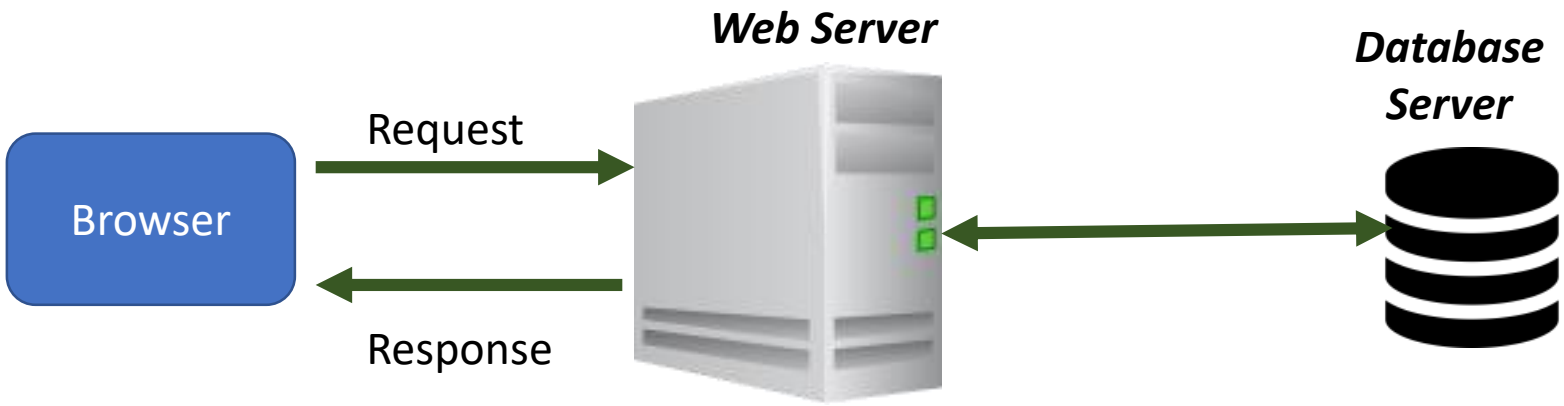
NoSQL - Advantages and Disadvantages

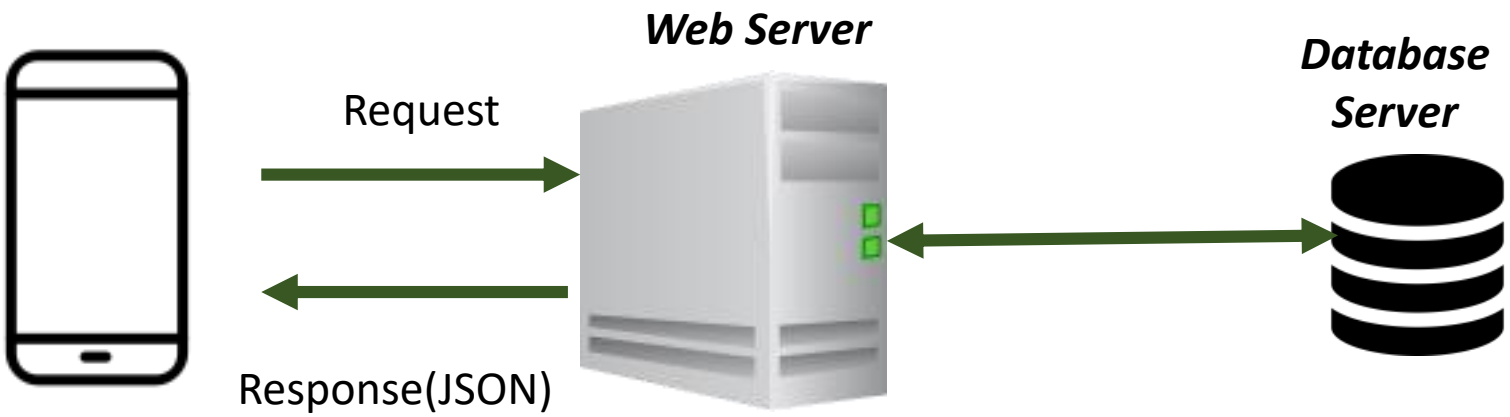
- **Schema-free**
- **Highly flexible to changes in your data**

- **Integrity issues**
  - **Can be handled by setting certain security parameters**

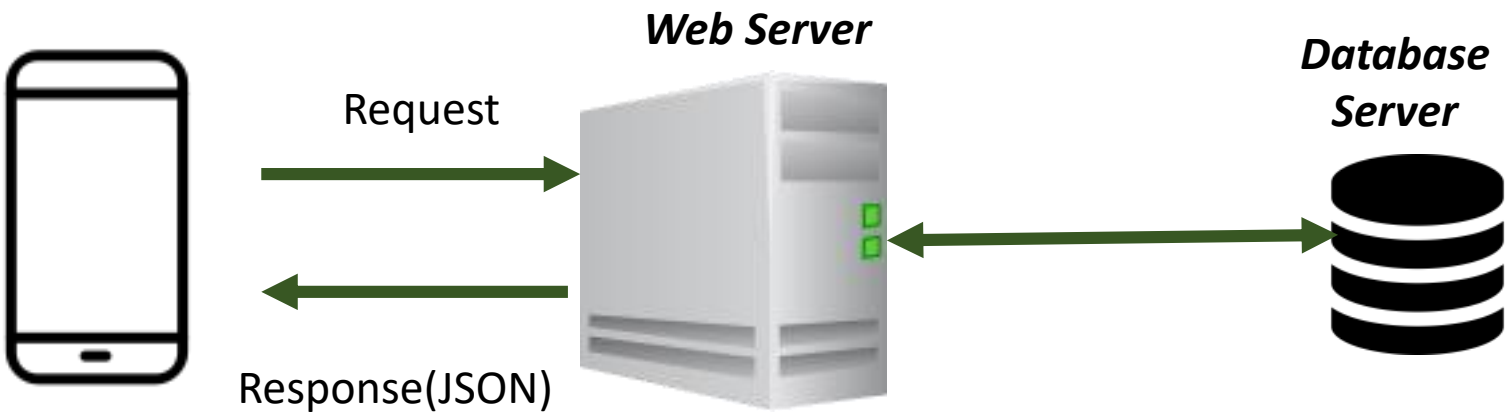
Remote Database

Client – Server(Server based)

