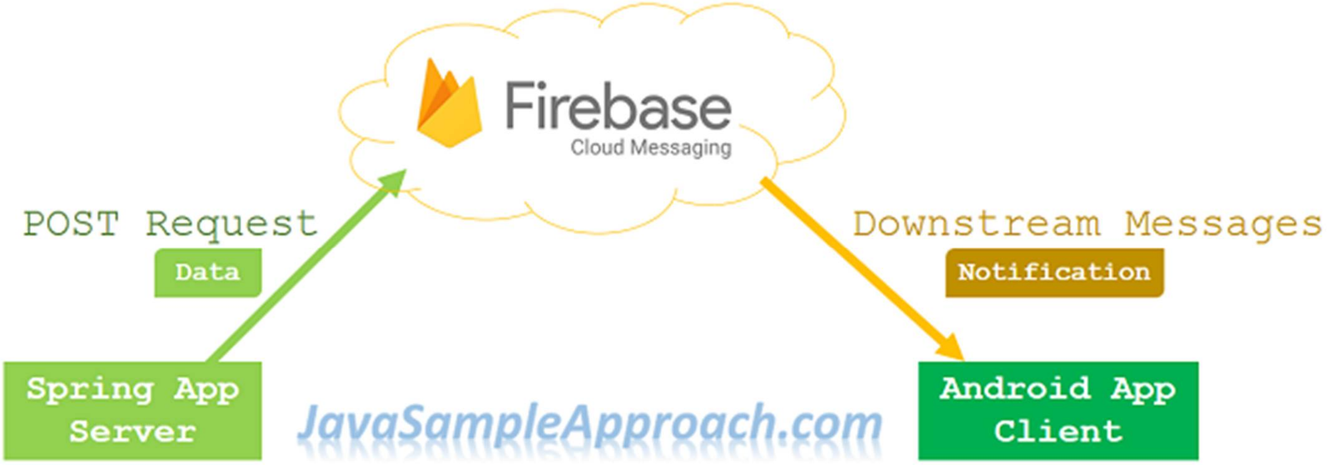
FCM 사용법

목차

1. Architecture

Firebase is a mobile and web application development platform developed by Google.

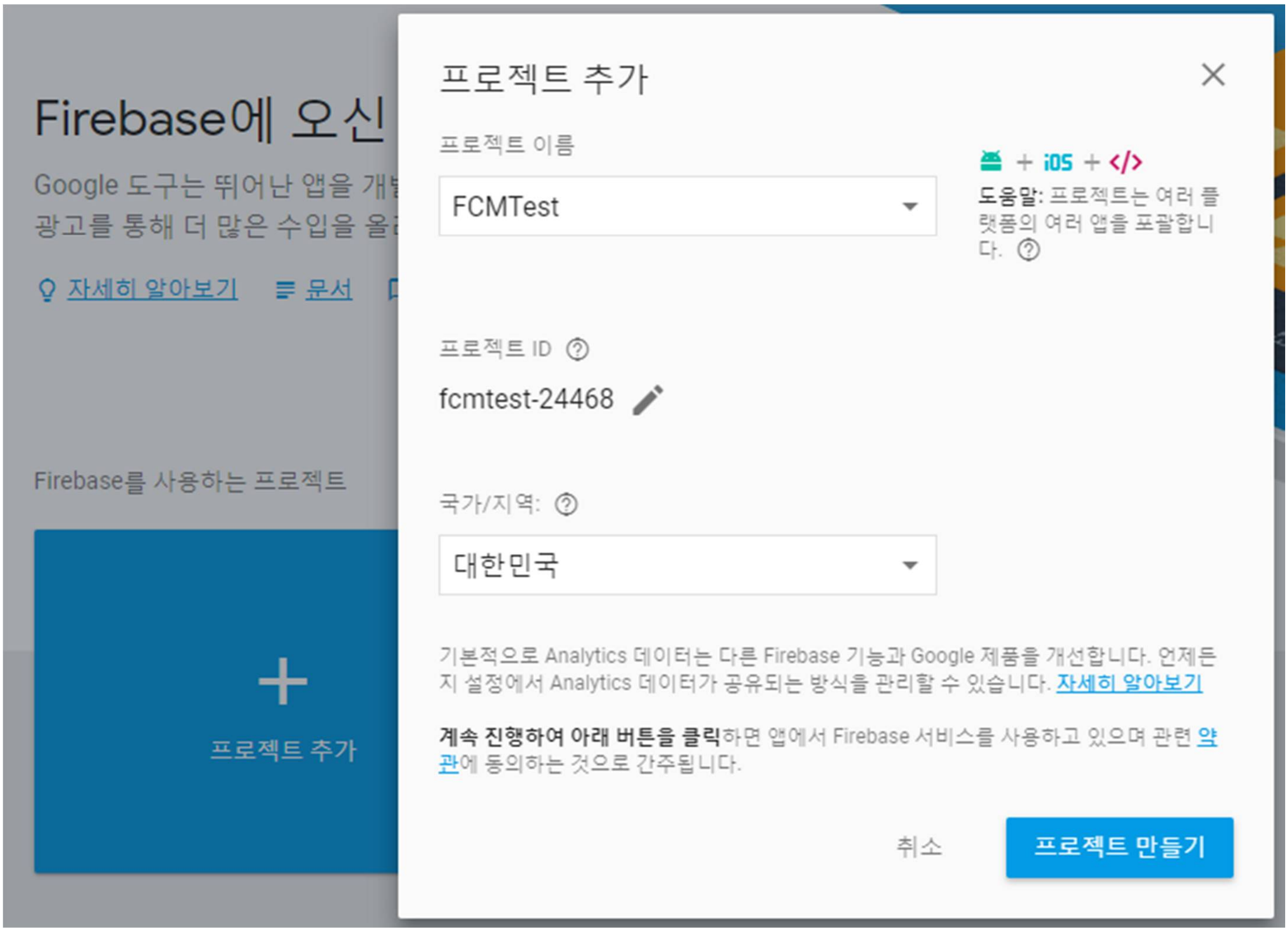


2. Firebase 프로젝트 만들기

2.1 Firebase Project 추가

Firebase 콘솔(https://console.firebase.google.com) 에 로그인하고 Firebase 프로젝트를 만든다.

