

Android 애플리케이션 프로그래밍 – firebase, API

Firestore, GoogleMap



부산대학교 정보·의생명공학대학
정보컴퓨터공학부



이론

강의 목표와 구성

❖ Firebase

- Firebase
- 데이터 저장
- 데이터 업데이트
- 데이터 불러오기

❖ GoogleMap

- 내 위치 확인하기
- 지도에 내 위치 보여주기
- 지도 위에 아이콘 표시하기

Firebase

❖ Firebase 란?

- 구글의 모바일 앱 개발을 위한 통합 개발 플랫폼
- 사용 방법
 - [파이어베이스 콘솔 이용\(권장\)](#)
 - 안드로이드 스튜디오에서 **Firestore Assistant**를 이용

❖ Firebase 의 기능

- 서버 기능을 함수 형태로 구현한 **serverless 컴퓨팅**을 가능하게 함
- **데이터 저장 및 조회**를 위한 주요 기능은 웹서버와 유사함
 - **Authentication**: 인증, 회원가입 및 로그인 처리
 - **Cloud Messaging**: 알림 전송
 - **Cloud Firestore**: 모바일 앱 데이터 저장 및 동기화
 - **Realtime Database**: 실시간 데이터 저장 및 동기화
 - **Cloud Storage**: 파일 저장소
 - **Hosting**: 웹 호스팅
 - **Functions**: 서버 관리 없이 모바일 백엔드 코드 실행
 - **Machine Learning**: 모바일 개발자용 머신러닝

❖ Cloud Firestore

- **NoSQL 데이터베이스**로서 테이블과 행이 없고, **컬렉션으로 정리되는 문서에 데이터가 저장됨**
- 문서에는 **키-값 쌍의 데이터가 저장됨**

https://firebase.google.com/products/firestore/?hl=ko&gclid=CjwKCAjwrfCRBhAXEiwAnkmKmUlnJwAf_GZnGzT4MfAVOacD0Zt_HTyYbg984MFDyiCqZRBJSF9k-BoCgBsQAvD_BwE&gclid=aw.ds



출처 : <https://firebase.google.com/codelabs/firestore-web?hl=da#5>

파이어스토어 데이터 모델: 컬렉션과 하위 컬렉션

Firestore 데이터 저장

❖ 데이터 저장

- 데이터 저장을 위해서는 **collection** 을 선택하고 문서 작업을 하는 **CollectionReference** 객체가 필요함
- **Firestore**는 제공되는 라이브러리를 사용해서 접근 할 수 있음

성공 콜백

실패 콜백

```
val db: FirebaseFirestore = FirebaseFirestore.getInstance()

val user = mapOf(
    "name" to "kkang",
    "email" to "a@a.com",
    "avg" to 10
)

val colRef: CollectionReference = db.collection( collectionPath: "users")
val docRef: Task<DocumentReference> = colRef.add(user)
docRef.addOnSuccessListener { documentReference ->
    Log.d( tag: "kkang", msg: "DocumentSnapshot added with ID : " +
        "${documentReference.id}")
}
docRef.addOnFailureListener { e ->
    Log.w( tag: "kkang", msg: "Error adding document", e)
}
```

myfirebase-e643a	users	Elr9n2Owwwe0eK6hT9FE
+ 컬렉션 시작	+ 문서 추가	+ 컬렉션 시작
users >	EIr9n2Owwwe0eK6hT9FE >	+ 필드 추가
		avg: 10
		email: "a@a.com"
		name: "kkang"

Firestore 데이터 업데이트

❖ 데이터 업데이트

- **Set() 함수**는 문서 전체를 덮어쓴다.
- 기존 문서의 특정 필드 값만 업데이트하기 위해서는 **update() 함수**를 이용한다.
- 여러 필드 값을 한번에 변경하고 싶다면 매개변수를 **Map 객체**로 지정한다.

데이터 수정

```
val user2 = User( name: "lee", email: "lee@a.com",  
                  avg: 30, isAdmin: true, isTop: true)  
db.collection( collectionPath: "users")  
  .document( documentPath: "ID01")  
  .set(user2) ①  
  
db.collection( collectionPath: "users")  
  .document( documentPath: "ID01")  
  .update( field: "email", value: "lee@b.com") ②  
  
db.collection( collectionPath: "users")  
  .document( documentPath: "ID01")  
  .update(mapOf(  
    "name" to "lee01",  
    "email" to "lee@c.com"  
  )) ③
```

1	2	3
avg: 30 email: "lee@a.com" isAdmin: true name: "lee" top: true	avg: 30 email: "lee@b.com" isAdmin: true name: "lee" top: true	avg: 30 email: "lee@c.com" isAdmin: true name: "lee01" top: true

Firestore 데이터 불러오기

❖ 데이터 불러오기

- 컬렉션의 전체 문서를 가져올 때는 **collection**을 지정한 다음 **get()** 함수를 호출함
- 단일 문서를 가져오려면 **document()** 메서드에 식별값으로 **문서를 지정**함

```
db.collection( collectionPath: "users")
  .get() ← 데이터 가져오기
  .addOnSuccessListener { result ->
    for( document in result){
      Log.d( tag: "kkang",
            msg: "${document.id} => ${document.data}")
    }
  }
  .addOnFailureListener{exception ->
    Log.d( tag: "kkang", msg: "Error getting documents:", exception)
  }
```

```
D/kkang: EIr9n20wwwe0eK6hT9FE => {avg=10, name=kkang, email=a@a.com}
ID01 => {avg=30, top=true, name=lee, isAdmin=true, email=lee@b.com}
iG5xsizUmggqLxQvgfFv => {avg=20, top=true, name=kim, isAdmin=true, email=kim@a.com}
```

```
val docRef = db.collection( collectionPath: "users").
  document( documentPath: "ID01")
```

googleMap - 위치 제공자

위치 제공자 확인

❖ 사용자 위치

- 위치 얻는 방법
 - 플랫폼 API 이용
 - 구글 play 서비스 라이브러리 이용
- 위치 접근 권한
 - android.permission.ACCESS_COARSE_LOCATION
 - android.permission.ACCESS_FINE_LOCATION
 - android.permission.ACCESS_BACKGROUND_LOCATION

```
val manager = getSystemService(Context.LOCATION_SERVICE)
    as LocationManager

var result = "All providers : "
val providers = manager.allProviders
for (provider in providers){
    result += "$provider, "
}
Log.d( tag: "kkang", result)
```

D/kkang: All providers : passive, gps, network,

❖ 위치 제공자

- **사용자의 위치를 얻기 위해 위치제공자가 필요함**
 - **GPS**: GPS 위성 이용
 - **Network**: 이동 통신망 이용
 - **Wifi**: 와이파이 이용
 - **Passive**: 다른 앱에서 이용한 마지막 위치 정보 이용

```
var result = "Enabled providers : "
val providers = manager.getProviders( enabledOnly: true)
for (provider in providers){
    result += "$provider, "
}
Log.d( tag: "kkang", result)
```

현재 사용 가능한
위치 제공자

D/kkang: Enabled providers :

위치 정보 얻기

❖ 위치 정보 얻기

- 위치 정보를 얻기 위해 **LocationManager의 getLastKnownLocation() 메서드를 이용함**
- getLastKnownLocation() 메서드의 **반환값은 Location**임
 - **getAccuracy():** 정확도
 - **getLatitude():** 위도
 - **getLongitude():** 경도
 - **getTime():** 획득시간
- getLastKnownLocation()는 위치를 한번만 가져올 때 사용하고, **계속 위치를 가져와야 한다면 LocationListener() 이용함**

```
if(ContextCompat.checkSelfPermission(  
    context: this, android.Manifest.permission.ACCESS_FINE_LOCATION)  
    === PackageManager.PERMISSION_GRANTED  
) {  
    val location: Location? = ← 위치 정보 가져오기  
        manager.getLastKnownLocation(LocationManager.GPS_PROVIDER)  
    location?.let { it: Location  
        val latitude = location.latitude  
        val longitude = location.longitude  
        val accuracy = location.accuracy  
        val time = location.time  
  
        Log.d(tag: "kkang2", msg: "$latitude, $longitude, $accuracy, $time")  
    }  
}
```

구글 위치 라이브러리

❖ 구글 위치 라이브러리

- Fused Location Provider의 핵심 요소
 - **FusedLocationProviderClient**: 위치 정보를 얻음
 - **GoogleApiClient**: 위치 제공자 준비 등 다양한 콜백을 제공함

GoogleApiClient
초기화

```
val connectionCallback=object: GoogleApiClient.ConnectionCallbacks {  
    override fun onConnected(p0: Bundle?){  
        // 위치 제공자 사용할 수 있을 때  
    }  
    override fun onConnectionSuspended(p0: Int) {  
        // 위치 제공자 사용할 수 없을 때  
    }  
}  
val onConnectionFailedCallback = object : GoogleApiClient  
    .OnConnectionFailedListener{  
    override fun onConnectionFailed(p0: ConnectionResult) {  
        // 사용할 수 있는 위치 제공자가 없을 때  
    }  
}  
val apiClient: GoogleApiClient = GoogleApiClient.Builder(this)  
    .addApi(LocationServices.API)  
    .addConnectionCallbacks(connectionCallback)  
    .addOnConnectionFailedListener(onConnectionFailedCallback)  
    .build()
```



```
val providerClient : FusedLocationProviderClient =  
    LocationServices.getFusedLocationProviderClient(this)
```

위치 정보 함수
초기화

구글 지도 설정

❖ 사용 설정

- **Gradle Scripts – build.gradle 파일**의 dependencies 항목에 지도 사용 선언
- 구글 지도는 구글 서버에서 전송되는 데이터이므로 **Manifest 파일**에 네트워크 통신 관련 권한 등록
- 지도 API 키 등록