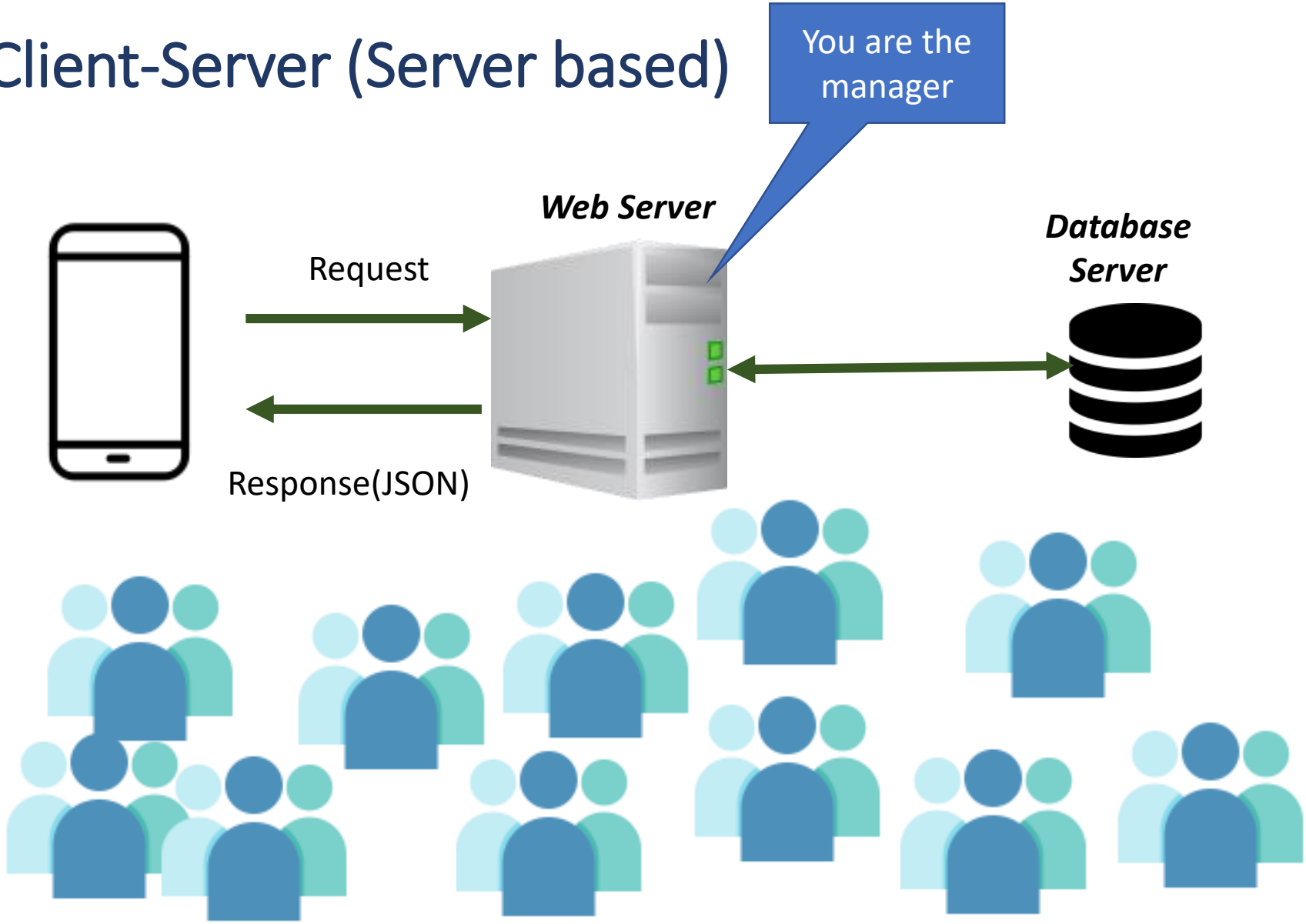




Client-Server (Server based)