프로젝트 사이트: https://console.firebase.google.com
콘솔 관리자 사이트: https://console.cloud.google.com



Firebase에 오신
Google 도구는 뛰어난 앱을 개
광고를 통해 더 많은 수입을 올
자세히 알아보기
문서
Firebase를 사용하는 프로젝트
+
프로젝트 추가
+ iOS + </>
취소
프로젝트 만들기

프로젝트 추가

프로젝트 이름: FCMTest

도움말: 프로젝트는 여러 플랫폼의 여러 앱을 포괄합니다.

프로젝트 ID: fcmtest-24468

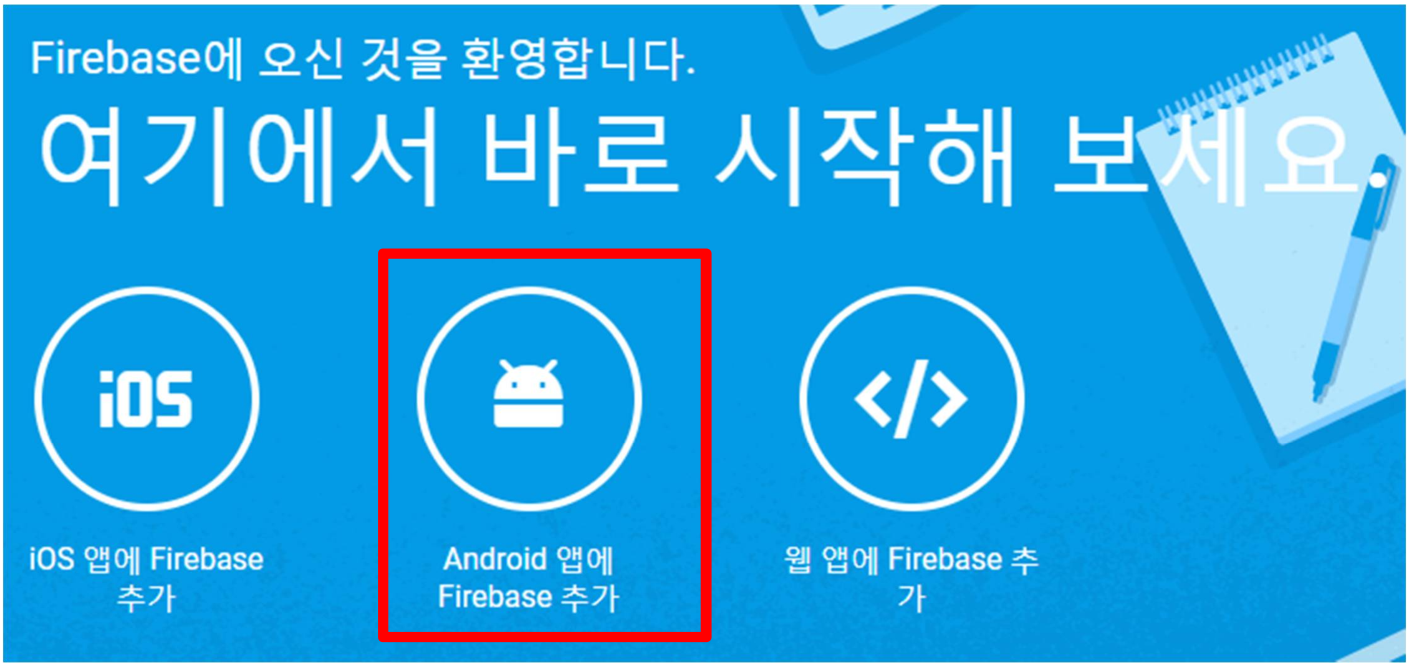
국가/지역: 대한민국

기본적으로 Analytics 데이터는 다른 Firebase 기능과 Google 제품을 개선합니다. 언제든지 설정에서 Analytics 데이터가 공유되는 방식을 관리할 수 있습니다. 자세히 알아보기

계속 진행하여 아래 버튼을 클릭하면 앱에서 Firebase 서비스를 사용하고 있으며 관련 약관에 동의하는 것으로 간주됩니다.