API 키				
이름	생성일	제한사항 ↓	키	작업
✓ API 키 1개	2022. 1. 5.	Android 앱, API 17개	AIzaSyBq4...i47sJMJQ	 

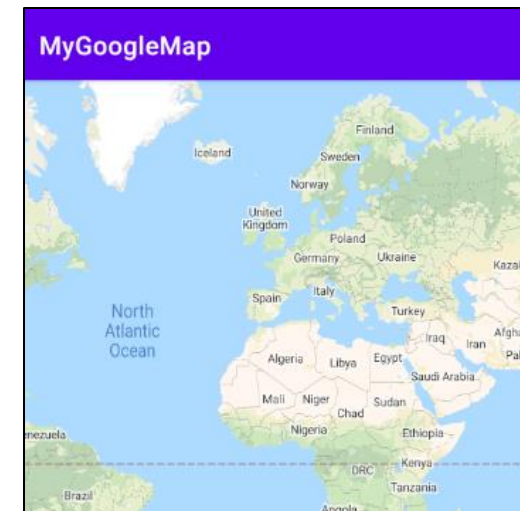
```
implementation 'com.google.android.gms:play-services:12.0.1'
```

Google 지도 사용 선언

네트워크 통신 권한 등록

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

```
<androidx.fragment.app.FragmentContainerView
    android:id="@+id/mapView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.google.android.gms.maps.SupportMapFragment"/>
```



구글 지도 제어

❖ 지도 중심 이동

- 사용자의 위치를 중심으로 지도를 보여줄 수 있어야 함
- 지도가 표시하는 지점을 이동하기 위해서는 **지도**를 출력하는 뷰 객체를 얻어야 함

Fragment 관리자에 접근

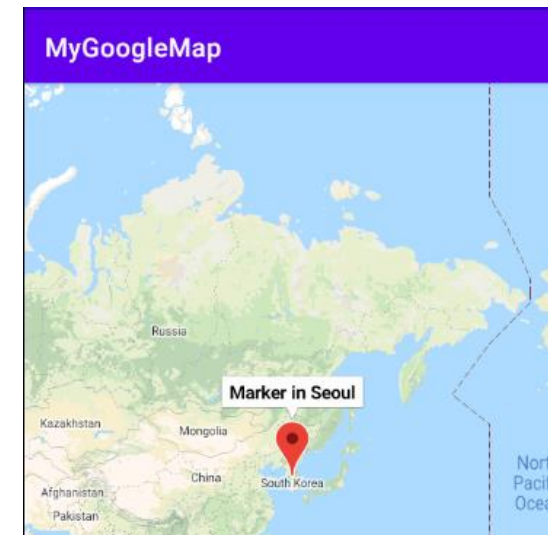
```
val mapFragment = supportFragmentManager  
    .findFragmentById(R.id.mapView) as SupportMapFragment  
mapFragment.getMapAsync(this)
```

❖ 사용자 이벤트 처리

- 앱에서 감지해 처리할 수 있는 이벤트는 다음과 같음
 - OnMapClickListener, OnMapLongClickListener, OnMarkerClickListener...

```
override fun onMapReady(googleMap: GoogleMap) {  
    map = googleMap  
    val seoul = LatLng(37.521814, 126.923596)  
    map.addMarker(MarkerOptions().position(seoul).title("Marker in Seoul"))  
    map.moveCamera(CameraUpdateFactory.newLatLng(seoul))  
}
```

지도 화면 위치 제어



실습

실습 목표와 구성

1. 기초(따라하기)

- Firebase 사용하기
- Google Map 사용하기

2. 응용(로직구현)

- Google Map 를 이용한 현재 위치 표현
- Firestore 데이터 실시간 읽고 쓰기

기초(따라하기) – 예제 1

❖ Firebase 사용하기

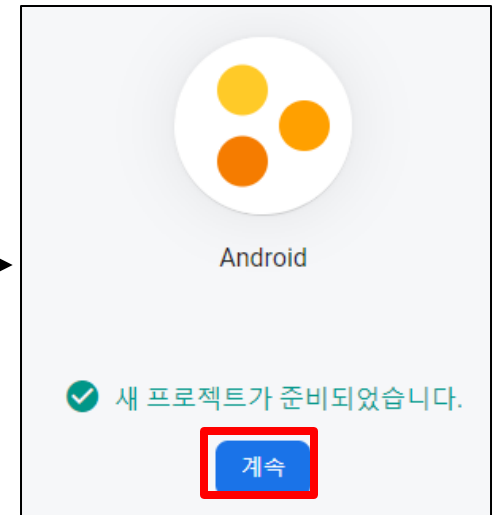
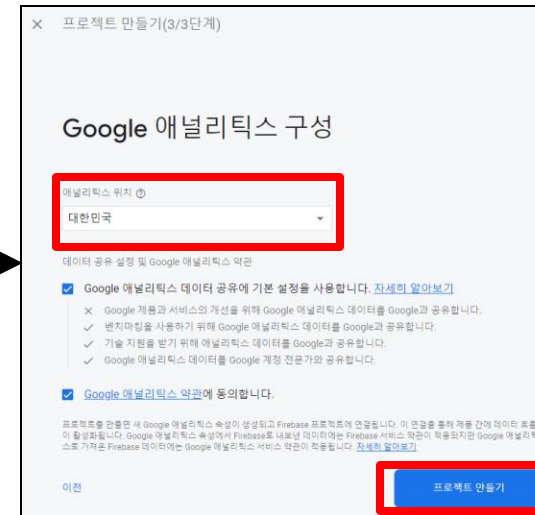
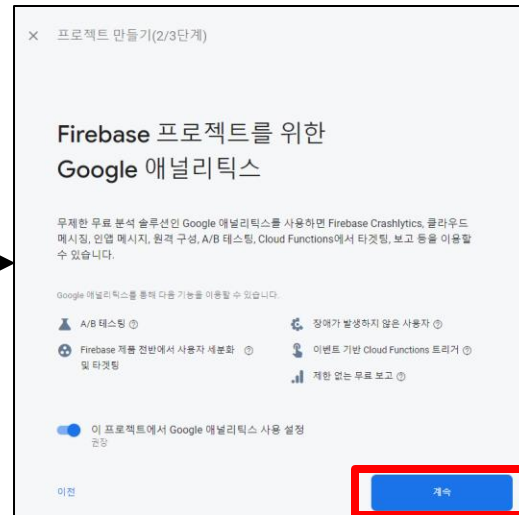
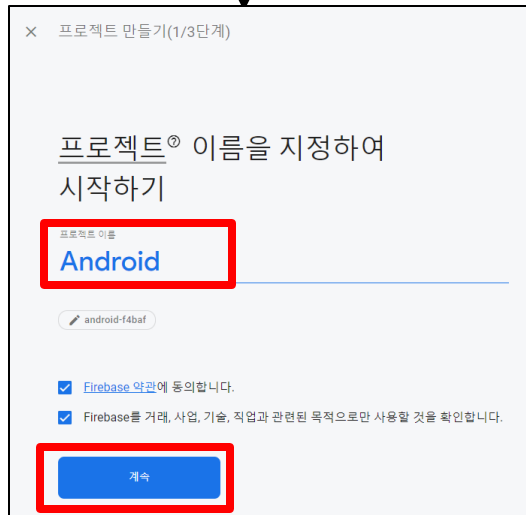
1. Firebase 프로젝트 생성
2. Firebase 데이터베이스 생성
3. Firebase SDK 설정
4. Firebase 연결
5. Firebase 데이터 읽기
6. Firebase 데이터 쓰기



기초(따라하기) – 예제 1

1. Firebase 프로젝트 생성

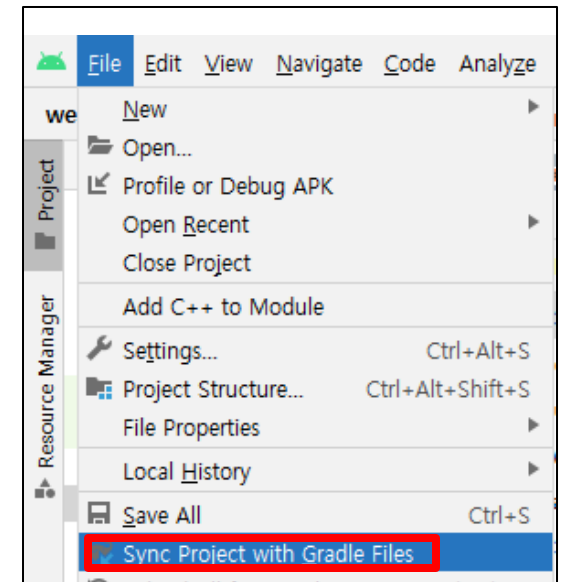
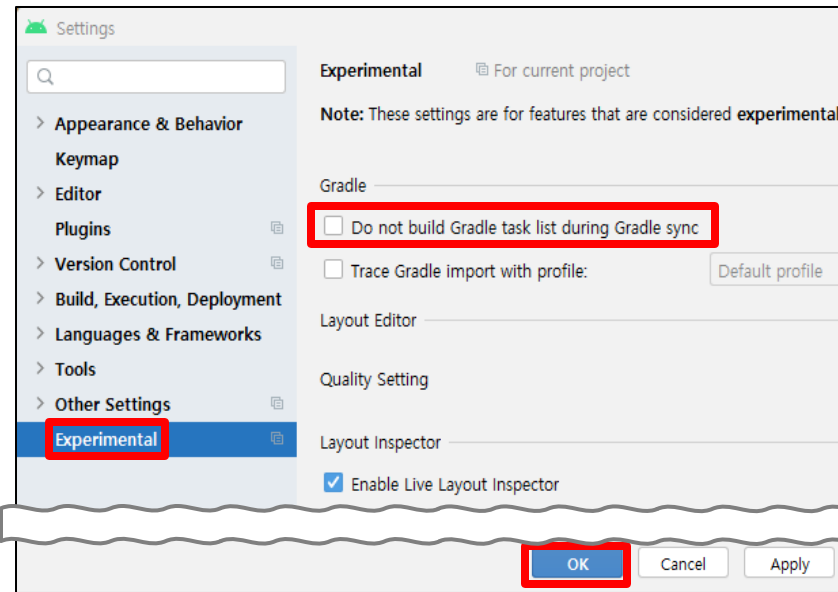
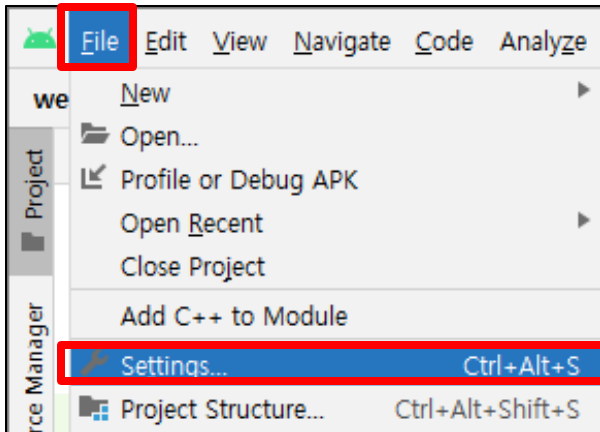
- Console.firebase.google.com 접속(구글 계정 로그인)
- 프로젝트 만들기