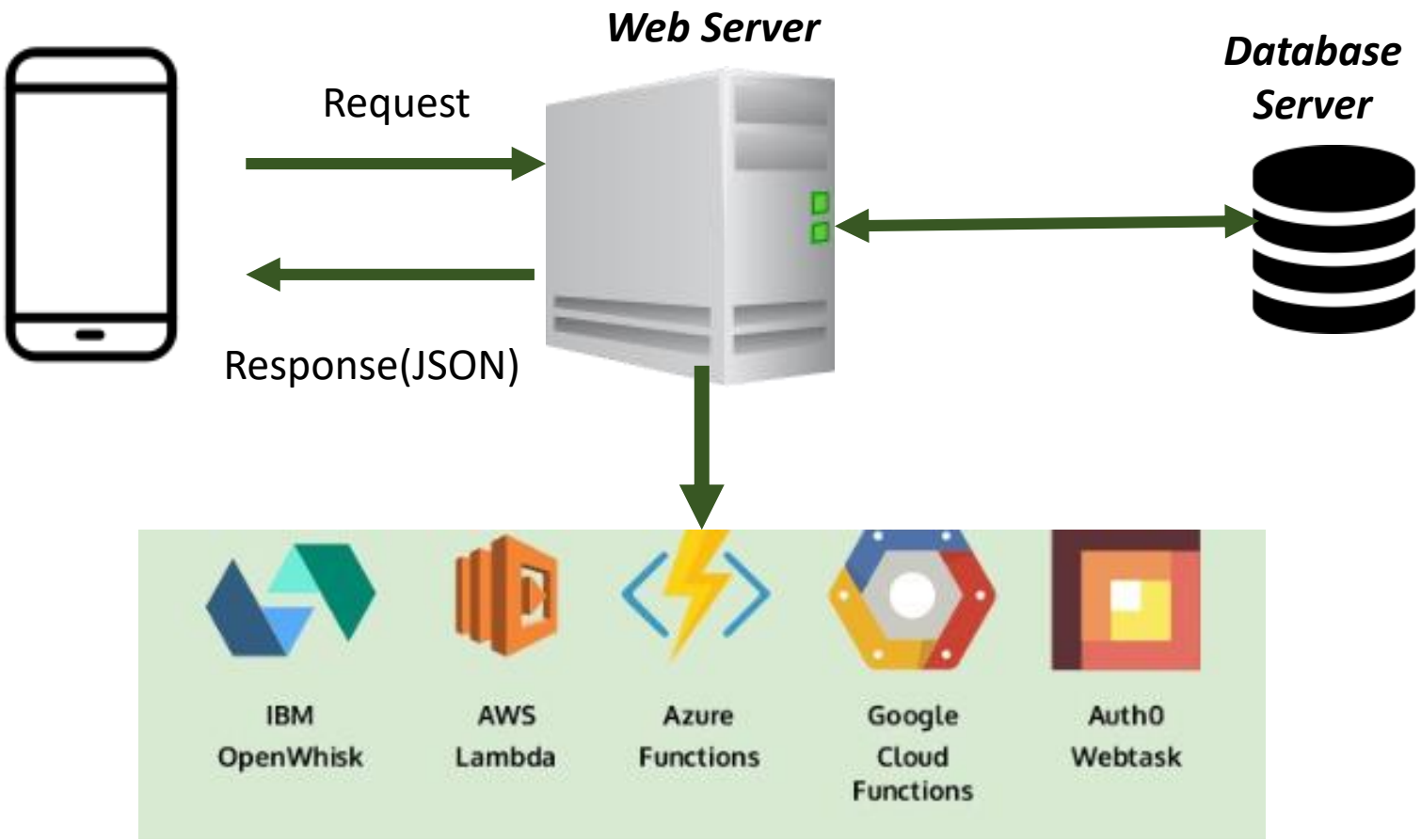
Client-Server (Server based)



Major Responsibilities and management issues

- Authentication
- Database management
- Storage
- Security
- Cost
- Scaling

ServerLess



[hackernoon.com/what-is-serverless-architecture-what-are-its-pros-and-cons-cc4b804022e9]

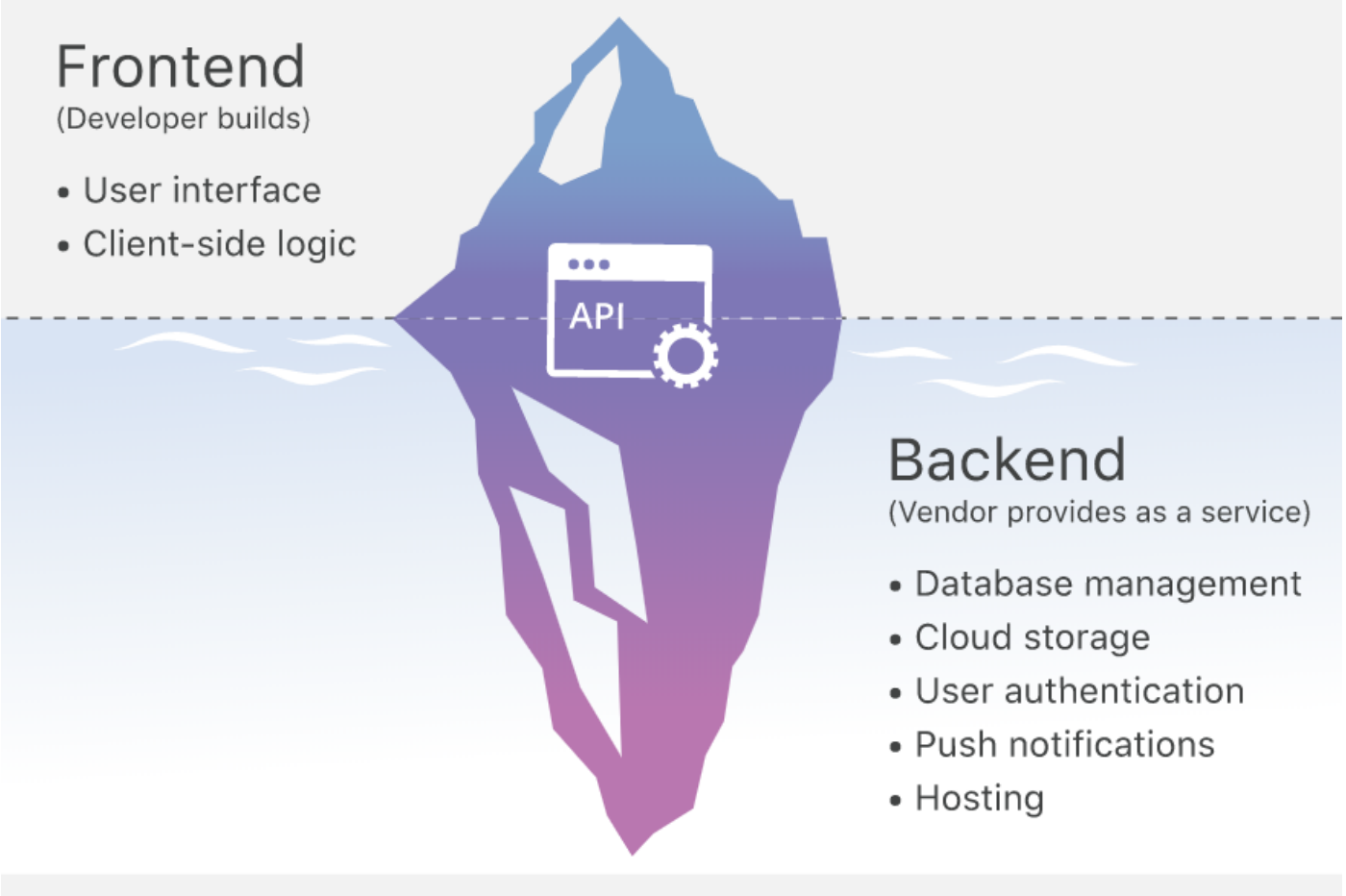
Major Responsibilities and management issues

