

Introduction Firebase Database

Mohammed Ouda El-krnshawy

Feb 2017

Contents

1	Introduction	1
1.1	Into To Book	1
1.2	Firebase Definition	1
2	Services	3
2.1	Analytics	4
2.1.1	Firebase Analytics	4
2.2	Develop	4
2.2.1	Firebase Cloud Messaging	4
2.2.2	Auth	4
2.2.3	Realtime database	4
2.2.4	Storage	4
2.2.5	Hosting	5
2.2.6	Test Lab for Android	5
2.2.7	Crash Reporting	5
2.3	Grow	5
2.3.1	Notifications	5
2.3.2	Remote Config	5
2.3.3	App Indexing	6
2.3.4	Dynamic Links	6
2.3.5	Invites	6
2.3.6	Google AdWords	6
3	Connect Firebase To App	7
3.1	Create Firebase Project	7
3.2	Add Firebase To App	9
3.3	Add Network Permission	11
4	Insert And Retrieve Data	13
4.1	Insert Data to Firebase Back-end	13
4.1.1	Create Class Places	13

4.1.2	Design of Layout	14
4.1.3	Code that Insert Data	15
4.2	Retrieve Data from Firebase Back-end	16
4.2.1	Design Retrieve layout	16
4.2.2	Code that retrieve data	17

Chapter 1

Introduction

1.1 Into To Book

we'll be going over Firebase which is a database backend And Will Covers Some Features Which it provide . Build And Implementation Some App in android Studio which Provides Some features such as update and retrieve data, take advantage of Firebase's offline capabilities, Realtime Sync,learn how to use Firebase's Security and Rules language to secure and add permissions to your data. And show code by photos And videos, And Add Firebase to it.

1.2 Firebase Definition

Firebase is a Mobile (ios , Android) and Web platform with tools and infrastructure That Help Developers to build high-quality apps and earn money [2], It enables you to quickly get synchronized data up and running for multi-user apps. This is important because nearly every mobile app these days requires authentication and real-time data updates[1].

Chapter 2

Services

Help Developer to built better apps and don't focus on operation . Save crucial development time and ship a high-quality, bug-free app. There are three main categories of services in Firebase, we will discuss there services in details .



Figure 2.1: Firebase Services

2.1 Analytics

2.1.1 Firebase Analytics

At the heart of Firebase is Firebase Analytics, a free App measurement solution and unlimited analytics solution. See user behavior and measure attribution from a single dashboard.

2.2 Develop

2.2.1 Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) is a cross-platform solution for messages and notifications for Android, iOS, and web applications. Deliver and receive messages reliably. Using FCM, you can notify a client app that new email or other data is available to sync. You can send notification messages to drive user and retention. A message can transfer a payload of up to 4KB to a client app.

2.2.2 Auth

This is the most important thing for most apps need to know the identity of a user. Knowing a user's identity allows an app to save user data in the cloud. It supports authentication using passwords, popular identity providers like Google, Facebook and Twitter, and more.

2.2.3 Realtime database

Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline. Data is stored as JSON and synchronized in realtime to every connected client.

2.2.4 Storage

Cloud Storage is built for app developers who need to store and serve user-generated content, such as photos or videos. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps.

2.2.5 Hosting

Firebase Hosting serve your web App,provides fast and secure static hosting for your web app.production-grade web content hosting for developers. With Hosting, you can quickly and easily deploy web apps and static content

2.2.6 Test Lab for Android

Firebase Test Lab for Android provides cloud-based infrastructure for testing Android apps. With one operation, you can testing your app across a wide variety of devices. Test results—including logs, videos, and screenshots—are made available in your project in the Firebase console. Test Lab can exercise your app automatically, looking for crashes.

2.2.7 Crash Reporting

Crash Reporting create reports of the error in Apps, Errors are grouped into issues based on having similar stack, and and notify in your project in the firebase console.

2.3 Grow

2.3.1 Notifications

Firebase Notifications is a free service that enables targeted user notifications for mobile app developers. Firebase Notifications provides an option for developers a flexible notification platform that requires minimal coding effort to get started, and a graphical console for sending messages. Using the Notifications console GUI.

2.3.2 Remote Config

Remote Config change the behavior and appearance of your app without publishing an app update.it a cloud service that lets you change the behavior and appearance of your app without requiring users to download an app update. When using Remote Config, you create in-app default values that control the behavior and appearance of your app. Then, you can later use the Firebase console to override in-app

default values for all app users. Your app controls when updates are applied, and it can frequently check for updates and apply them .

2.3.3 App Indexing

Firestore App Indexing gets your app into Google Search. If users have your app installed, they can launch your app and go directly to the content they're searching for.