기초(따라하기) – 예제 1

1. Firebase 프로젝트 생성

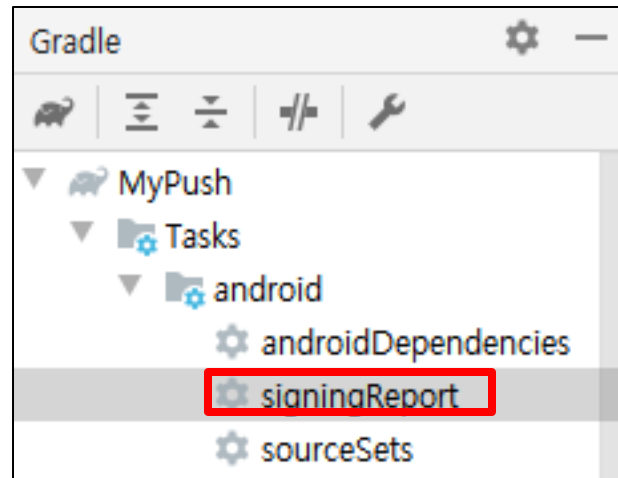
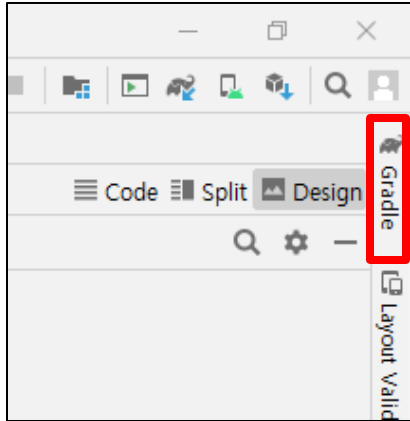
- 숨겨진 Gradle 설정 열기



기초(따라하기) – 예제 1

1. Firebase 프로젝트 생성

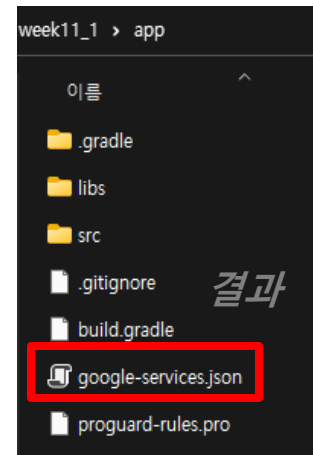
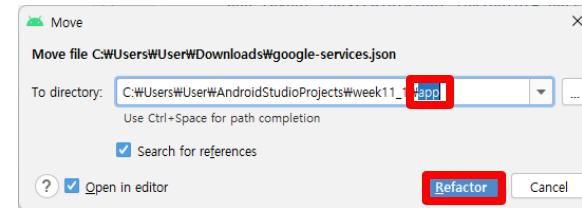
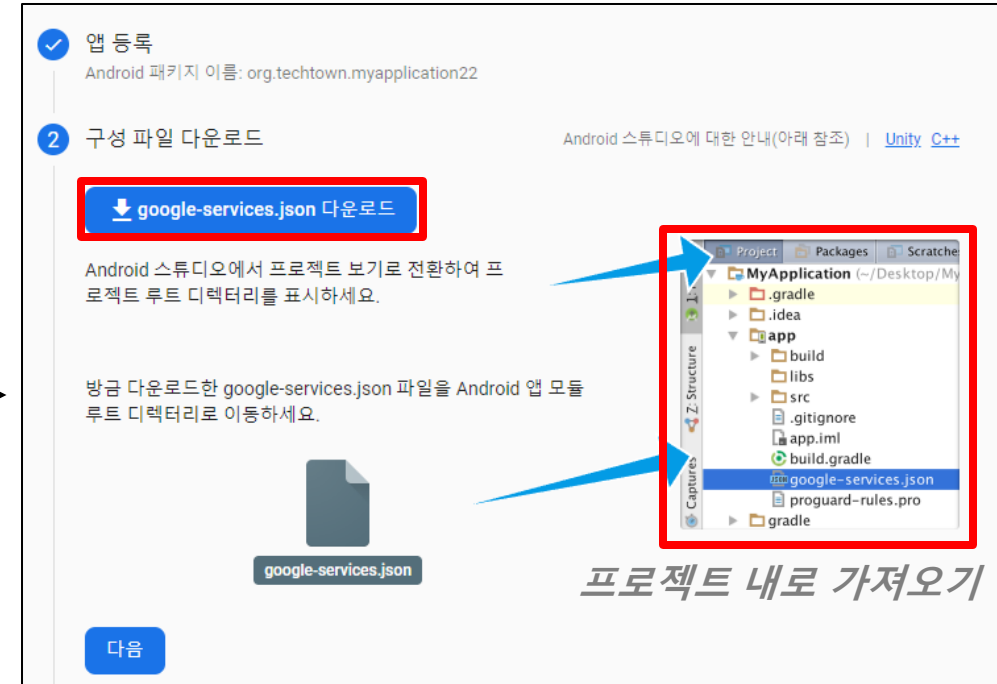
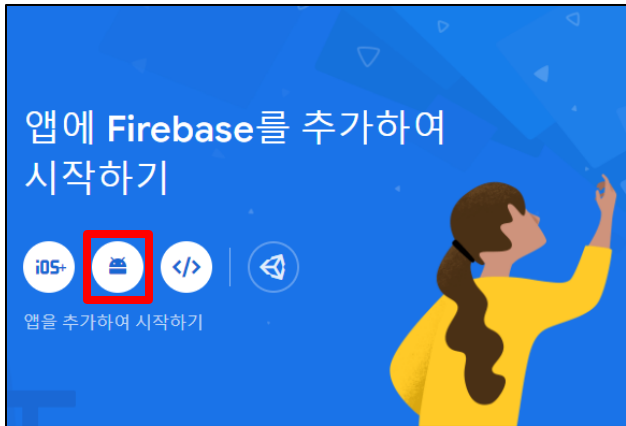
- SigningReport 열기



기초(따라하기) – 예제 1

1. Firebase 프로젝트 생성

- Firebase 프로젝트 생성 및 연결하기
- 프로젝트 내로 가져올 때, app 폴더로 가져오기(드래그)



프로젝트 내로 가져오기

결과

SigningReport

```
Variant: debugAndroidTest  
Config: debug  
Store: C:\Users\User\android\debug.keystore  
Alias: AndroidDebugKey  
MD5: 4C:9A:3A:0B:F0:8D:16:9F:5B:AD  
SHA1: A9:D7:99:3[REDACTED]4:9D:DE:EC:55:52  
SHA-256: 44:93:23:14:[REDACTED]4:EN:47:36:20:7E:BF:A2:4B:D8:  
Valid until: 2051 3 23
```

복사하기

Build.gradle(Module:~)