- Authentication
- Database management
- Storage
- Security
- Cost
- Scaling

Cloud Vendors are responsible

BaaS – Backend as a service

Backend as a Service(BaaS)



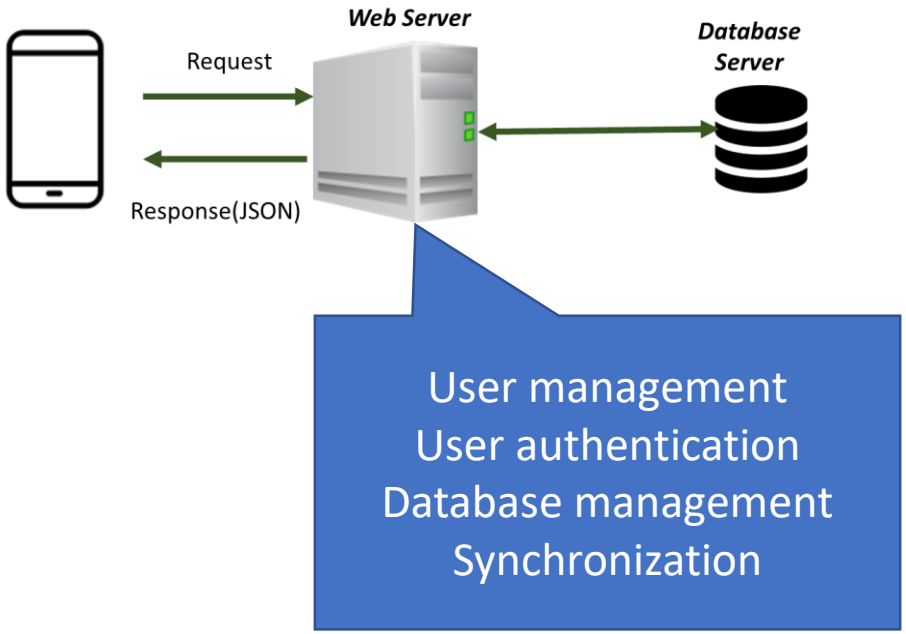
[cloudflare.com/learning/serverless/glossary/backend-as-a-service-baas/]