2.3.4 Dynamic Links

Firestore Dynamic Links are links that work the way you want, on multiple platforms. With Dynamic Links, your users get the best available experience for the platform they open your link on. If a user opens a Dynamic Link on iOS or Android, they can be taken directly to the linked content in your native app. If a user opens the same Dynamic Link in a desktop browser, they can be taken to the equivalent content on your website.

2.3.5 Invites

Firestore Invites are an out-of-the-box solution for app sharing via email or SMS. To customize the invitation user experience, or to generate links programmatically

2.3.6 Google AdWords

Reach potential customers with online ads.

Chapter 3

Connect Firebase To App

3.1 Create Firebase Project

Open Browser and go to "https://console.firebase.google.com" to create app in firebase console to add to your app in android studio

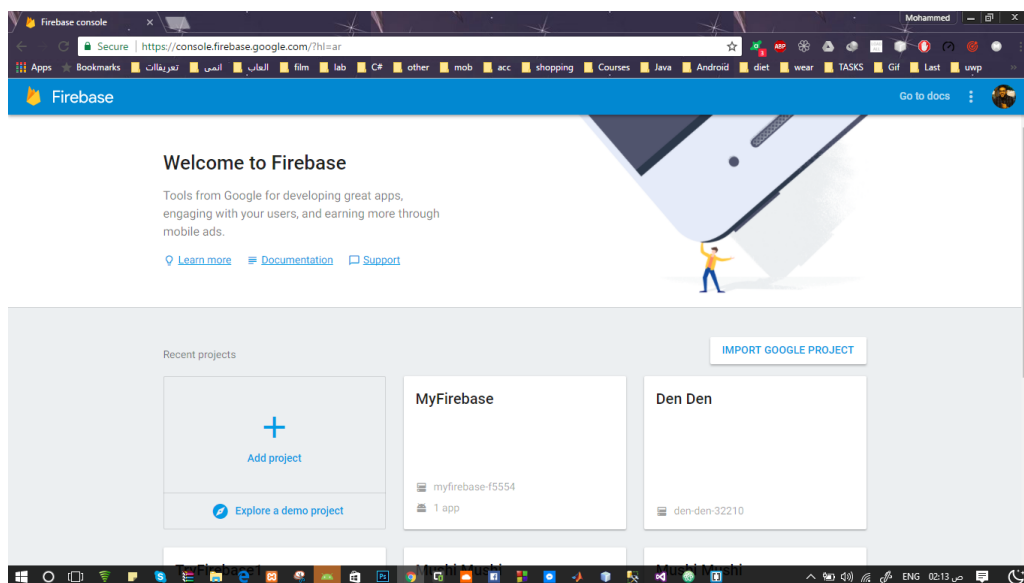


Figure 3.1: Firebase Console

click on Add Project to display this form and enter the name of your app and your Country/region

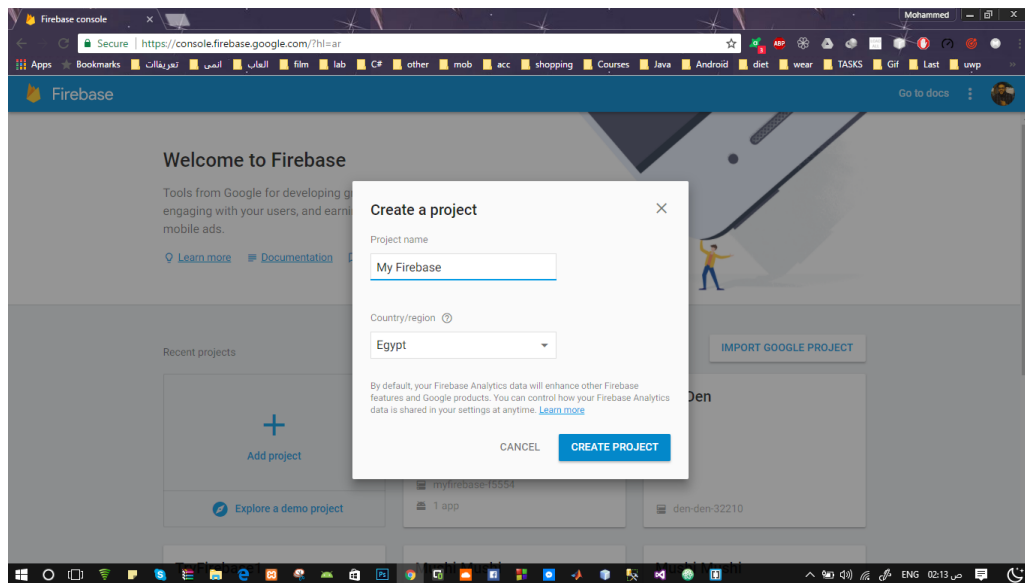


Figure 3.2: Create new project

After enter the data and click on 'Create Project'