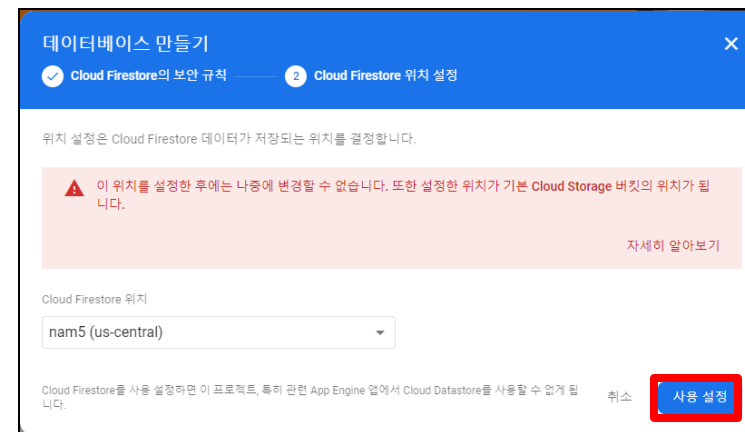
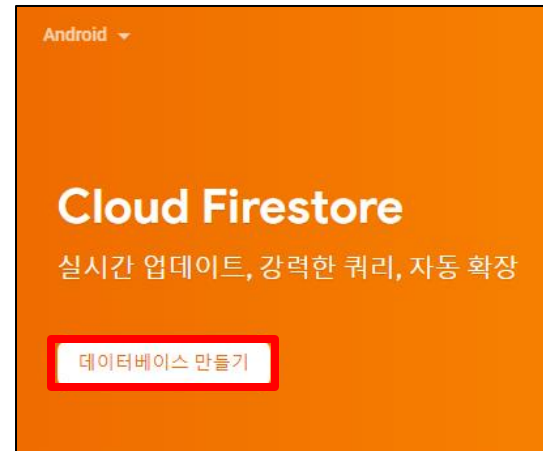
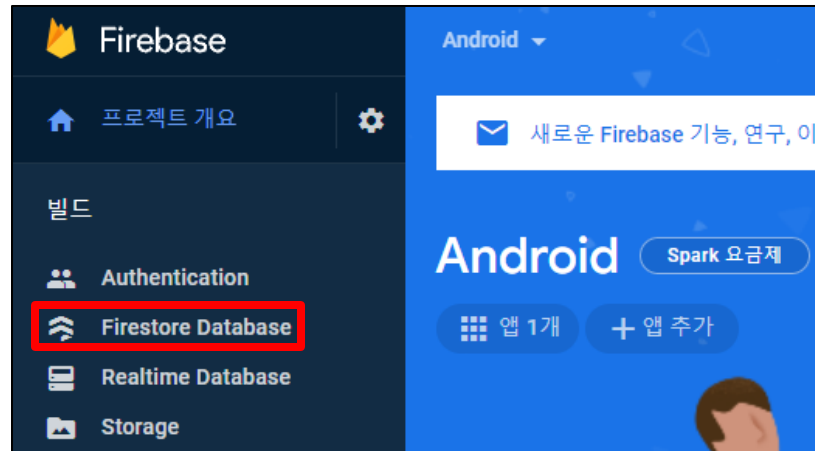
```
6 android {
7     compileSdk 32
8
9     defaultConfig {
10         applicationId "com.example.week11_1"
11         minSdk 21
```

복사하기

기초(따라하기) – 예제 1

2. Firebase 데이터베이스 생성

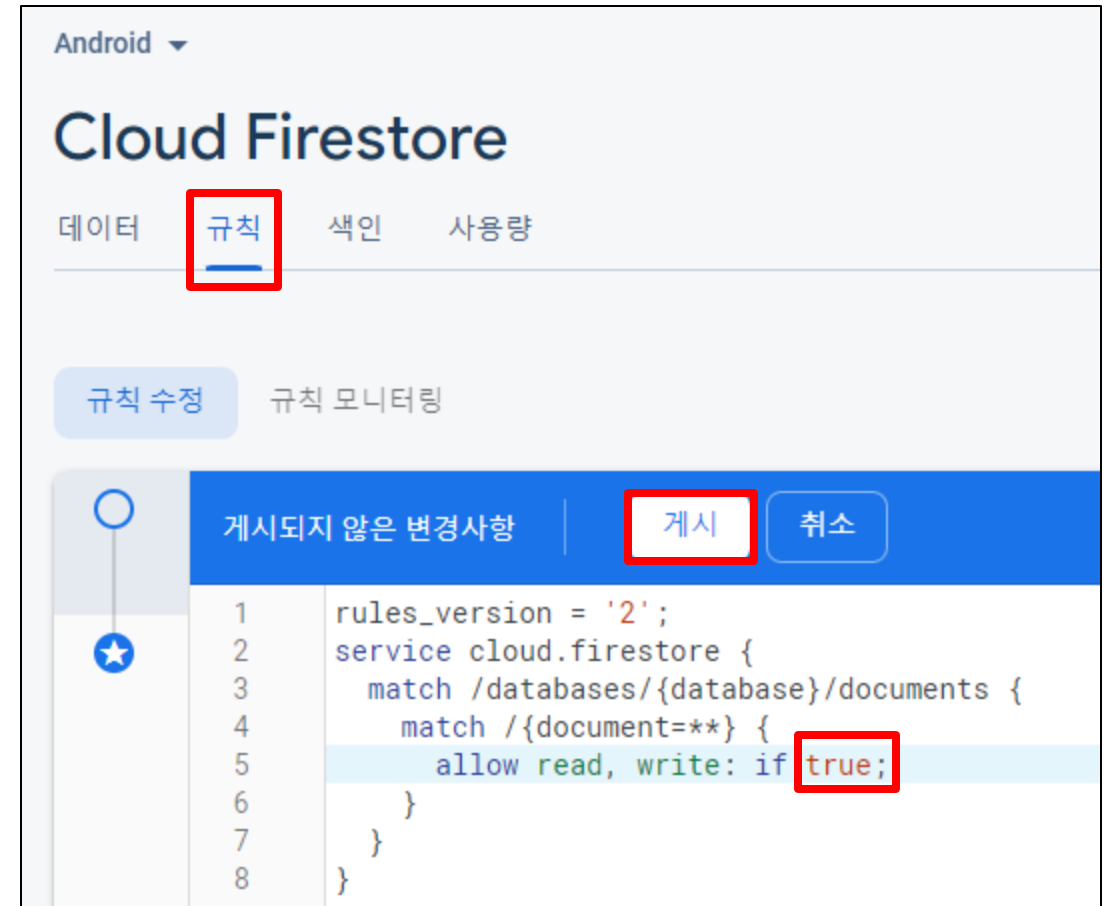
- 데이터베이스 생성하기



기초(따라하기) – 예제 1

2. Firebase 데이터베이스 생성

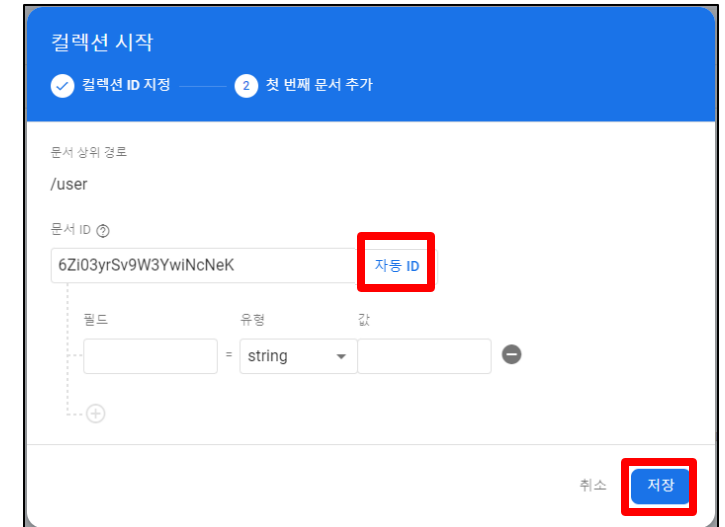
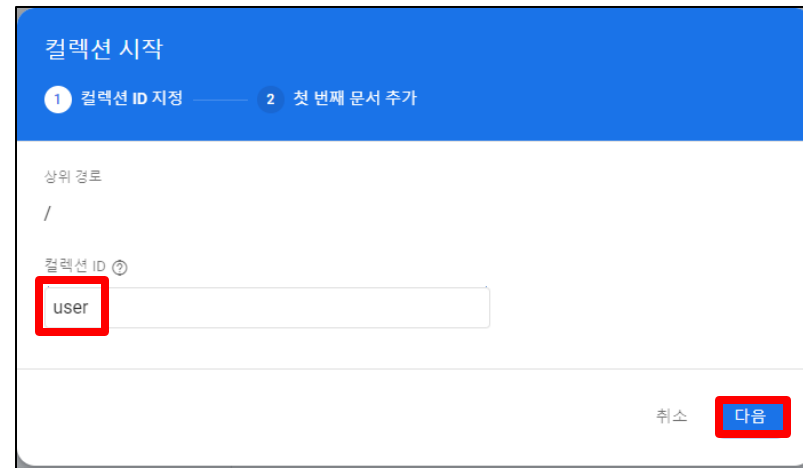
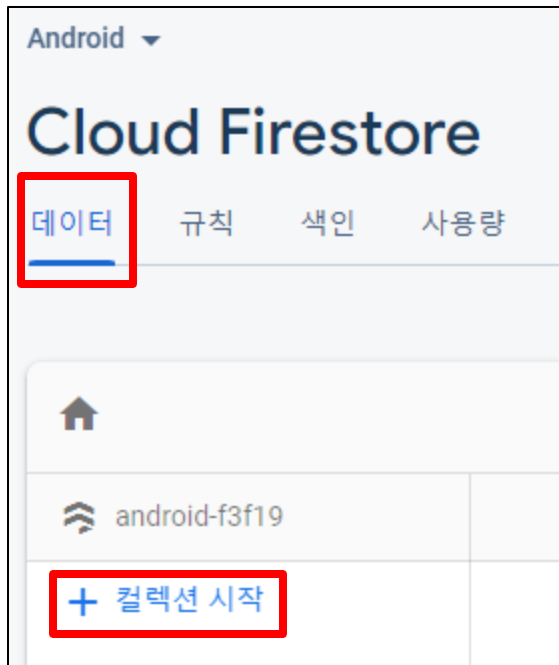
- 규칙 수정하기
 - read, write 권한 허용