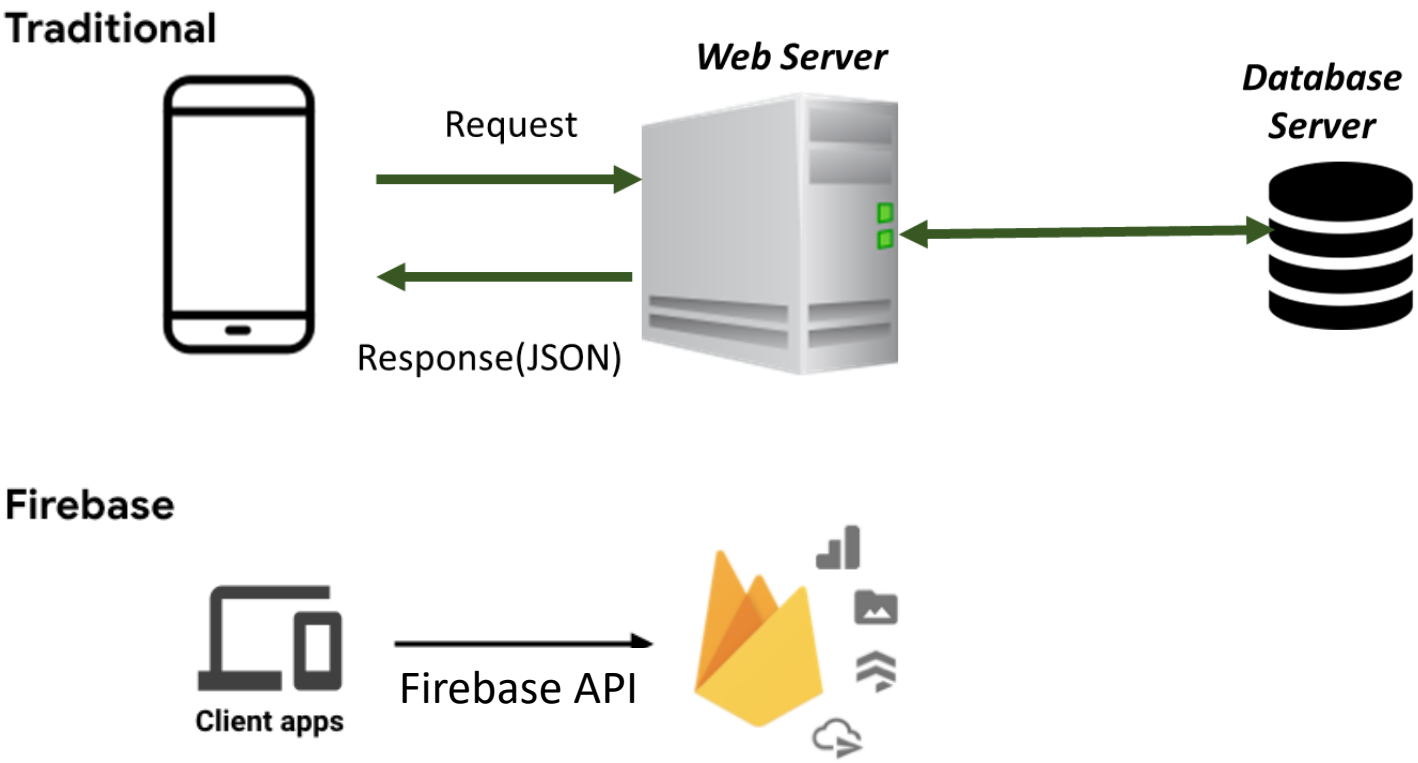
Firebase

Firebase

- BaaS – provides API for all Backend functionalities.
  - Realtime database with cloud solutions

- Authentication
- Realtime database
- Storage
- Hosting
- Cloud Messaging
- MLKit





[medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0]

Firebase – Realtime database

- **Realtime database** –
  - NoSQL DB
  - stores data to cloud and notifies all users in **milliseconds**

- **Realtime data synchronization**
  - Allow users to extract data from any device

- Database security rules to manage user access.

- No server maintenance/operations

[medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0]

Root

Child

```
joule-test-3d32f
  task1
    anything: "name the vars weirdly plz
    manual_description: "you do the stuff with some stuff , maybe add a
    manual_task: "Stuff with Stift
    timeframe: "4-6 years
    user: "Hannan Rhodes
  task2
    anything: "dont make him em
    manual_description: "Raise my kid 4 me plz, thnx i cant because that
    manual_task: "Raize my kidz
    timeframe: "18-37 years
    user: "John Snow
```

[stackoverflow.com/questions/43573762/printing-all-the-data-from-a-firebase-db-in-react]

Firebase

- ***Authentication – manages user registration and authentication***
- ***Realtime database – Cloud hosted NoSQL Database***
- ***Storage – storage management for various files(images, documents)***
- Hosting – Global web hosting
- MLKit – SDKs for ML tasks

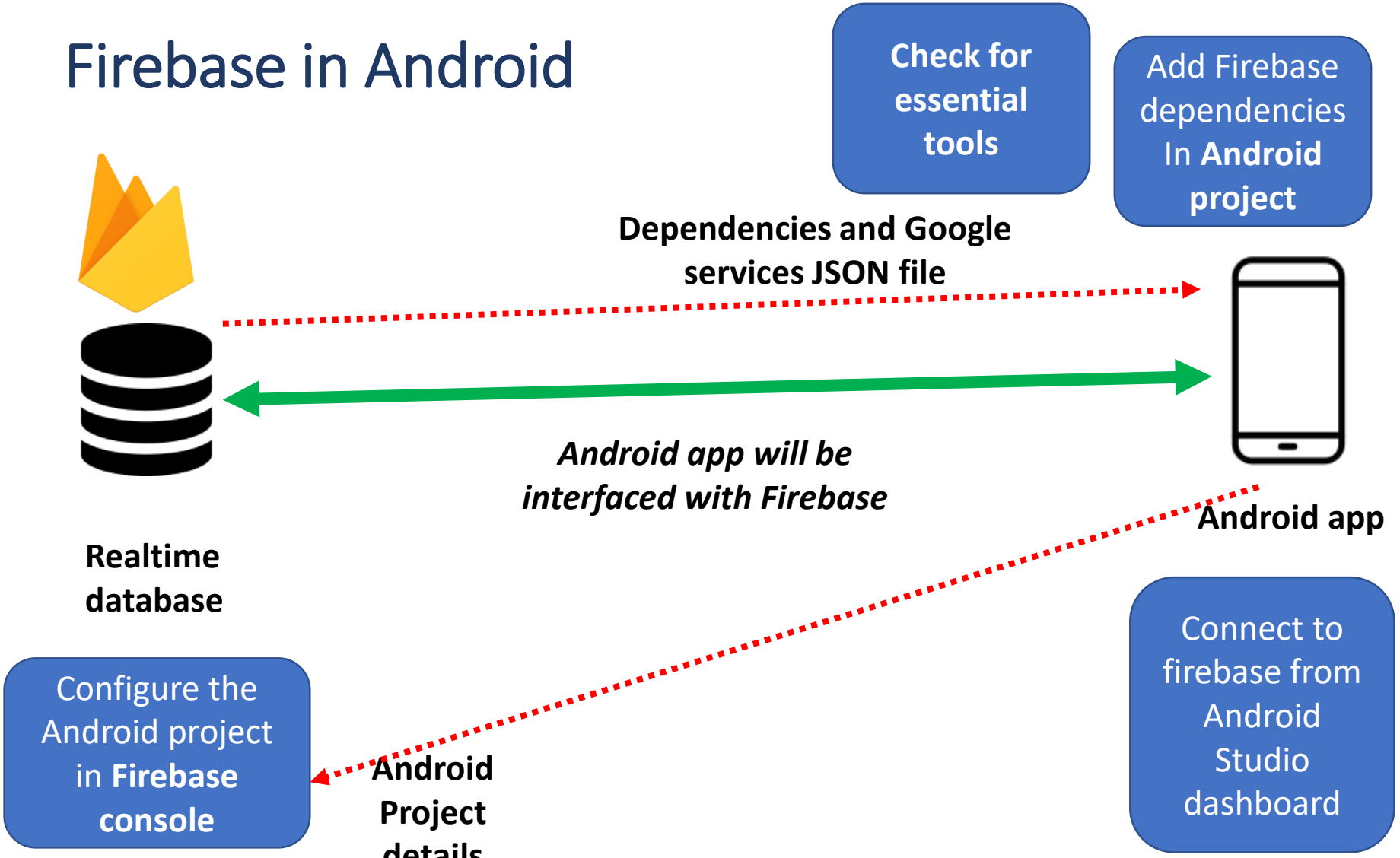
Connecting Firebase to Android App?