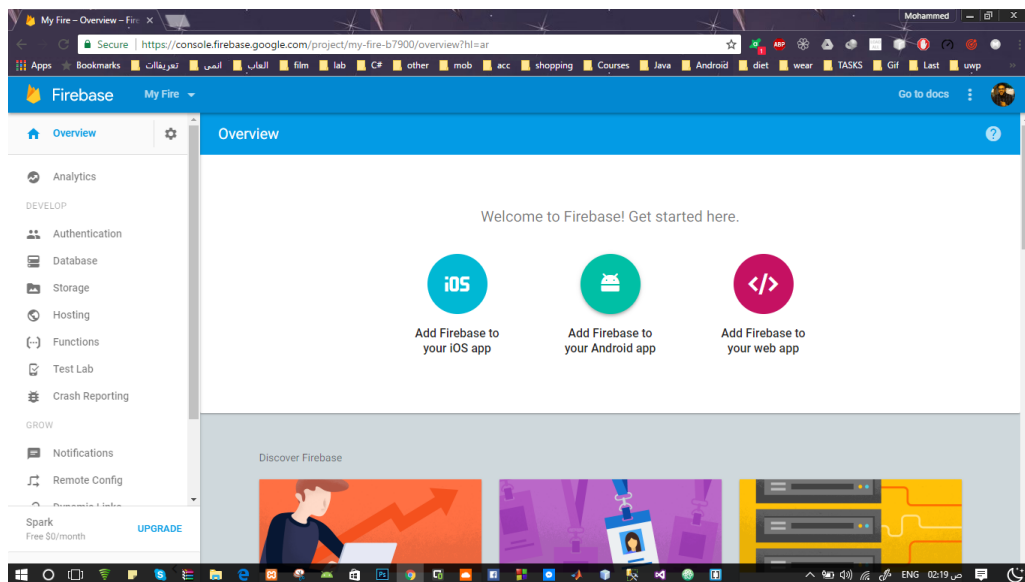


Figure 3.3: Choose Your platform

choose platform which i need to built my app in , in our case we choose 'Android' as platform

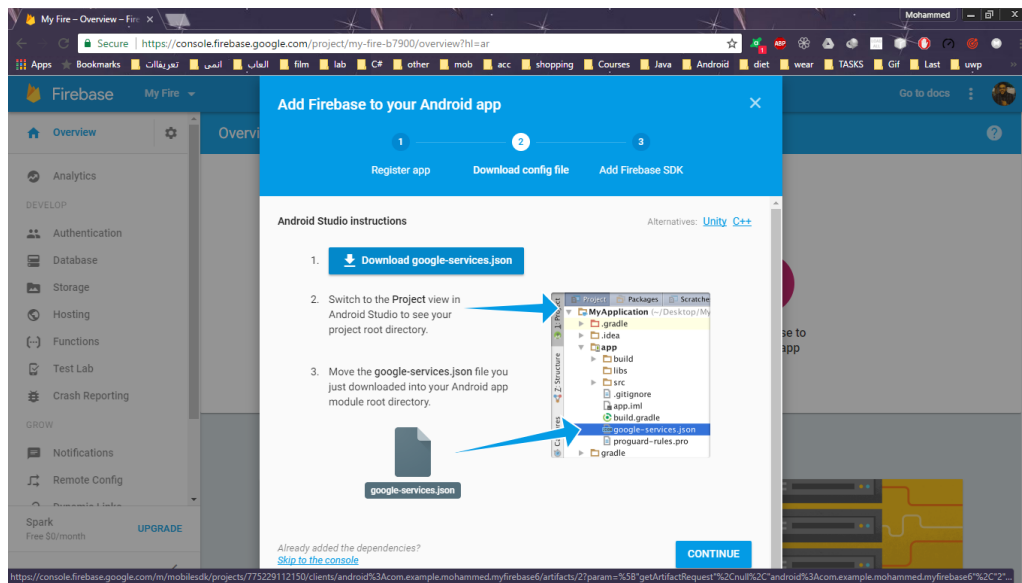


Figure 3.4: Download your json file

3.2 Add Firebase To App

Open Your Android studio add your Json file which your downloaded on your app file