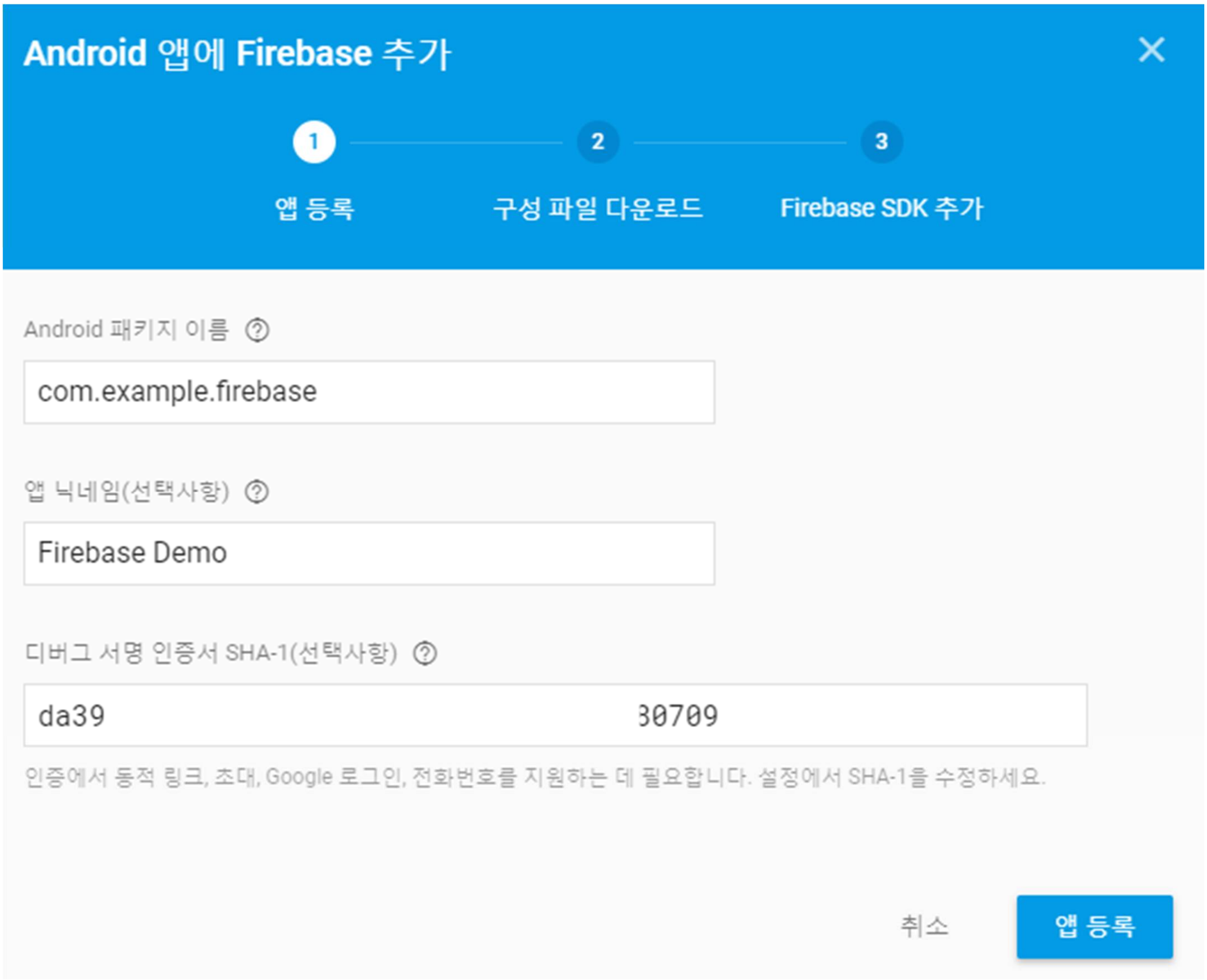
2.2 App 추가

Fill Android package name, App nickname, and SHA-1 (got from step 1):



Firebase에 오신 것을 환영합니다.

여기에서 바로 시작해 보세요.



Android 앱에 Firebase 추가

Android 패키지 이름 ⓘ

com.example.firebase

앱 닉네임(선택사항) ⓘ

Firebase Demo

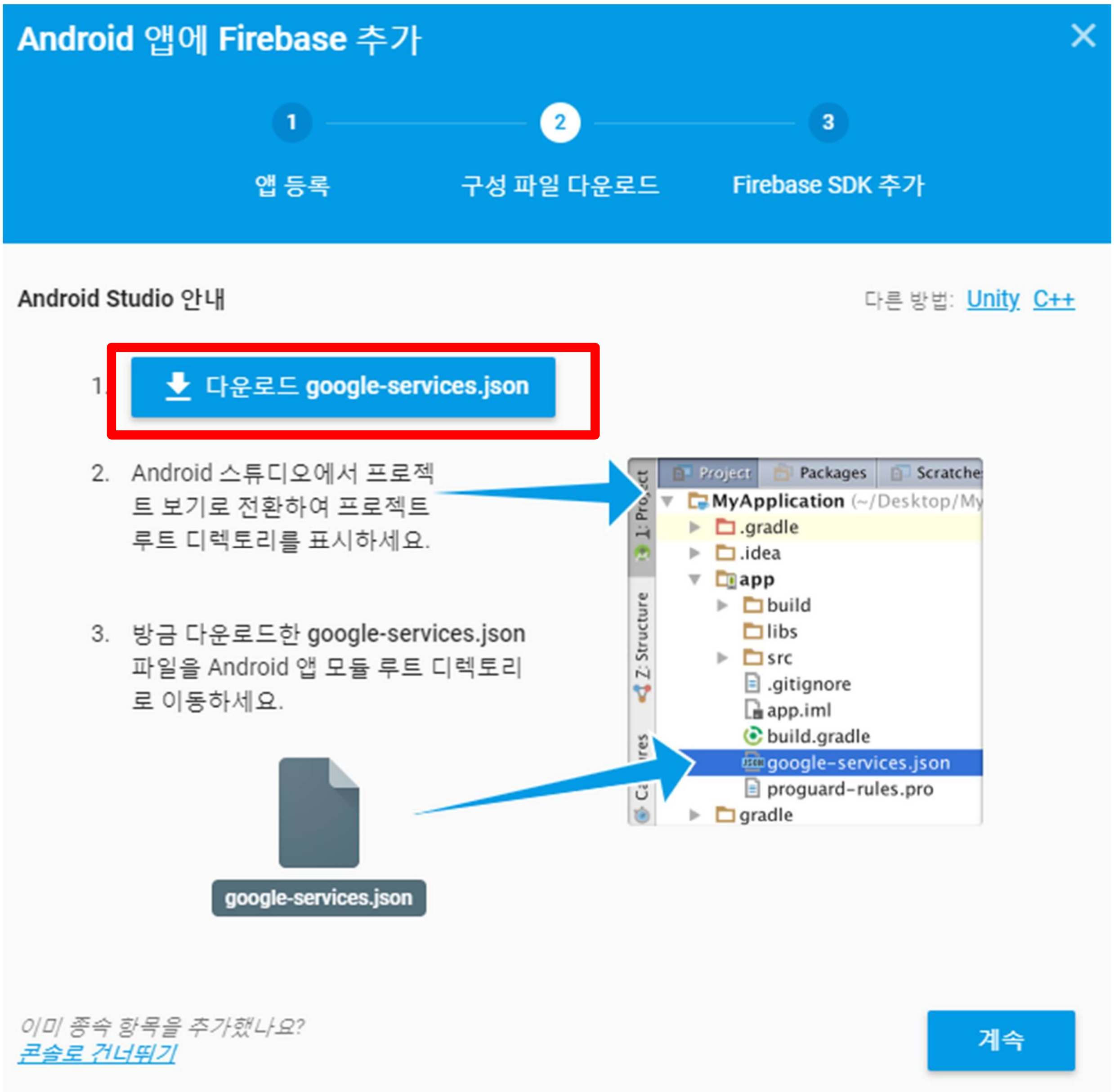
디버그 서명 인증서 SHA-1(선택사항) ⓘ

da39 30709

인증에서 동적 링크, 초대, Google 로그인, 전화번호를 지원하는 데 필요합니다. 설정에서 SHA-1을 수정하세요.