Connect Firebase with Android project

- **Part 1 – Firebase console**
  - Create project for Android platform
  - Include the Android PackageName and SHA1
  - Grab the Google PlayServices JSON file
  - Copy the required ***Dependencies***

- **Part 2 – Android Studio Project**
  - Enable AndroidX artefacts while creating the project
  - Integrating firebase API - Add the required dependencies for firebase in Gradle file
  - Open firebase console in Android(Tools -> Firebase)
    - Realtime database -> enable the first two steps

Firebase in Android



Firebase in Android

- **Configuration**
  - **Firebase console**
  - **Android Studio**

- **Data transactions**
  - **Read/write operation from your android app**

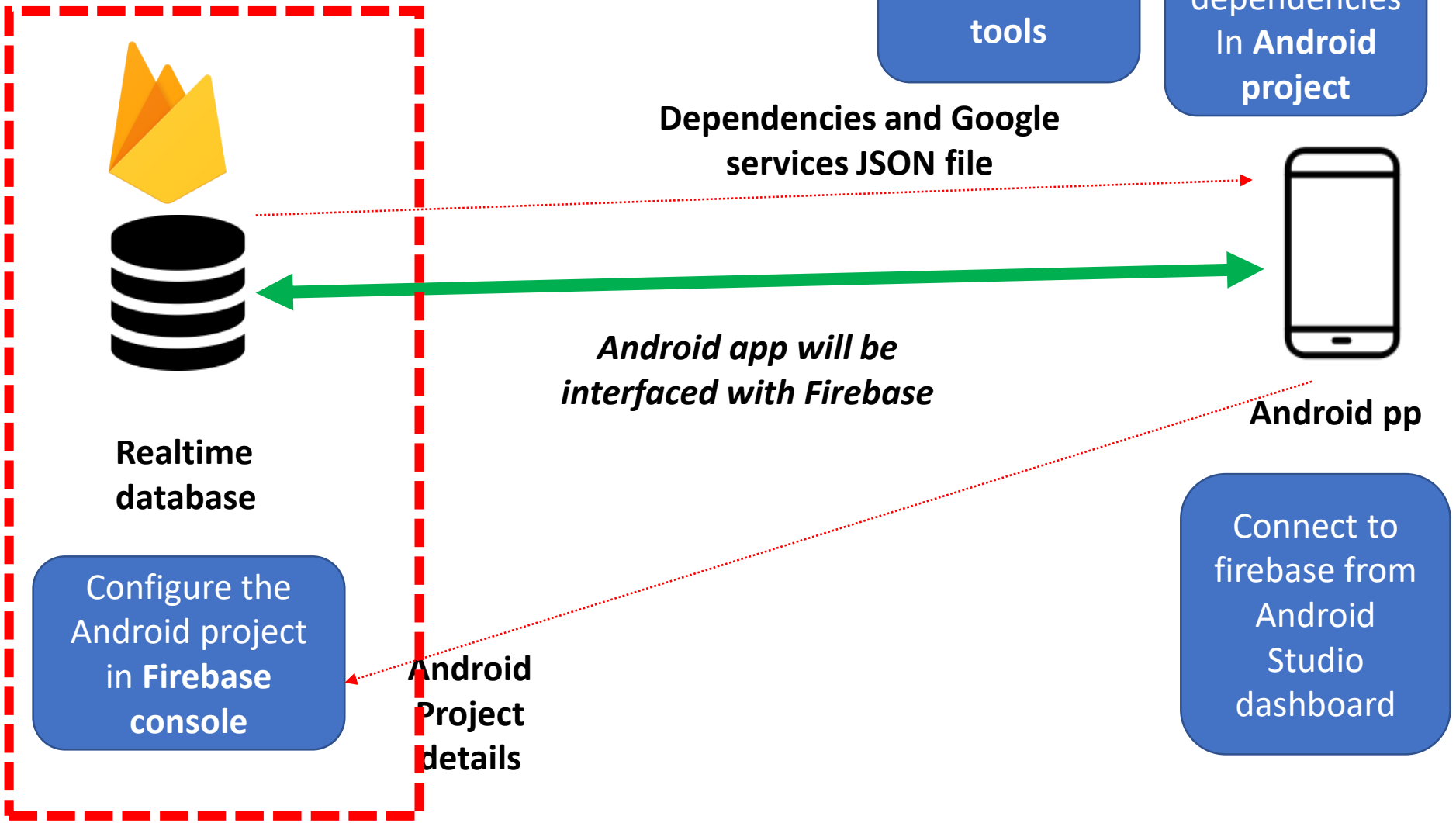
Let’s connect the Firebase with our App!