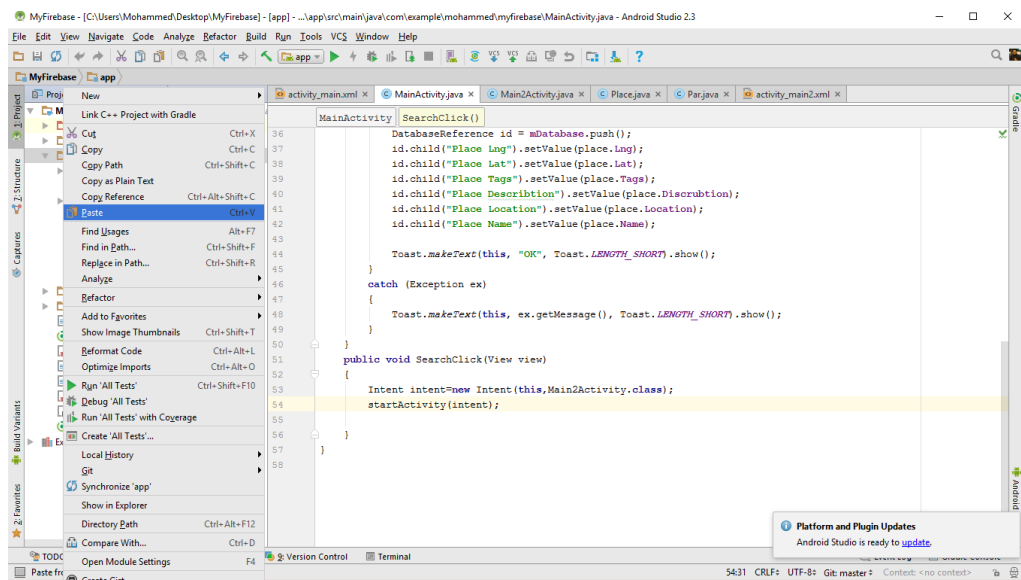


Figure 3.5: Add json file to your app

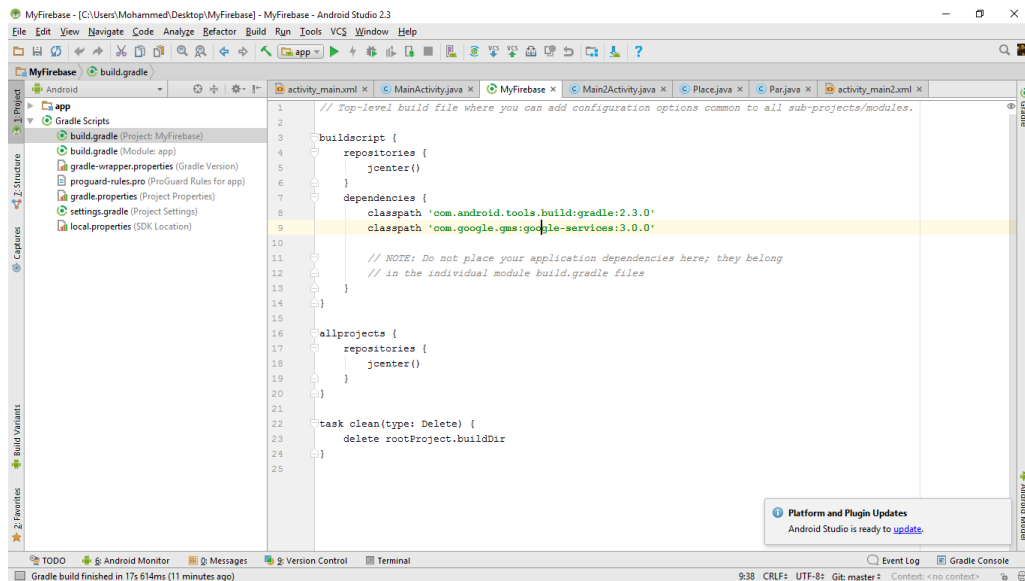


Figure 3.6: Add classpath to Gridle Project

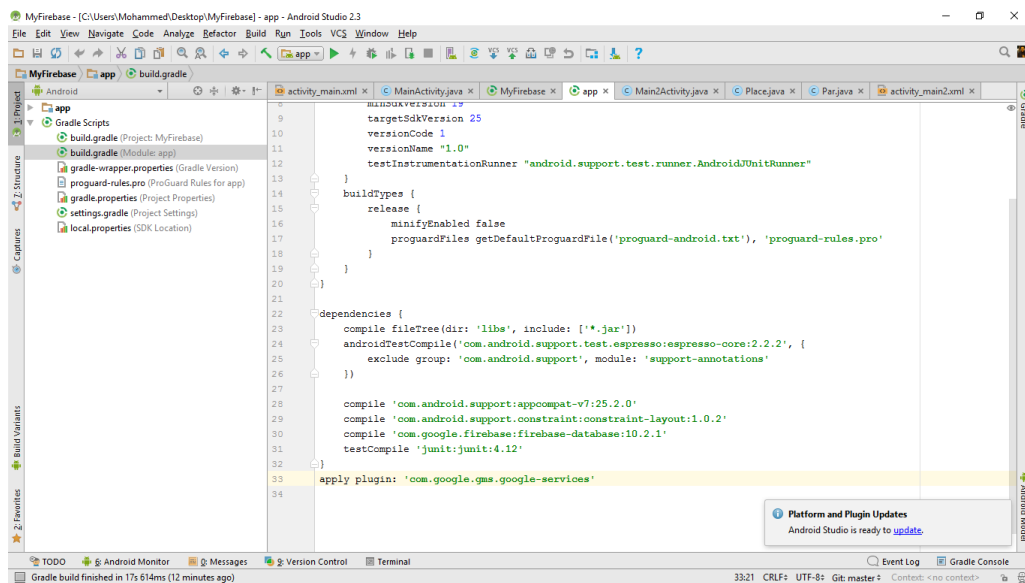


Figure 3.7: Add Compile in Gridle App

your app in android studio connected with your firebase app in firebase console

3.3 Add Network Permission