2.3 google-services.json 다운로드

- Click on Download google-services.json to download the config JSON file:



Android 앱에 Firebase 추가

1 앱 등록 — 2 구성 파일 다운로드 — 3 Firebase SDK 추가

Android Studio 안내

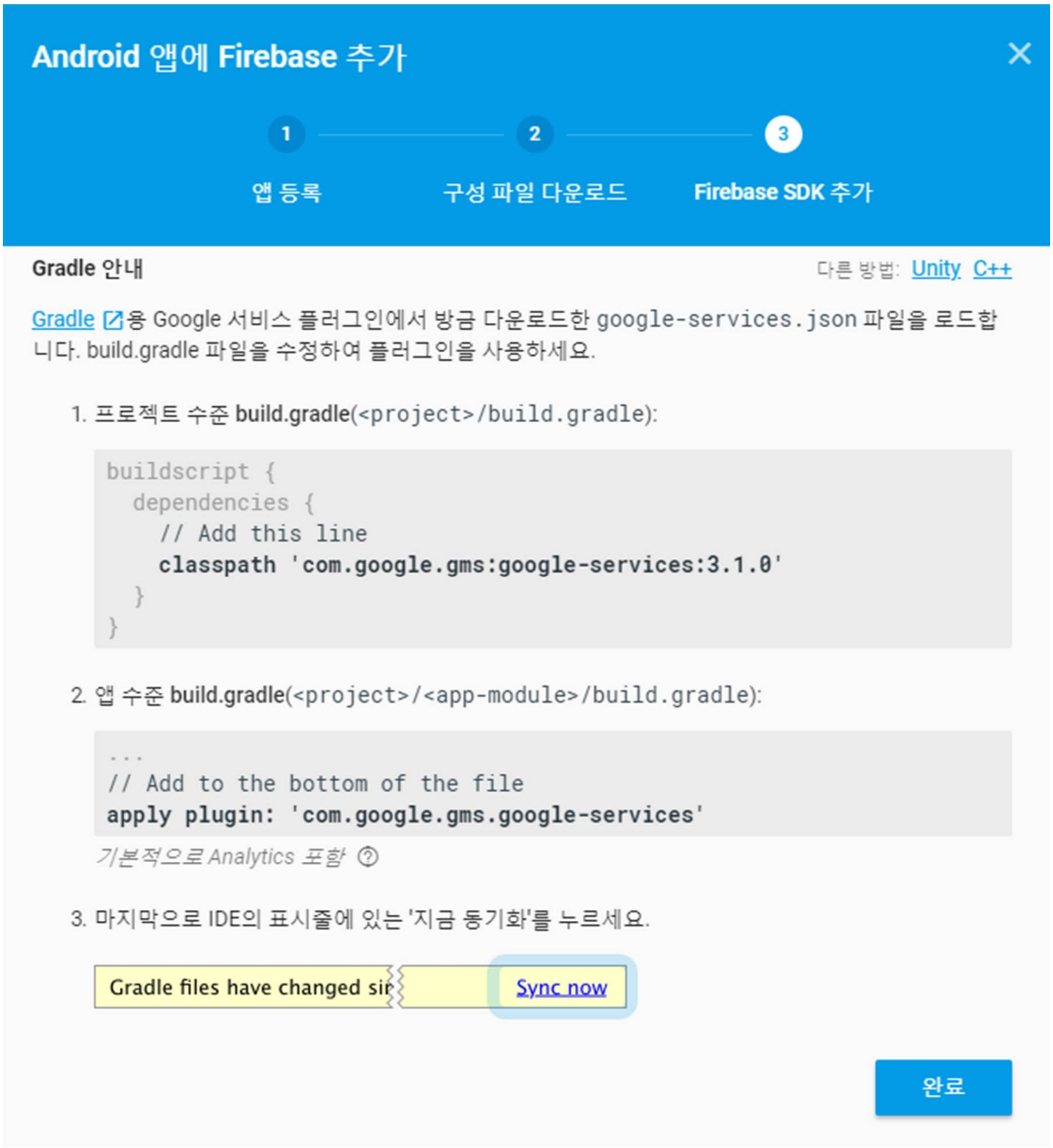
다른 방법: Unity C++

1. 다운로드 google-services.json
2. Android 스튜디오에서 프로젝트 보기로 전환하여 프로젝트 루트 디렉토리를 표시하세요.
3. 방금 다운로드한 google-services.json 파일을 Android 앱 모듈 루트 디렉토리로 이동하세요.

이미 종속 항목을 추가했나요?
콘솔로 건너뛰기

계속

2.4 Add Firebase SDK



Gradle 용 Google 서비스 플러그인에서 방금 다운로드한 google-services.json 파일을 로드합니다. build.gradle 파일을 수정하여 플러그인을 사용하세요.

1. 프로젝트 수준 **build.gradle**(`<project>/build.gradle`):

```
buildscript {
  dependencies {
    // Add this line
    classpath 'com.google.gms:google-services:3.1.0'
  }
}
```

2. 앱 수준 **build.gradle**(`<project>/<app-module>/build.gradle`):

```
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

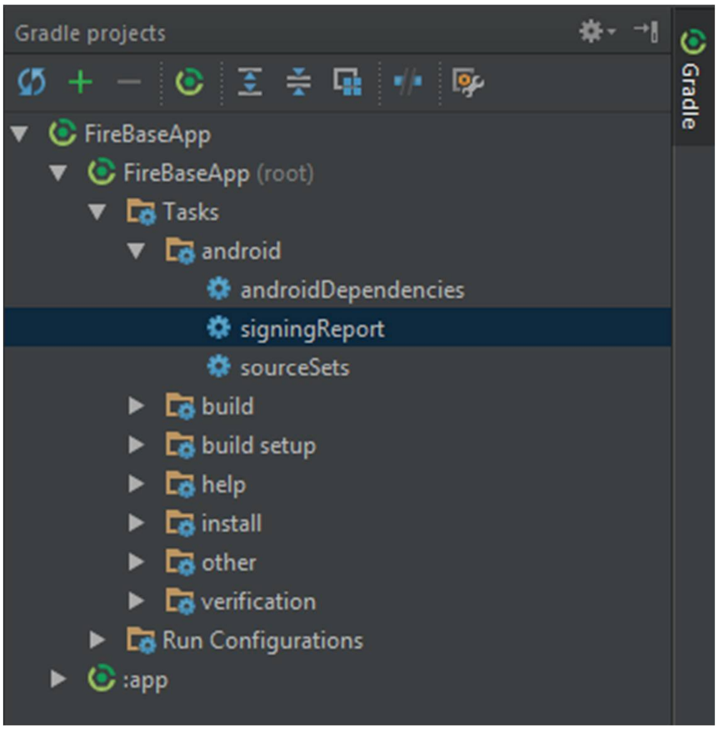
기본적으로 Analytics 포함 ⓘ

3. 마지막으로 IDE의 표시줄에 있는 '지금 동기화'를 누르세요.

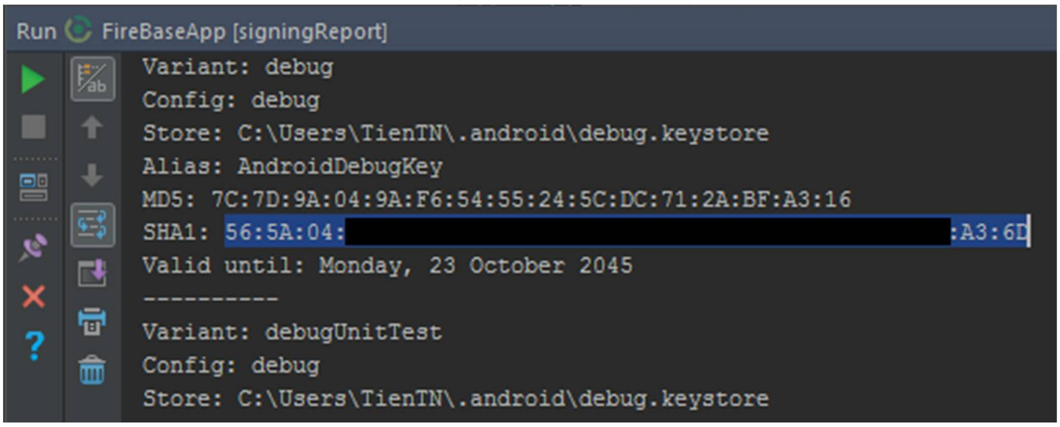
3. 안드로이드 프로젝트 만들기

3.1 SHA1 Key 만들기

- Open your Android Project, in Gradle projects window, double-click on signingReport:

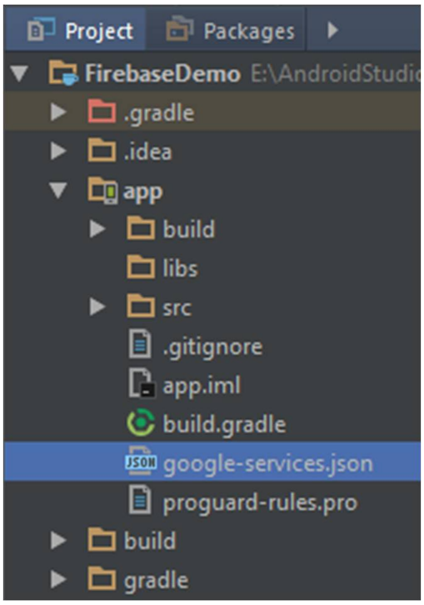


- Look at Run window, you can find SHA-1 Key:



```
Variant: debug
Config: debug
Store: C:\Users\TienTN\.android\debug.keystore
Alias: AndroidDebugKey
MD5: 7C:7D:9A:04:9A:F6:54:55:24:5C:DC:71:2A:BF:A3:16
SHA1: 56:5A:04:[redacted]:A3:6D
Valid until: Monday, 23 October 2045
----------
Variant: debugUnitTest
Config: debug
Store: C:\Users\TienTN\.android\debug.keystore
```

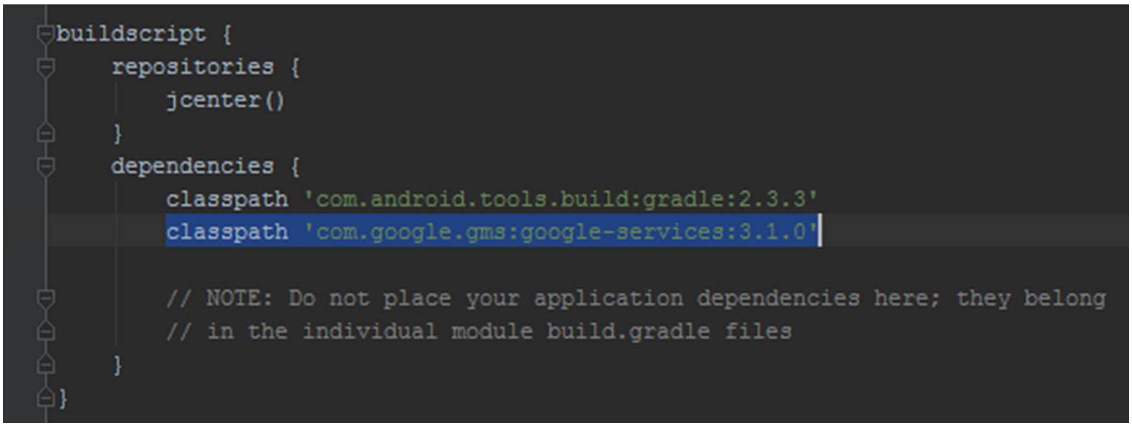
3.2 add google-services.json into your Android App



```
FirebaseDemo E:\AndroidStudio
  .gradle
  .idea
  app
    build
    libs
    src
    .gitignore
    app.iml
    build.gradle
    google-services.json
    proguard-rules.pro
  build
  gradle
```

3.2.1 프로젝트 build.gradle(<project>/build.gradle) 수정

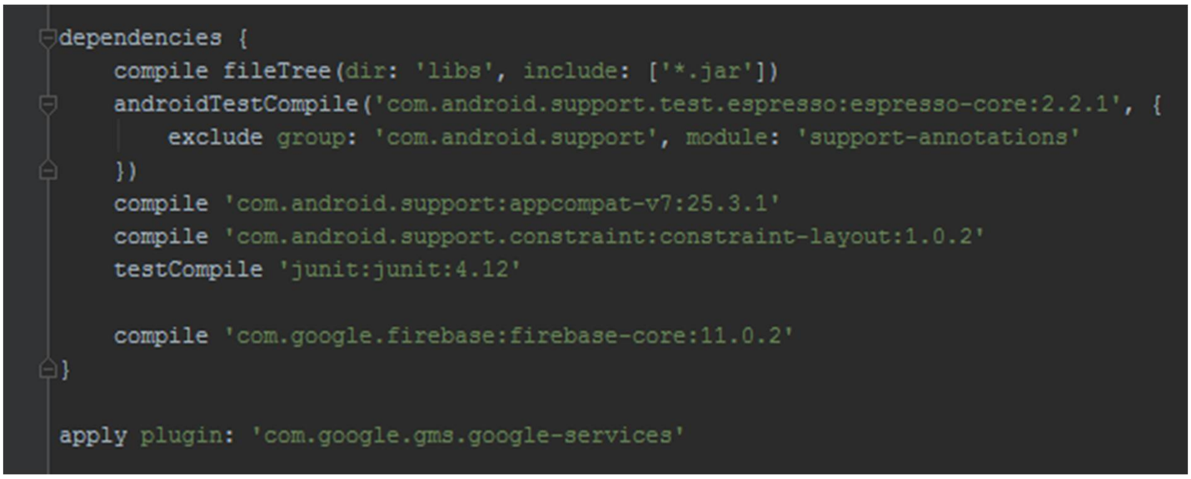
```
buildscript {
  dependencies {
    // Add this line
    classpath 'com.google.gms:google-services:3.2.0'
  }
}
```