기초(따라하기) – 예제 1

2. Firebase 데이터베이스 생성

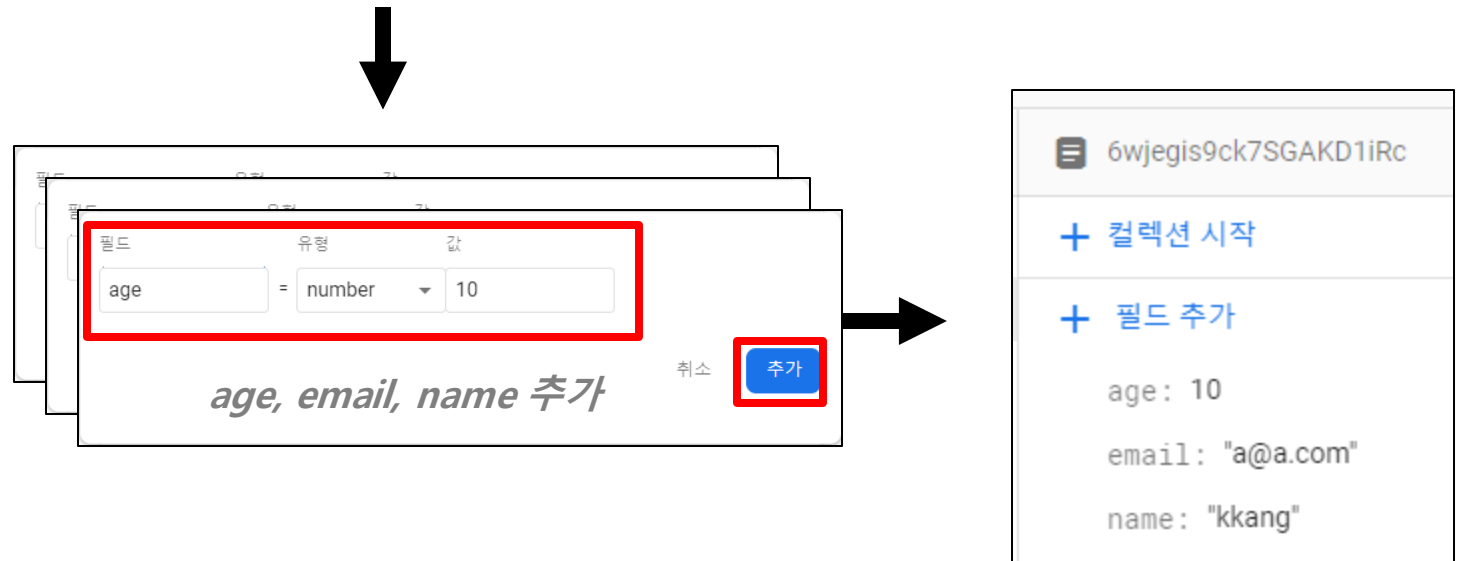
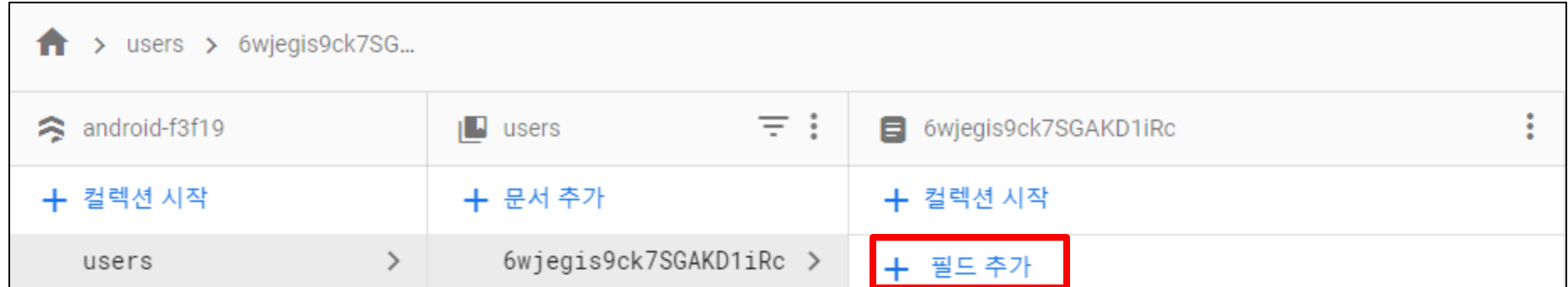
- 컬렉션 시작하기



기초(따라하기) – 예제 1

2. Firebase 데이터베이스 생성

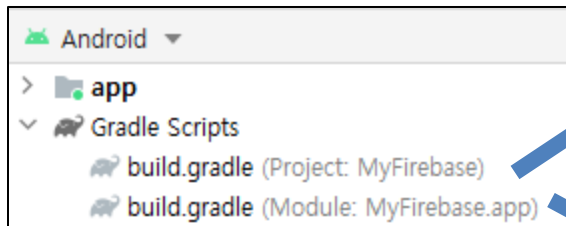
- 필드 추가하기
 - age : number
 - email : string
 - name : string



기초(따라하기) – 예제 1

3. Firebase SDK 설정

- Build.gradle(Module:~) 수정
 - plugins 추가
 - Dependencies 추가
- Build.gradle(Project:~) 수정
 - Dependencies 추가
- Sync Now



Open

Build.gradle(Module:~)

```
1 plugins {  
2     id 'com.android.application'  
3     id 'kotlin-android'  
4     id 'com.google.gms.google-services'  
5     id 'kotlin-android-extensions'
```

```
37 dependencies {  
38     implementation platform('com.google.firebase:firebase-bom:29.3.1')  
39     implementation 'com.google.firebase:firebase-firestore-ktx:21.2.1'  
40     implementation 'androidx.core:core-ktx:1.7.0'  
41     implementation 'androidx.appcompat:appcompat:1.4.1'
```

추가하기

Build.gradle(Project:~)

```
6 dependencies {  
7     classpath "com.android.tools.build:gradle:7.0.4"  
8     classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:1.5.20"  
9     classpath 'com.google.gms:google-services:4.3.3'
```


기초(따라하기) – 예제 1

Mainactivity.kt

4. Firebase 연결

- (13 line) : Firestore 연결
- (19-21 line) : View 연결
- (23-37 line) : SetOnClickListener 구현

```
12 class MainActivity : AppCompatActivity() {
13     val db: FirebaseFirestore = FirebaseFirestore.getInstance()
14
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         setContentView(R.layout.activity_main)
18
19         val EditName = findViewById<EditText>(R.id.name)
20         val EditEmail = findViewById<EditText>(R.id.email)
21         val EditAge = findViewById<EditText>(R.id.age)
22
23         readBtn.setOnClickListener { it: View!
24             readFirebase()
25         }
26
27         writeBtn.setOnClickListener { it: View!
28             var Name : String
29             var Email : String
30             var Age : Int
31
32             if(EditName.length() == 0) Name = "null" else Name = EditName.text.toString()
33             if(EditEmail.length() == 0) Email = "null" else Email = EditEmail.text.toString()
34             if(EditAge.length() == 0) Age = 0 else Age = EditAge.text.toString().toInt()
35
36             writeFirebase(Name, Email, Age)
37         }
38     }
```

기초(따라하기) – 예제 1

Mainactivity.kt

5. Firebase 데이터 읽기

- (42 line) : collection 객체 연결
- (43 line) : 데이터 가져오기 함수
- 리스너를 통한 요청 결과 확인
 - (44 line) : addOnSuccessListener 기능 구현
 - (48 line) : addOnFailureListener 기능 구현

```
40 fun readFirebase()
41 {
42     db.collection(collectionPath: "users")
43     .get()
44     .addOnSuccessListener {
45         result -> for (document in result)
46             output1.append("${document.data} \n")
47     }
48     .addOnFailureListener { it: Exception
49         output1.append("Failure \n")
50     }
51 }
```

기초(따라하기) – 예제 1

6. Firebase 데이터 쓰기

- (60 line) : collection 생성 혹은 연결
- (61 line) : document 추가
- (62-67 line) : 리스너 기능 구현

Mainactivity.kt

```
53 fun writeFirebase(Name: String, Email: String, Age : Int)
54 {
55     val user = mapOf(
56         "name" to Name,
57         "email" to Email,
58         "age" to Age
59     )
60     val colRef: CollectionReference = db.collection( collectionPath: "Users")
61     val docRef: Task<DocumentReference> = colRef.add(user)
62     docRef.addOnSuccessListener {
63         documentReference -> output1.append("Success : " + "${documentReference.id} \n")
64     }
65     docRef.addOnFailureListener{ it: Exception
66         output1.append("Failure \n")
67     }
68 }
69 }
```