Go to manifest file and add the permission to access the internet

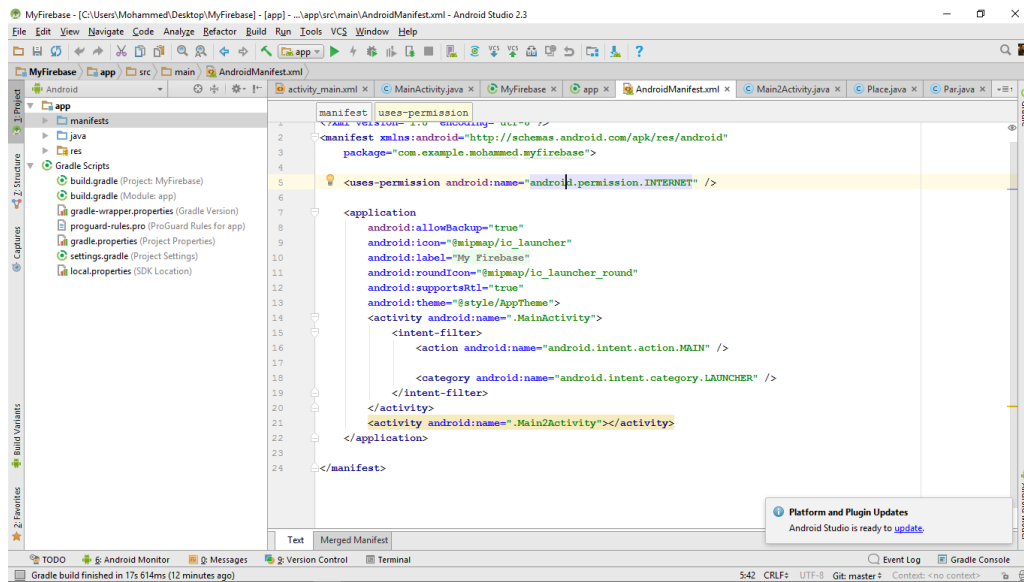


Figure 3.8: Add Compile in Gridle App

Chapter 4

Insert And Retrieve Data

4.1 Insert Data to Firebase Back-end

4.1.1 Create Class Places

Create Class of Places and include the place name , Location , Description , Tags .