Required Tools

- GooglePlay Services 9.0 or Later.

- Android SDK tools:
  - Google PlayServices 30.0 or Later
  - Google Repository 26.0 or Later

Firebase in Android



Connect Firebase with Android project

- **Part 1 – Firebase console**
  - Create project for Android platform
  - Include the Android PackageName and SHA1
  - Grab the Google PlayServices JSON file
  - Copy the required ***Dependencies***

- **Part 2 – Android Studio Project**
  - Enable AndroidX artefacts while creating the project
  - Integrating firebase API - Add the required dependencies for firebase in Gradle file
  - Open firebase console in Android(Tools -> Firebase)
    - Realtime database -> enable the first two steps

Part 1 - Firebase console

Part 1 – Firebase Console

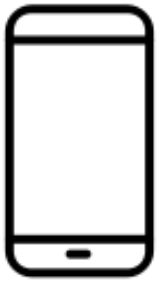
- **Part 1 – Firebase console**
  - Create project for Android platform
  - Include the Android PackageName and SHA1
  - Grab the Google PlayServices JSON file
  - Copy the required ***Dependencies***

Firebase in Android

Check for essential tools

Add Firebase dependencies In **Android project**

Dependencies and Google services JSON file



Android app will be interfaced with Firebase

Android pp

Realtime database

Configure the Android project in **Firebase console**

Android Project details

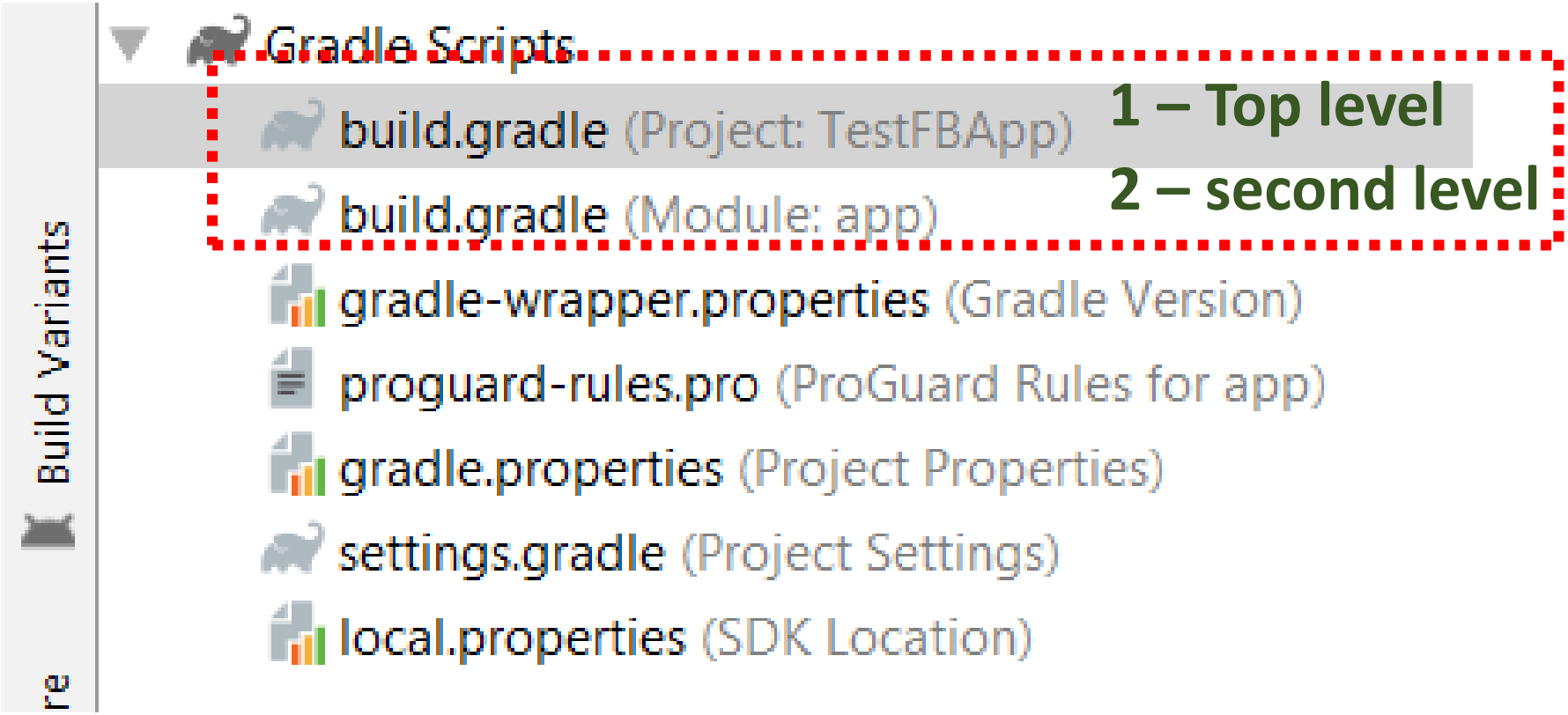
Connect to firebase from Android Studio dashboard

Part 2 – Android Studio

Adding dependencies

- You need to add the Firebase dependencies to the **gradle files**
  - These dependencies will allow you to access the firebase functionalities

Adding dependencies



Adding dependencies

Make sure to Sync your Gradle files

- **build.gradle -> 1. Top level**
  - Add the below line to the dependencies block

```
classpath 'com.google.gms:google-services:4.2.0'
```