기초(따라하기) – 예제 2

❖ Google Map 사용하기

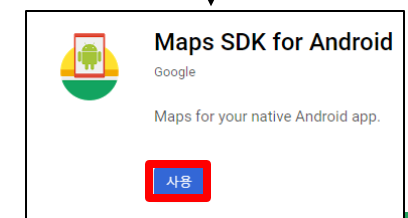
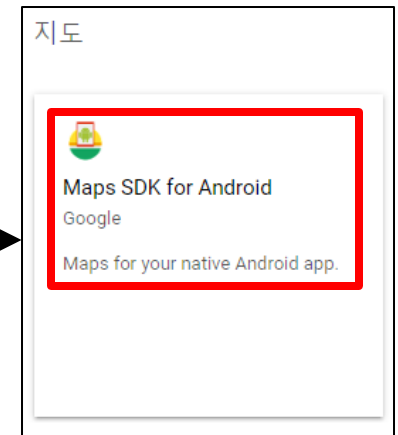
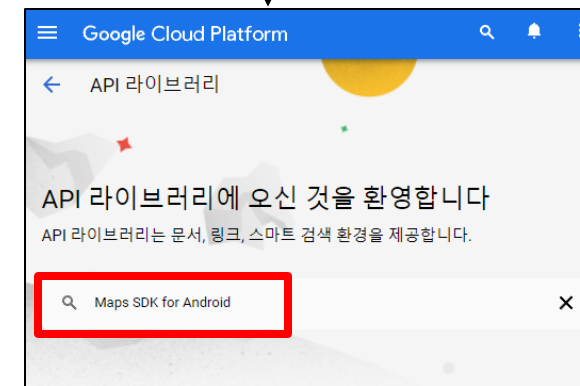
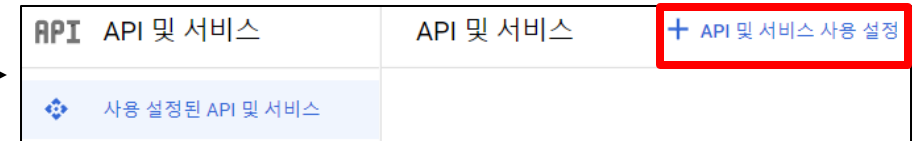
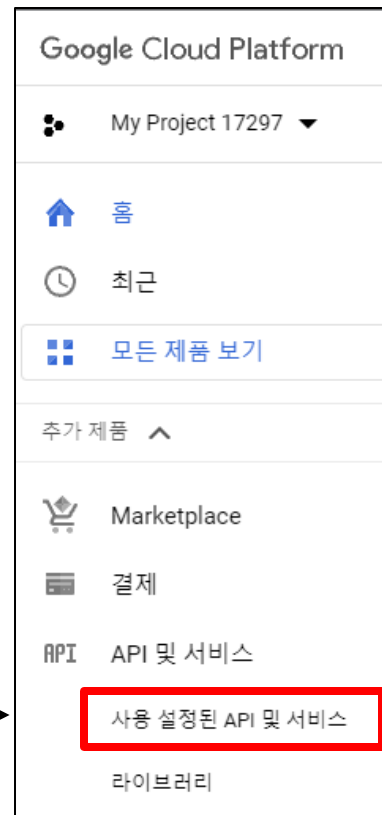
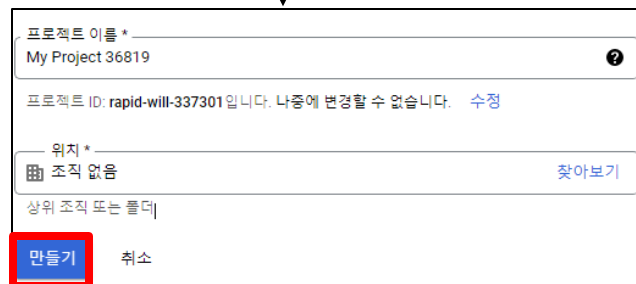
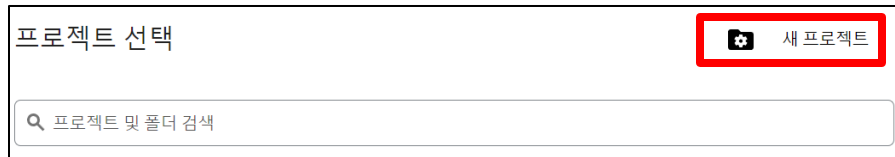
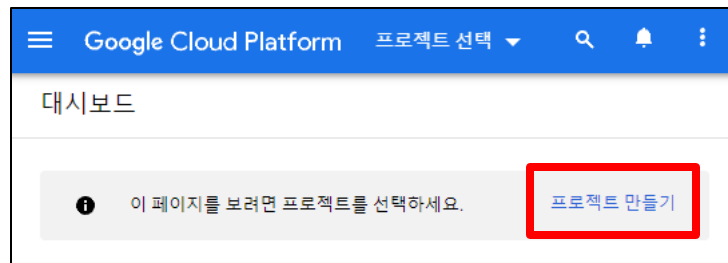
1. Google Map API 키 발급
2. Google Map API 관련 프로젝트 설정
3. Google Map 레이아웃
4. Google Map 제어



기초(따라하기) – 예제 2

❖ Google Map API 키 발급

- 구글 개발자 콘솔 접속 및 프로젝트 생성
 - console.cloud.google.com
- API 및 서비스 – 라이브러리 생성
 - Maps SDK for Android



기초(따라하기) – 예제 2

❖ Google Map API 키 발급

- API 및 서비스에서 '사용자 인증 정보'를 선택한다.
- '사용자 인증 정보 만들기'를 선택하고 'API 키'를 선택한다.
- 지도 API 키를 안드로이드 앱에서 사용하려면 키 제한을 걸어야 한다.



기초(따라하기) – 예제 2

❖ Google Map API 관련 프로젝트 설정

- (35 line) : 구글 지도 사용 선언을 위한 dependencies 항목 추가
- (4-6 line) : Google Map 사용을 위한 관련 권한 부여
- (24 line) : 데이터 전송 에러 방지
- (25-26 line) : 구글 API 설정

build.gradle(Module:~)

```
34 dependencies {  
35     implementation 'com.google.android.gms:play-services:12.0.1'  
36     implementation 'androidx.core:core-ktx:1.7.0'
```

Manifests.xml

```
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"  
3     package="com.example.week11_2">  
4     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>  
5     <uses-permission android:name="android.permission.INTERNET"/>  
6     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
  
24     <uses-library android:name="org.apache.http.legacy" android:required="false" />  
25     <meta-data android:name="com.google.android.maps.v2.API_KEY" android:value="API Key" />  
26     <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version" />  
27 </application>
```


기초(따라하기) – 예제 2

❖ Google Map 레이아웃

- (1-6 line) : Paly-services 라이브러리에서 지도는 fragment로 제공하므로 지도를 출력할 화면에 fragment로 등록한다.

activity_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <fragment xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/mapView"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:name="com.google.android.gms.maps.SupportMapFragment"/>
```

기초(따라하기) – 예제 2

❖ Google Map 제어

- (18-21 line) : `getMapAsync()` 함수를 이용해 지도 객체를 등록.
- (23-28line) : 지도 객체를 사용할 수 있을 때 `onMapReady()` 함수가 자동으로 호출.
- (25 line) : `LatLng` 객체를 이용해 특정 위치를 표현.
- (27 line) : `CameraUpdateFactory` 함수를 사용해 `LatLng` 객체가 표현하는 위치 출력.

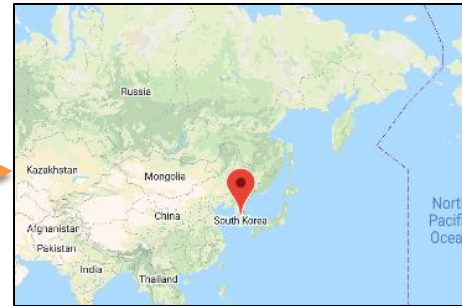
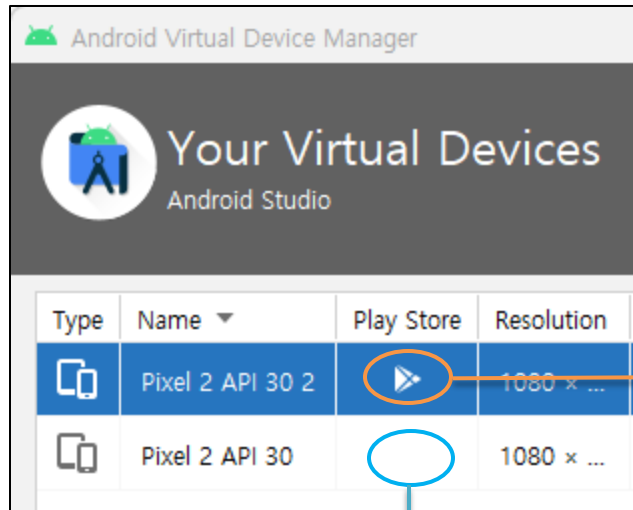
Mainactivity.kt