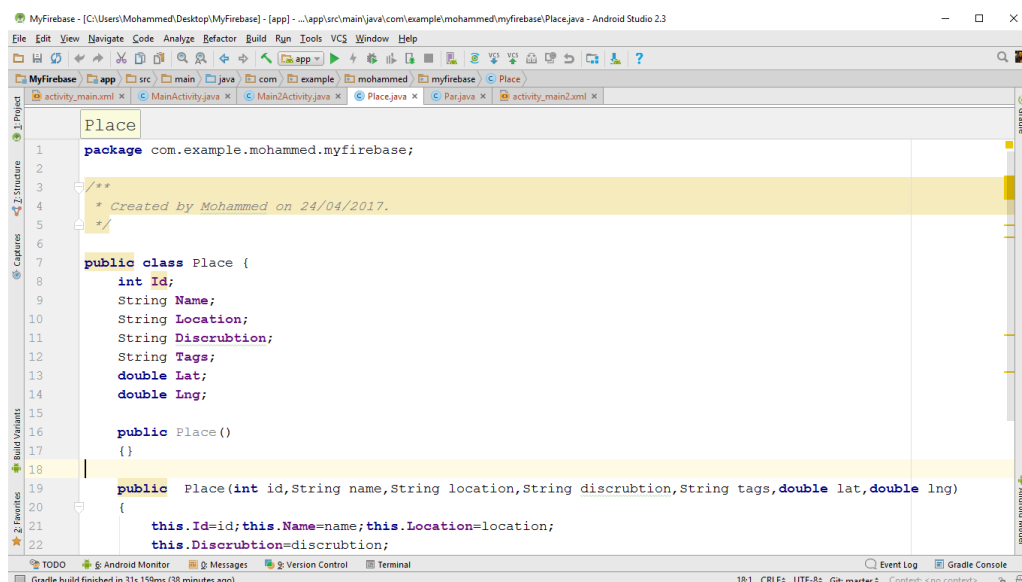


Figure 4.1: Class of Place

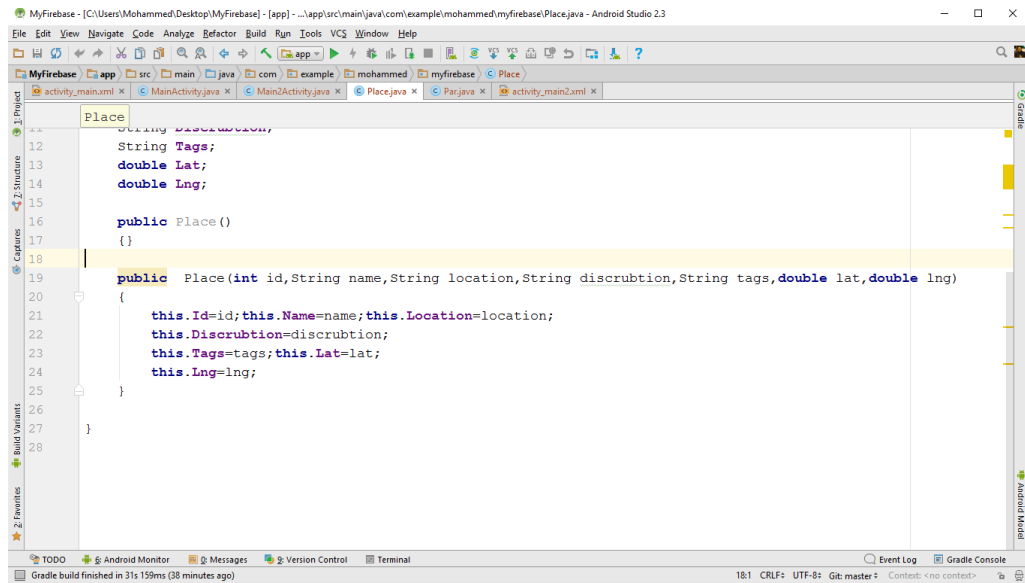


Figure 4.2: Class of Place

4.1.2 Design of Layout

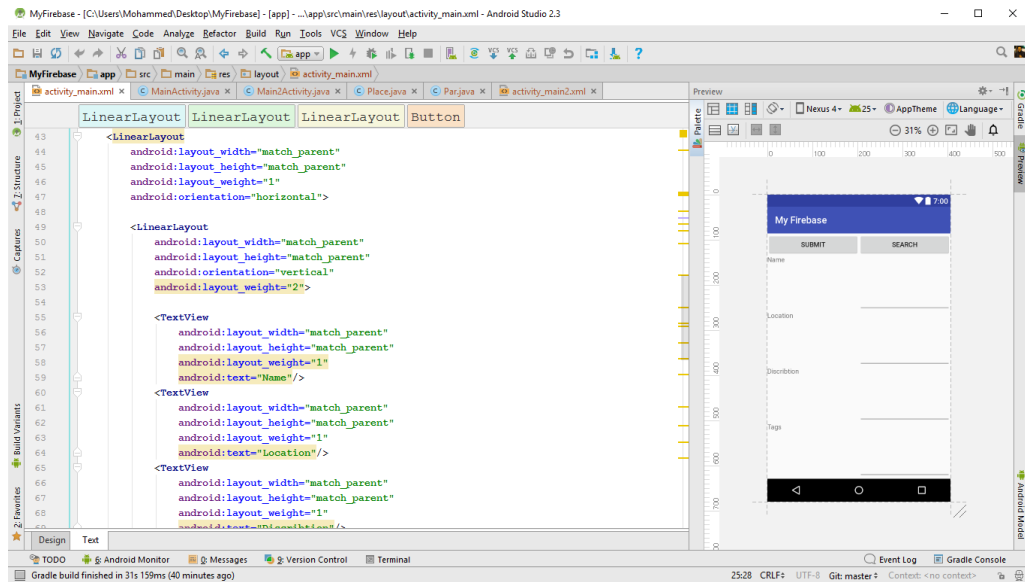


Figure 4.3: Design of layout

4.1.3 Code that Insert Data

Create Object from Database References and set Url to my database root(Places)