```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.3'
        classpath 'com.google.gms:google-services:3.1.0'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}
```

3.2.2 앱 build.gradle 수정(<project>/<app-module>/build.gradle)

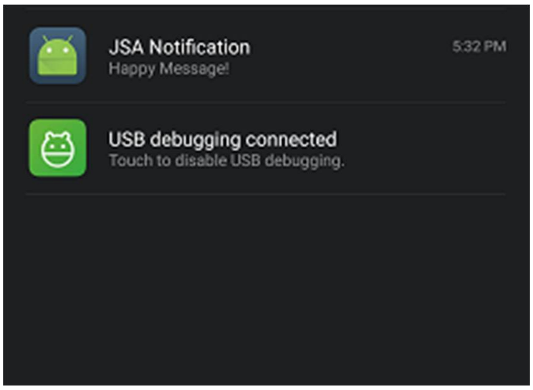
```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

// 생략 ...

dependencies {

    // Firebase 라이브러리 추가
    compile 'com.google.firebase:firebase-core:11.4.2'
    compile 'com.google.firebase:firebase-messaging:11.4.2'
}
```



```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.1', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'

    compile 'com.google.firebase:firebase-core:11.0.2'
}

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

3.3 Re-sync Gradle files:

Tool -> Android -> Sync Project with Gradle Files.

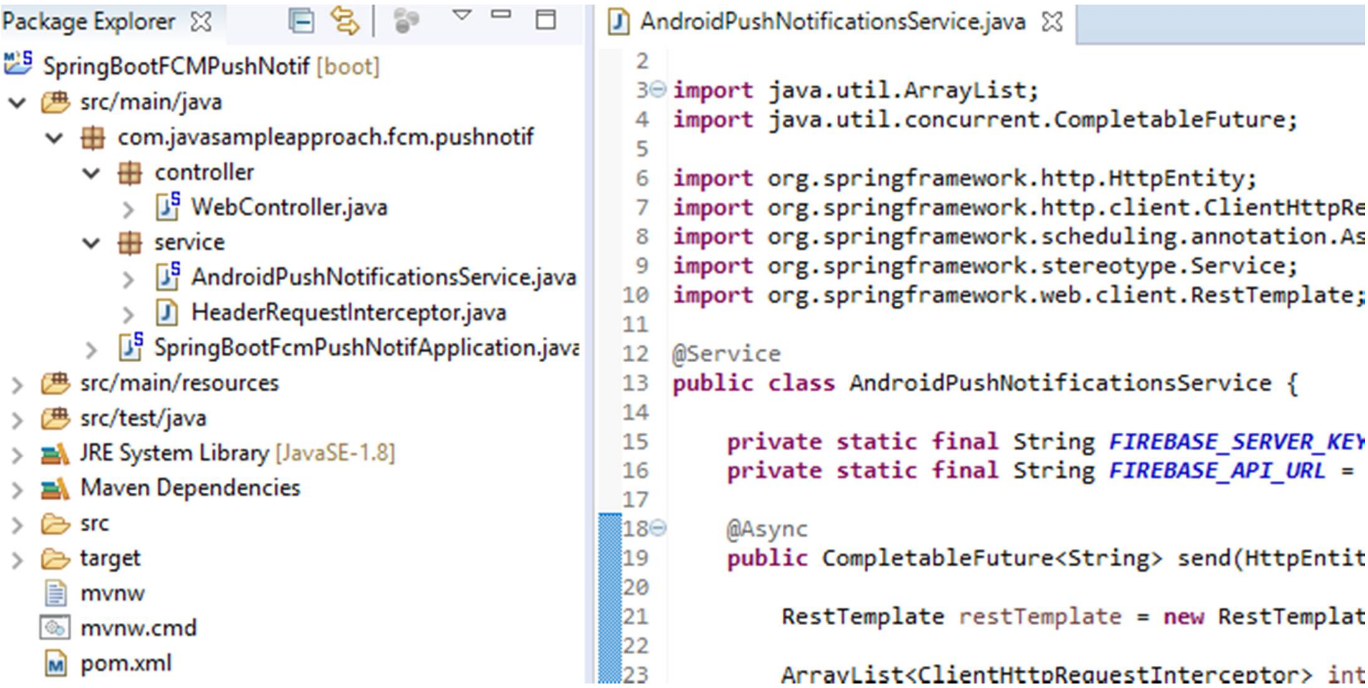


JSA Notification 5:32 PM
Happy Message!

USB debugging connected
Touch to disable USB debugging.

4. 스프링 프로젝트 만들기

4.1 스프링 프로젝트 Structure



```
SpringBootFCMPushNotif [boot]
  src/main/java
    com.javasampleapproach.fcm.pushnotif
      controller
        WebController.java
      service
        AndroidPushNotificationsService.java
        HeaderRequestInterceptor.java
      SpringBootFcmPushNotifApplication.java
  src/main/resources
  src/test/java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  mvnw
  mvnw.cmd
  pom.xml
```

```java

import java.util.ArrayList;
import java.util.concurrent.CompletableFuture;

import org.springframework.http.HttpEntity;
import org.springframework.http.client.ClientHttpRe
import org.springframework.scheduling.annotation.As
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

@Service
public class AndroidPushNotificationsService {

    private static final String FIREBASE_SERVER_KEY
    private static final String FIREBASE_API_URL =

    @Async
    public CompletableFuture<String> send(HttpEntit

        RestTemplate restTemplate = new RestTemplat

        ArrayList<ClientHttpRequestInterceptor> int
```

- AndroidPushNotificationsService is the service that provides send() method to FCM.
- WebController is a REST Controller to push Notification.