```
12 class MainActivity : AppCompatActivity(), OnMapReadyCallback {
13     private lateinit var map: GoogleMap
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         setContentView(R.layout.activity_main)
17
18         val mapFragment = supportFragmentManager
19             .findFragmentById(R.id.mapView) as SupportMapFragment
20         mapFragment.getMapAsync(this)
21     }
22
23     override fun onMapReady(googleMap: GoogleMap) {
24         map = googleMap
25         val seoul = LatLng(37.521814, 126.923596)
26         map.addMarker(MarkerOptions().position(seoul).title("Marker in Seoul"))
27         map.moveCamera(CameraUpdateFactory.newLatLng(seoul))
28     }
29 }
```

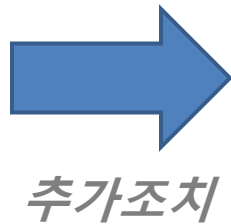
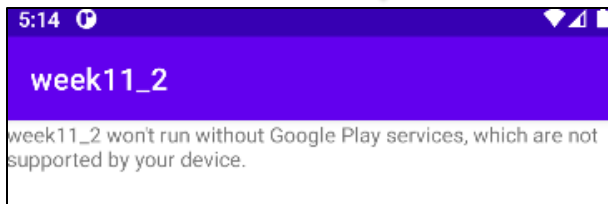
기초(따라하기) – 예제 2

❖ Google Map 제어

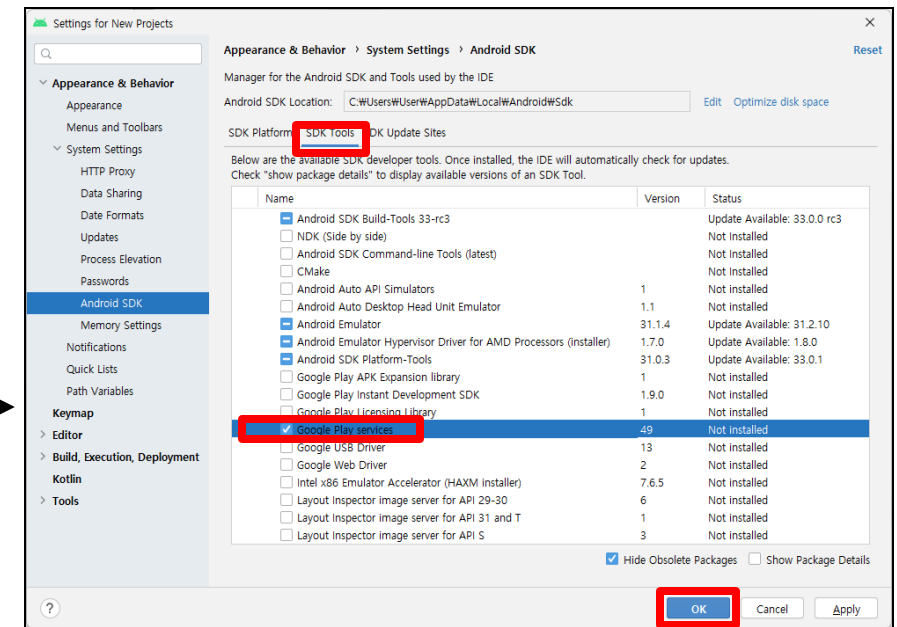
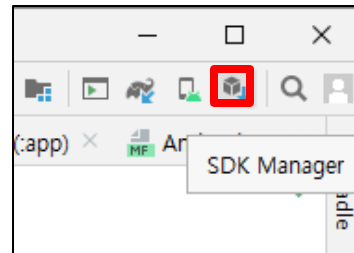
- Play Store 지원 AVM 으로 실행
- Play Store 미지원시, SDK 추가 설치 필요



실행결과



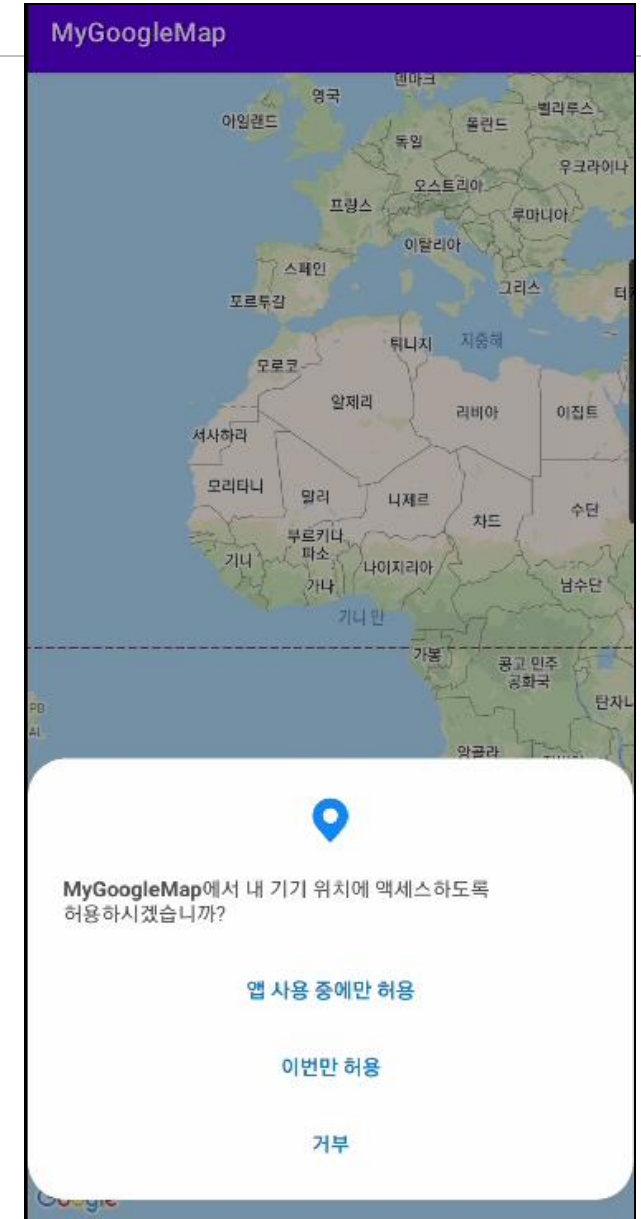
추가조치



응용(로직구현) – 예제 3

❖ Google Map 를 이용한 현재 위치 표현

1. Google Map API 관련 프로젝트 설정
2. AVM 위치 설정
3. Google Map 객체 생성
4. 획득 권한 확인 및 권한 요청
5. 위/경도 값 사용하기



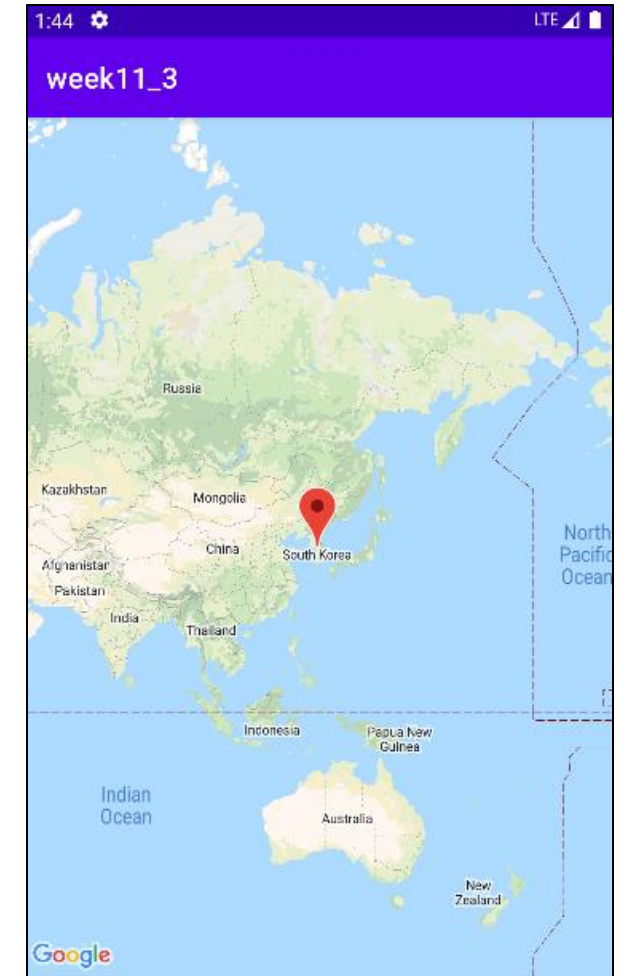
응용(로직구현) – 예제 3

1. Google Map API 관련 프로젝트 설정

- 예제 2 선행 필요
 - 우측 정상 실행 화면 확인 후 예제 3 진행
- (4-5 line) : 위치 권한과 관련된 내용을 추가한다.

Manifests.xml

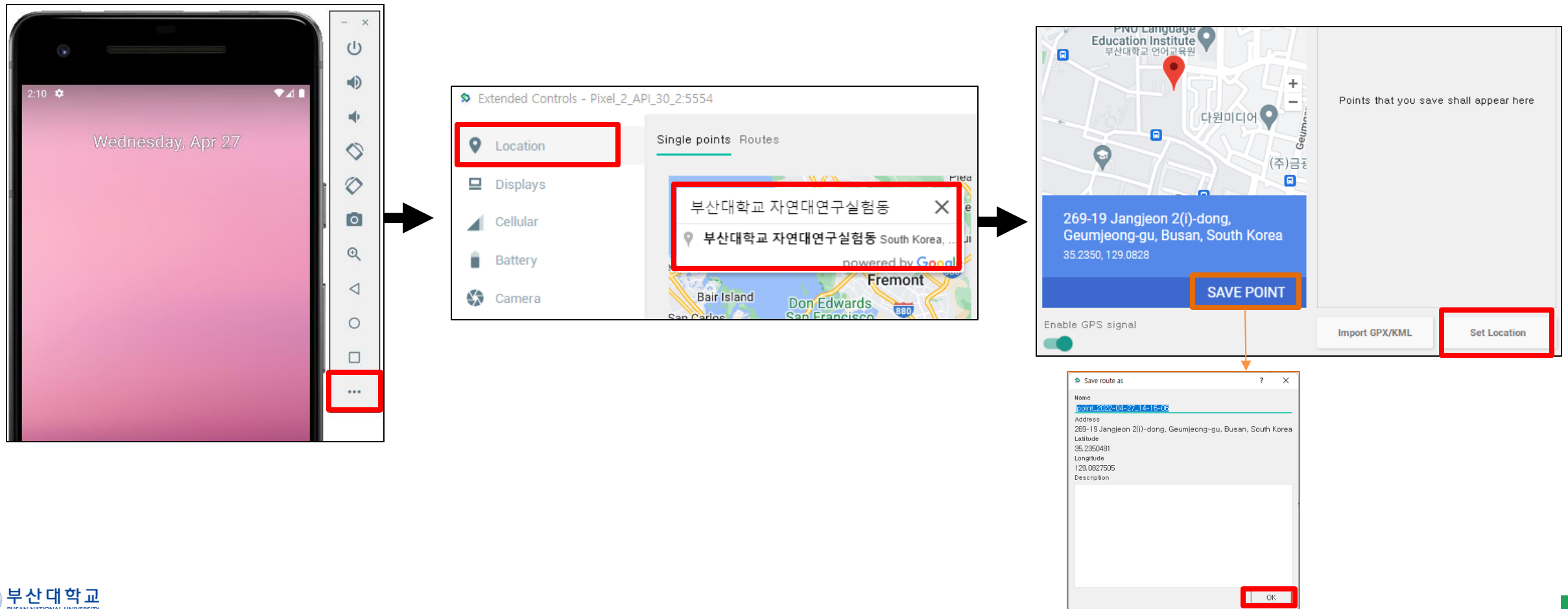
```
4 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
5 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
6 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```