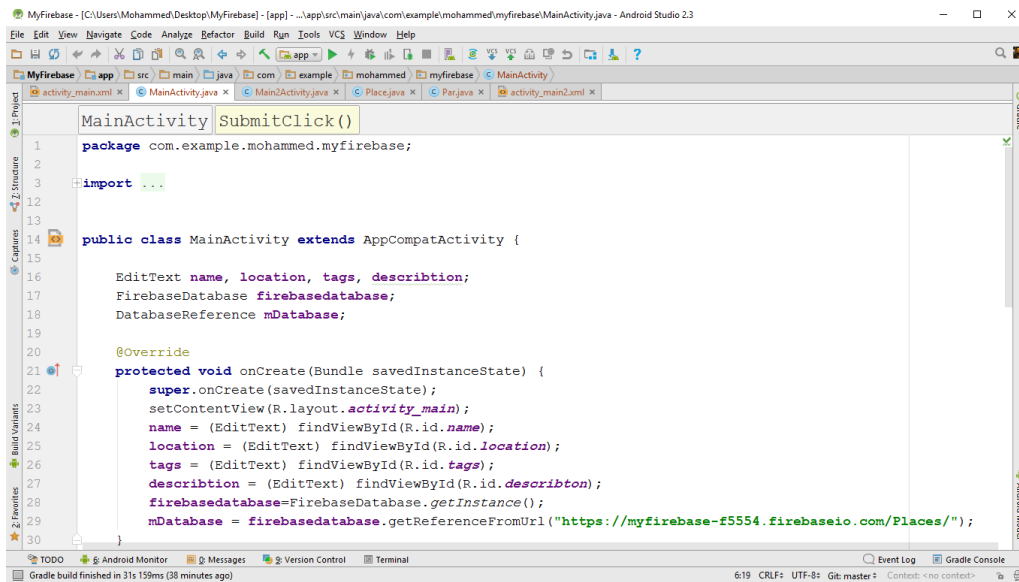


Figure 4.4: Code 1

Make Object from my Class places and set Variable to the place and push it in database back-end