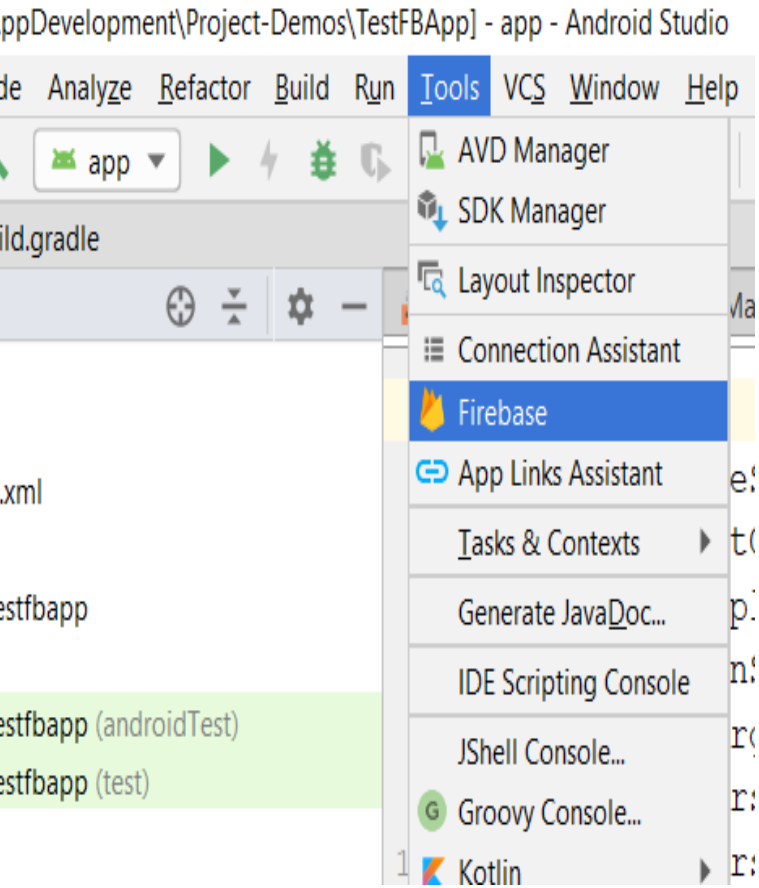
- **build.gradle -> 2 . Second level**
  - **Add the below lines to dependencies block**

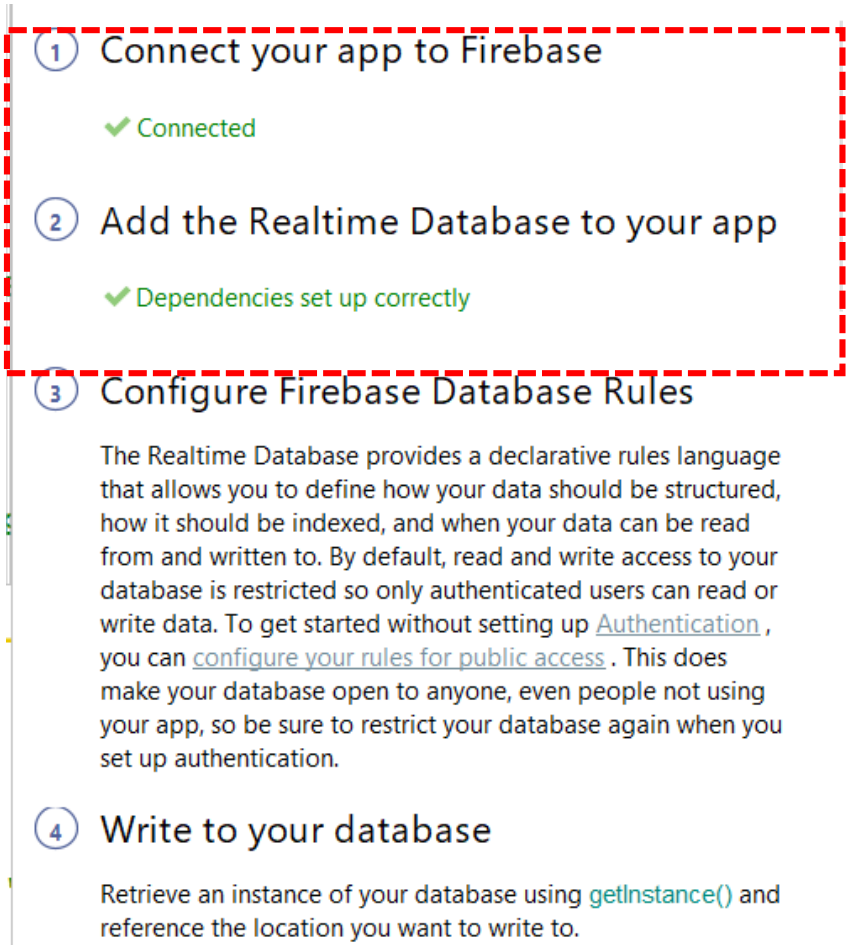
```
implementation 'com.google.firebase:firebase-core:17.0.0'
implementation 'com.google.firebase:firebase-database:18.0.0'
```

Add the below line to the end of the file

```
apply plugin: 'com.google.gms.google-services'
```

Check the Firebase status on Android Studio





Launch the App and check the Logcat

- At this point, there shouldn't be any errors on your Logcat. If there are any issues, then the configuration of Firebase has some problems.

- Check for the following
  - Necessary SDK tools
  - Proper versions of Firebase dependencies
    - Check this link - https://firebase.google.com/docs/android/setup

Firebase documentation

- https://firebase.google.com/docs/android/setup#available_libraries