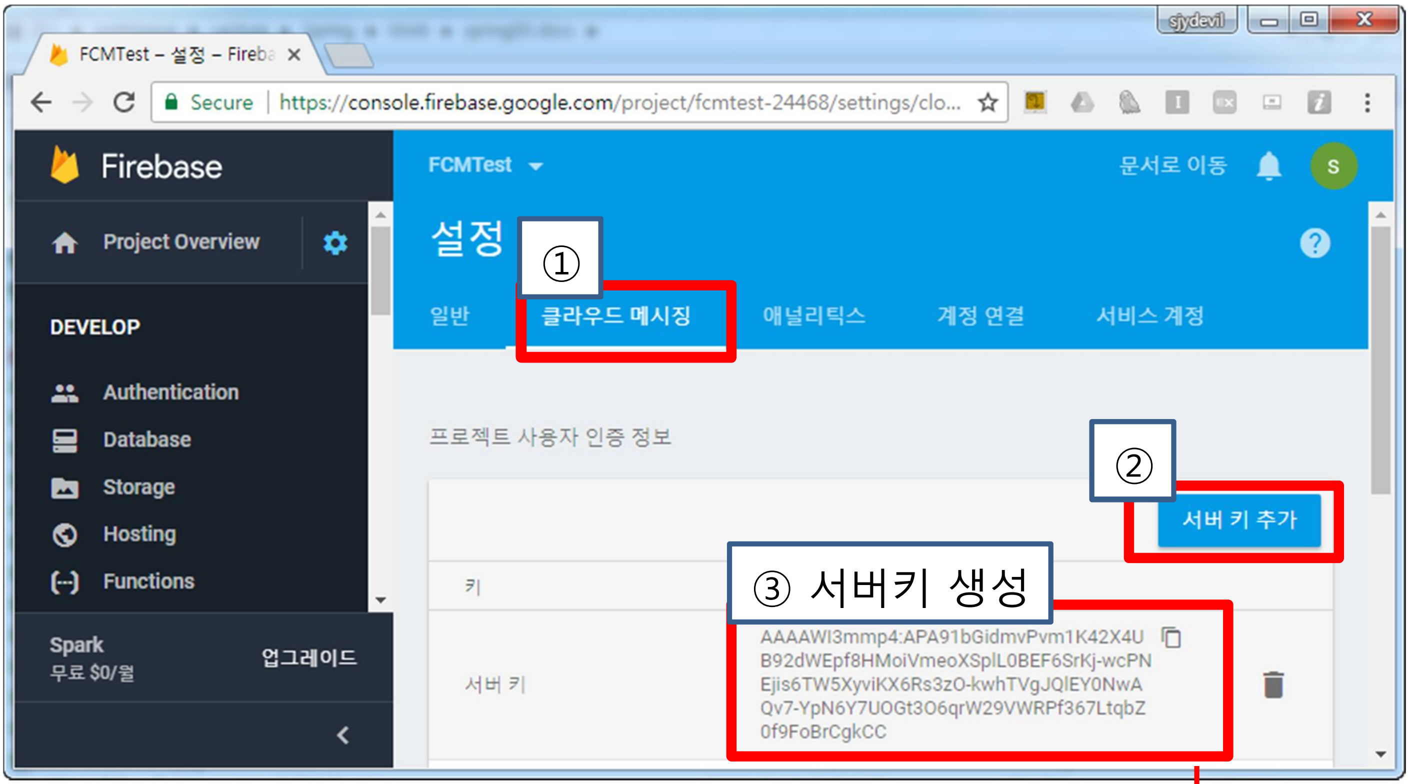
4.2 스프링 build.gradle 에 라이브러리 추가

```
compile "org.json:json:20160810"
compile "org.springframework.boot:spring-boot-starter-web:1.5.8.RELEASE"
testCompile(group: 'org.springframework.boot', name: 'spring-boot-starter-test' ) {
    exclude(module: 'commons-logging')
}
```

4.3 Create Push Notification Service

```
package com.example.fcm;

import java.io.IOException;

import org.springframework.http.HttpRequest;
import org.springframework.http.client.*;
import org.springframework.http.client.support.HttpRequestWrapper;

public class HeaderRequestInterceptor implements ClientHttpRequestInterceptor {

      private final String headerName;
      private final String headerValue;

      public HeaderRequestInterceptor(String headerName, String headerValue) {
            this.headerName = headerName;
            this.headerValue = headerValue;
      }

      @Override
      public ClientHttpResponse intercept(HttpRequest request, byte[] body,
ClientHttpRequestExecution execution)
                  throws IOException {
            HttpRequest wrapper = new HttpRequestWrapper(request);
            wrapper.getHeaders().set(headerName, headerValue);
            return execution.execute(wrapper, body);
      }
}
```

4.4 AndroidPushNotificationsService



AAAAWI3mmp4:APA91bGidmvPvm1K42X4U
B92dWEpf8HMoiVmeoXSplL0BEF6SrKj-wcPN
Ejis6TW5XyviKX6Rs3zO-kwhTVgJQlEY0NwA
Qv7-YpN6Y7UOGt3O6qrW29VWRPf367LtqbZ
0f9FoBrCgkCC

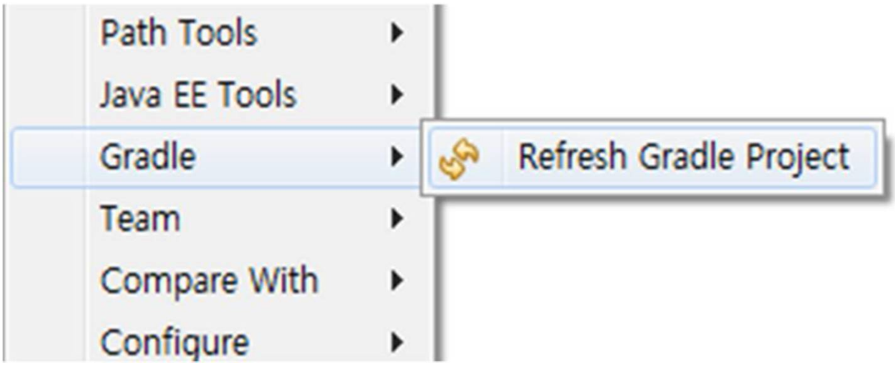
```
package com.example.fcm;

import java.util.ArrayList;
import java.util.concurrent.CompletableFuture;

import org.springframework.http.HttpEntity;
import org.springframework.http.client.ClientHttpRequestInterceptor;
import org.springframework.scheduling.annotation.Async;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

@Service
public class AndroidPushNotificationsService {

      private static final String FIREBASE_SERVER_KEY = "Your Server Key here!";
      private static final String FIREBASE_API_URL =
"https://fcm.googleapis.com/fcm/send";

      @Async
      public CompletableFuture<String> send(HttpEntity<String> entity) {
            RestTemplate restTemplate = new RestTemplate();

            /**
            https://fcm.googleapis.com/fcm/send
            Content-Type:application/json
            Authorization:key=FIREBASE_SERVER_KEY*/

            ArrayList<ClientHttpRequestInterceptor> interceptors = new ArrayList<>();
            interceptors.add(new HeaderRequestInterceptor("Authorization", "key=" +
FIREBASE_SERVER_KEY));
            interceptors.add(new HeaderRequestInterceptor("Content-Type",
"application/json"));
            restTemplate.setInterceptors(interceptors);

            String firebaseResponse = restTemplate.postForObject(FIREBASE_API_URL, entity,
String.class);

            return CompletableFuture.completedFuture(firebaseResponse);
      }
}
```