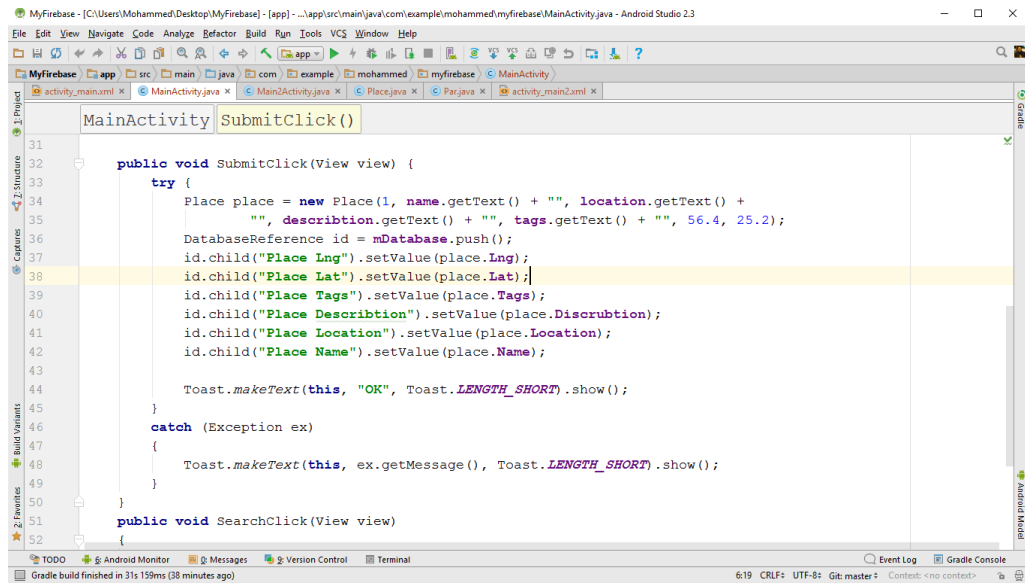


Figure 4.5: Code 2

4.2 Retrieve Data from Firebase Back-end

4.2.1 Design Retrieve layout

Retrieve data in listview and show the name of place only

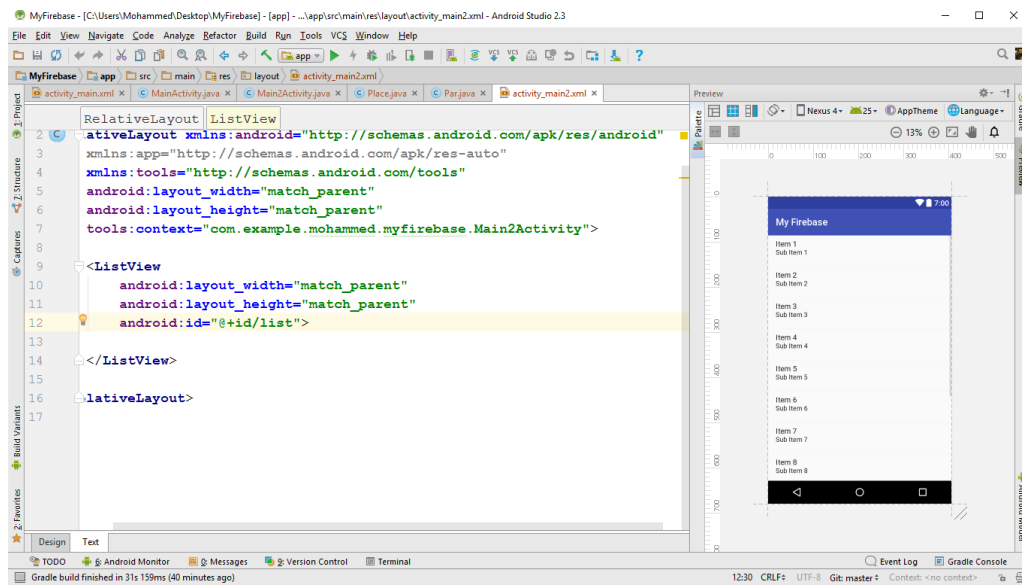


Figure 4.6: Design of layout

4.2.2 Code that retrieve data

Access To root of database (Places) and make 'For-Loob' on all child

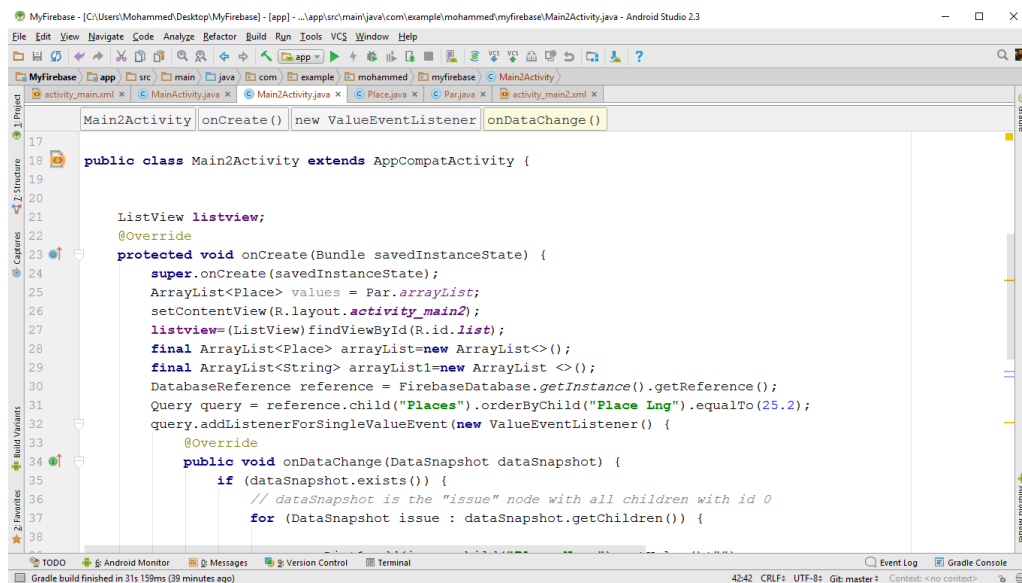


Figure 4.7: Code 1

display the name of place in listview

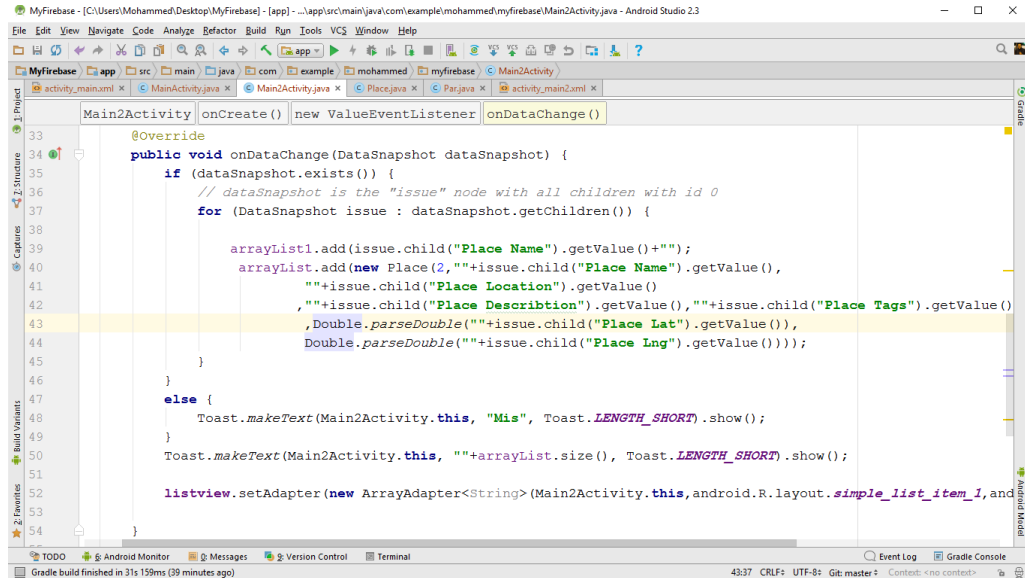


Figure 4.8: Code 2