응용(로직구현) – 예제 3

2. AVM 위치 설정

- AVM 상의 현재 위치 설정
 - 본인이 가고 싶은 장소로 설정
 - 실제 디바이스일 경우 GPS 상 현 위치 고정



응용(로직구현) – 예제 3

3. Google Map 객체 생성

- 예제 2 MainActivity.kt 코드 제거
- (24 line) : 현재 위치 검색 객체
- (25 line) : 위치 요청 반환 객체
- (30-33 line) : 필요한 권한 추가하기
 - ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATION

MainActivity.kt

```
22  class MainActivity : AppCompatActivity(), OnMapReadyCallback {
23      private lateinit var mMap: GoogleMap
24      private lateinit var fusedLocationClient: FusedLocationProviderClient
25      private lateinit var locationCallback: LocationCallback
26      override fun onCreate(savedInstanceState: Bundle?) {
27          super.onCreate(savedInstanceState)
28          setContentView(R.layout.activity_main)
29
30          val permissions = arrayOf(
31              android.Manifest.permission.ACCESS_COARSE_LOCATION,
32              android.Manifest.permission.ACCESS_FINE_LOCATION)
33          requirePermissions(permissions, requestCode: 999)
34      }
```

응용(로직구현) – 예제 3

4. 획득 권한 확인 및 권한 요청

- (36 line) : 프래그먼트 연결 및 지도 기능 시작
 - 지도 준비 완료 알람 획득
- (42 line) : 버전 관리 코드
- (45 line) : 권한 획득 여부 확인
- (51 line) : 권한 요청 팝업

MainActivity.kt

```
36 fun startProcess() {
37     val mapFragment = supportFragmentManager.findFragmentById(R.id.mapView) as SupportMapFragment
38     mapFragment.getMapAsync(this)
39 }
40
41 fun requirePermissions(permissions: Array<String>, requestCode: Int) {
42     if(Build.VERSION.SDK_INT < Build.VERSION_CODES.M) {
43         permissionGranted(requestCode)
44     } else {
45         val isAllPermissionsGranted = permissions.all { it: String
46             checkSelfPermission(it) == PackageManager.PERMISSION_GRANTED
47         }
48         if (isAllPermissionsGranted) {
49             permissionGranted(requestCode)
50         } else {
51             ActivityCompat.requestPermissions(activity: this, permissions, requestCode)
52         }
53     }
54 }
```


응용(로직구현) – 예제 3

MainActivity.kt

5. 권한 요청 결과 반환

- (80-82 line): 권한이 있는 경우
 - 구글 지도를 보여주는 코드 실행
- (84-89 line): 권한이 없는 경우 안내
 - 권한 승인이 요구됨을 알리는 Toast 메시지 출력
- (91-95 line): onMapReady함수로부터 현재 위치 획득

```
67 override fun onRequestPermissionsResult(  
68     requestCode: Int,  
69     permissions: Array<out String>,  
70     grantResults: IntArray  
71 ) {  
72     super.onRequestPermissionsResult(requestCode, permissions, grantResults)  
73     if (grantResults.all { it == PackageManager.PERMISSION_GRANTED }) {  
74         permissionGranted(requestCode)  
75     } else {  
76         permissionDenied(requestCode)  
77     }  
78 }  
  
80 fun permissionGranted(requestCode: Int) {  
81     startProcess()  
82 }  
83  
84 fun permissionDenied(requestCode: Int) {  
85  
86  
87  
88  
89 }  
  
91 override fun onMapReady(googleMap: GoogleMap) {  
92     mMap = googleMap  
93     fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)  
94     updateLocation()  
95 }
```

Toast 메시지 출력

응용(로직구현) – 예제 3

6. 위/경도 값 사용하기

- (97 line) : 115 line의 자동완성(에러 보정) 기능을 사용해서 해당코드 삽입
- (100-103 line) : 1초 간격으로 사용자의 위치 갱신
- (108-111 line) : 사용자의 위도와 경도 값 받기
- (109 line) : 위도, 경도를 Log 혹은 Toast로 출력

MainActivity.kt

```
97  @SuppressWarnings("MissingPermission")
98  fun updateLocation() {
99      val locationRequest = LocationRequest.create()
100     locationRequest.run { this: LocationRequest!
101         priority = LocationRequest.PRIORITY_HIGH_ACCURACY
102         interval = 1000
103     }
104
105     locationCallback = object : LocationCallback() {
106         override fun onLocationResult(locationResult: LocationResult?) {
107             locationResult?.let { it: LocationResult
108                 for(location in it.locations) {
109                     위/경도 출력
110                     setLastLocation(location)
111                 }
112             }
113         }
114     }
115     fusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, Looper.myLooper())
116 }
```

응용(로직구현) – 예제 3

6. 위/경도 값 사용하기

- (120-122 line): 사용자의 위치 표시 Marker 생성
 - Marker title : Here!
- (124-130 line): 전달받은 위도, 경도가 나타내는 곳으로 전환하여 화면에 출력

MainActivity.kt

```
118 fun setLastLocation(lastLocation: Location) {
119     val LATLNG = LatLng(lastLocation.latitude, lastLocation.longitude)
120     val markerOptions =
121
122
123
124     val cameraPosition = CameraPosition.Builder()
125         .target(LATLNG)
126         .zoom(15.0f)
127         .build()
128     mMap.clear()
129     mMap.addMarker(markerOptions)
130     mMap.moveCamera(CameraUpdateFactory.newCameraPosition(cameraPosition))
131 }
```



사용자의 위치 표시 마커 생성

응용(로직구현) – 예제 4

❖ Firestore 데이터 실시간 읽고 쓰기

- 안드로이드에서 일정시간 간격으로 숫자를 랜덤으로 발생시킴
- 발생한 데이터를 파이어베이스 DB에 저장함
- 이후, 안드로이드로 읽어와서 화면에 출력함

1. 프로젝트 설정
2. 레이아웃 생성
3. Firestore 설정 및 thread 설정
4. 저장 데이터 읽어오기
5. Thread를 이용한 랜덤 값 생성 및 저장

 player	 test@gmail.com
+ 문서 추가	+ 컬렉션 시작
test@gmail.com >	+ 필드 추가
	xloc: 10

82

응용(로직구현) – 예제 4

1. 프로젝트 설정

- 파이어베이스 사용 설정
 - 예제 1 참조 – Build.gradle(Modules), Build.gradle(Project), JSON file
- BuildFeatrues 추가

2. 레이아웃 생성

- (14 line) : 가운데 정렬
- (16 line) : 문자 크기 32sp

build.gradle(Module:~)

```
10      buildFeatures{
11          viewBinding true
12          dataBinding true
13      }
```

activity_main.xml

```
9      <TextView
10          android:id="@+id/textView"
11          android:layout_width="0dp"
12          android:layout_height="wrap_content"
13          android:layout_marginTop="316dp"
14          가운데 정렬 추가
15          android:text="TextView"
16          문자 크기 설정
17          app:layout_constraintEnd_toEndOf="parent"
18          app:layout_constraintHorizontal_bias="0.0"
19          app:layout_constraintStart_toStartOf="parent"
20          app:layout_constraintTop_toTopOf="parent" />
```

응용(로직구현) – 예제 4

3. Firestore 설정 및 thread 설정

- (13 line) : ActivityMainBinding 클래스 변수 선언
- (14 line) : Firestore 설정
- (15 line) : thread state
 - 0 : 시작 전, 1 : 실행 중, 2 : 실행 완료
- (20-21 line) : LayoutInflater를 통해 xml을 View 객체로 변환

MainActivity.kt

```
12 class MainActivity : AppCompatActivity() {  
13     private lateinit var binding: ActivityMainBinding  
14     val db = Firebase.firestore  
15     var oSysMainLoop = 0  
16     override fun onCreate(savedInstanceState: Bundle?) {  
17         super.onCreate(savedInstanceState)  
18         setContentView(R.layout.activity_main)  
19  
20         binding = ActivityMainBinding.inflate(layoutInflater)  
21         setContentView(binding.root)
```

응용(로직구현) – 예제 4

4. 저장 데이터 읽어오기

- (23 line) : 특정 컬렉션의 해당 문서 가져오기
- (25-29 line) : 실시간으로 바뀐 데이터 업데이트

MainActivity.kt

```
23      val adocRef = db.collection( collectionPath: "player").document( documentPath: "test@gmail.com")
24
25      adocRef.addSnapshotListener(EventListener<DocumentSnapshot> { snapshot, e ->
26          if (snapshot != null && snapshot.exists()) {
27              binding.textView.text = snapshot.data!!["xloc"].toString()
28          }
29      })
```

응용(로직구현) – 예제 4

5. Thread를 이용한 랜덤 값 생성 및 저장

- (33 line) : 1.5초 간격으로 호출, 처음 시작 시 딜레이 1초
- (38 line) : 0부터 100사이 랜덤 숫자 생성
- (39-44 line) : axloc 값 변경
- (40-43 line) : 성공, 실패 callback 리스너 기능 구현
 - Toast 를 사용해서 성공 유무를 제공

MainActivity.kt

```
31     if (oSysMainLoop == 0) {
32         oSysMainLoop = 1
33         timer(period = 1500, initialDelay = 1000)
34         { this: TimerTask
35             if (oSysMainLoop != 1) {
36                 cancel()
37             }
38             val axloc = hashMapOf("xloc" to (0..100).random())
39             db.collection( collectionPath: "player").document( documentPath: "test@gmail.com").set(axloc)
40
41
42
43         }
44     }
45 }
46 }
```

성공, 실패 callback 리스너 기능 구현