4.5 Create Controller

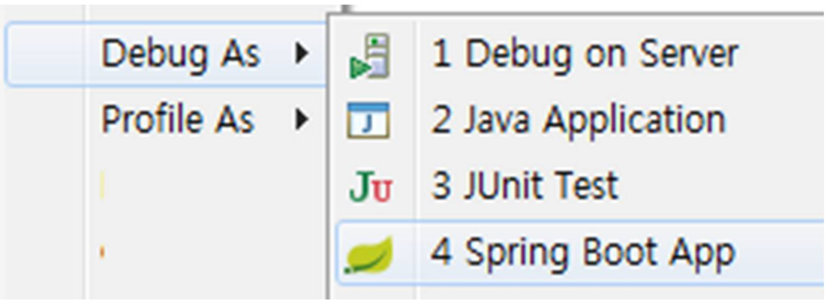
```
package com.example.fcm;

import java.util.concurrent.CompletableFuture;
import java.util.concurrent.ExecutionException;

import org.json.JSONException;
import org.json.JSONObject;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

import com.example.fcm.service.AndroidPushNotificationsService;

@RestController
public class WebController {
    private final String TOPIC = "JavaSampleApproach";

    @Autowired
    AndroidPushNotificationsService androidPushNotificationsService;

    @RequestMapping(value = "/fcm/send", method = RequestMethod.GET, produces =
"application/json")
    public ResponseEntity<String> send() throws JSONException {

        JSONObject body = new JSONObject();
        body.put("to", "/topics/" + TOPIC);
        body.put("priority", "high");

        JSONObject notification = new JSONObject();
        notification.put("title", "JSA Notification");
        notification.put("body", "Happy Message!");

        JSONObject data = new JSONObject();
        data.put("Key-1", "JSA Data 1");
        data.put("Key-2", "JSA Data 2");

        body.put("notification", notification);
        body.put("data", data);

        HttpEntity<String> request = new HttpEntity<>(body.toString());

        CompletableFuture<String> pushNotification =
androidPushNotificationsService.send(request);
        CompletableFuture.allOf(pushNotification).join();

        try {
            String firebaseResponse = pushNotification.get();

            return new ResponseEntity<>(firebaseResponse, HttpStatus.OK);
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (ExecutionException e) {
            e.printStackTrace();
        }

        return new ResponseEntity<>("Push Notification ERROR!",
HttpStatus.BAD_REQUEST);
    }
}
```

4.6 Run & Check Result

- - Config gradle clean build



- - Run project with mode Spring Boot App.



- - Open Browser, enter URL:

✓ Send Data

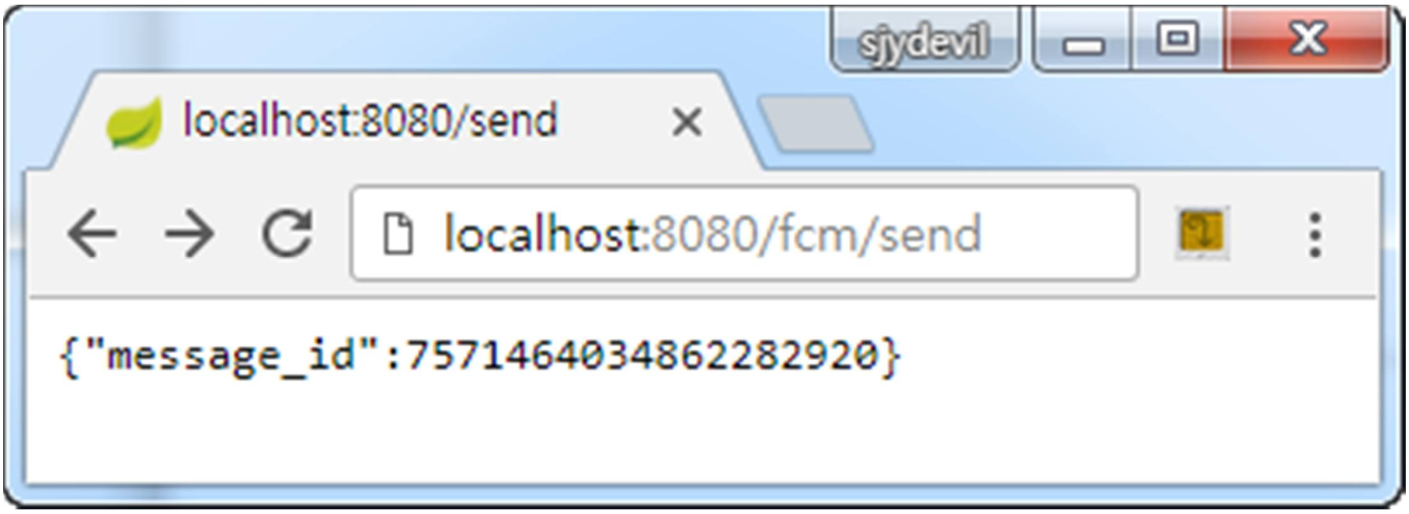
```
https://fcm.googleapis.com/fcm/send
Content-Type:application/json
Authorization:key=SERVER_KEY
{
   "to": "/topics/JavaSampleApproach",
   "notification": {
      "title": "TITLE",
      "body": "BODY"
   },
   "data": {
      "Key-1": "VALUE 1",
      "Key-2": "VALUE 2"
   },
}
```

Receive Data : Success

```
{
  "message_id": "123456"
}
```

Receive Data : Failure

```
{
  "error": "TopicsMessageRateExceeded"
}
```



```
{"message_id":7571464034862282920}
```

5. Reference

http://javasampleapproach.com/android/how-to-integrate-firebase-android-app-android-studio#3_Add_Firebase_to_Android_App

http://javasampleapproach.com/spring-framework/spring-boot/firebase-cloud-messaging-server-spring-to-push-notification-example-spring-boot

http://javasampleapproach.com/android/firebase-cloud-messaging-how-to-subscribe-topic-receive-messages-android

https://m.blog.naver.com/momsoccer/220749543950

https://m.blog.naver.com/momsoccer/220753681866