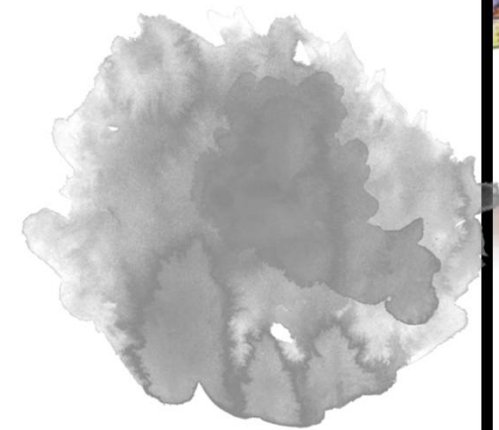


45.Push Messaging

- SMS / MMS
- C2DM
- GCM(Google Cloud Messaging)
- FCM(Firebase Cloud Messaging)





이번 장에서는 무엇을 다룰까요?



푸시로 공지사항을 단말로 보내고 싶어요.

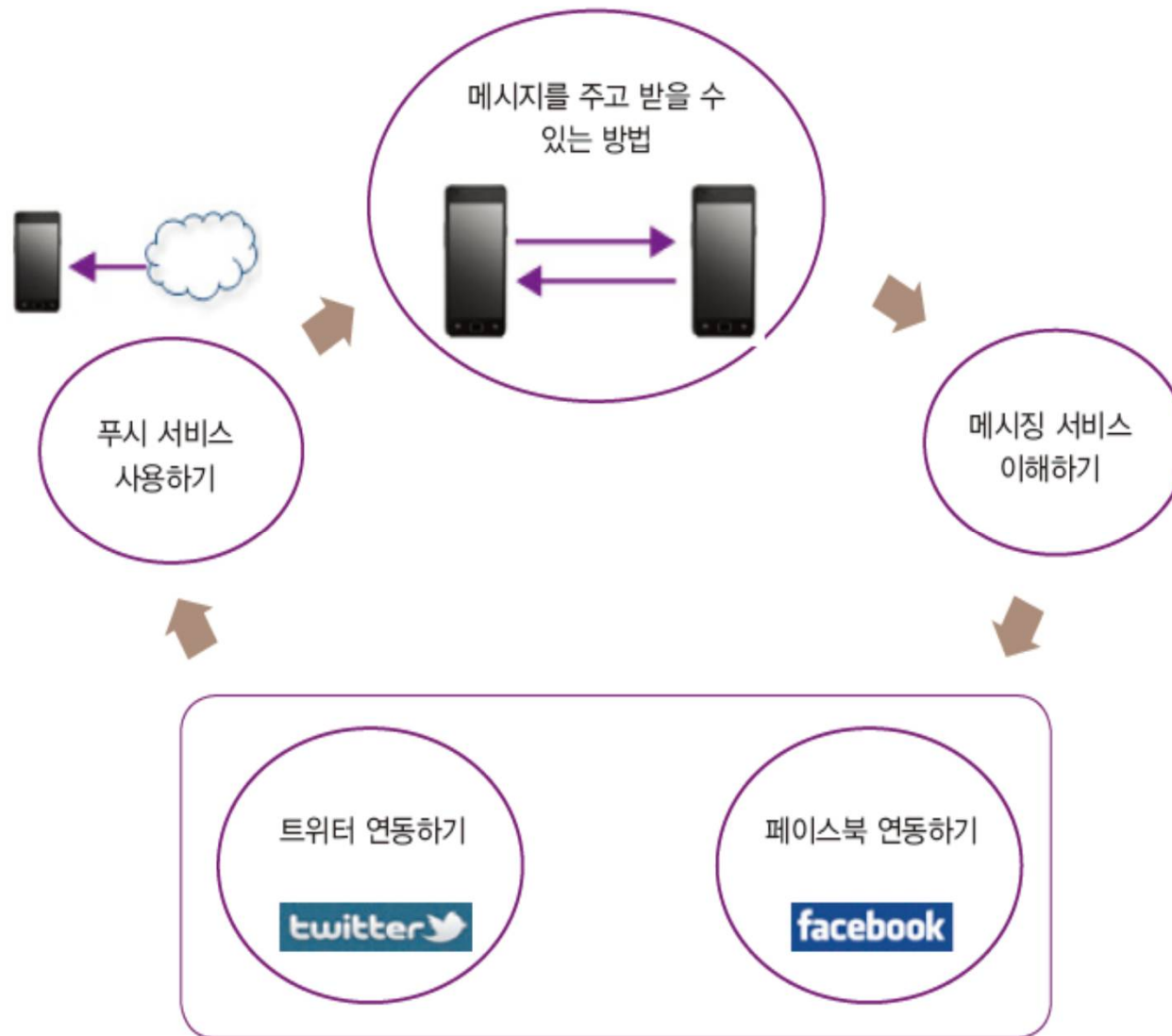


- 메시지를 주고받을 수 있는 방법들에 대해 알아보을까요?
- 단말로 메시지를 보내는 푸쉬 서비스에 대해 알아보을까요?
- 페이스북을 연동하는 방법에 대해 알아보을까요?





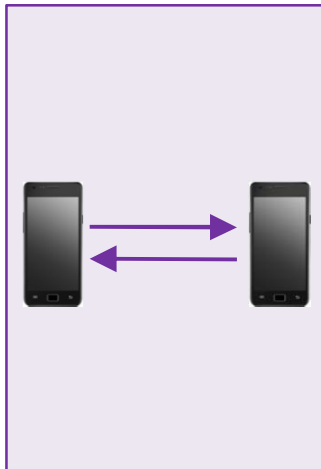
이번 장에서는 무엇을 다룰까요?





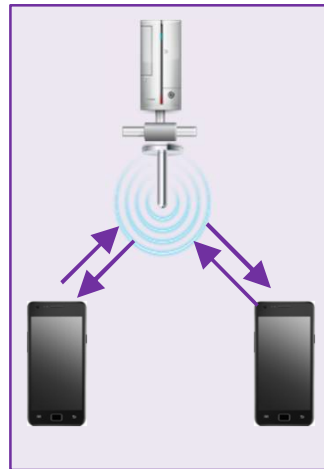
메시징 서비스의 구분

1. 단말 간 메시지
송수신



SMS / MMS

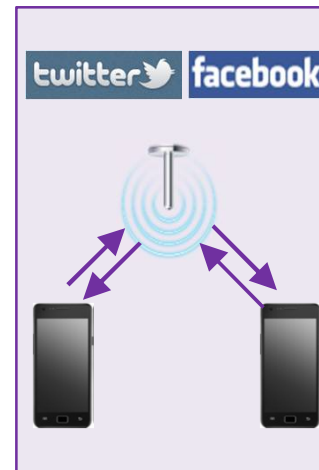
2. 서버를 통한
단말 간 메시지 송수신



XMPP

SIP

3. SNS를 통한
단말 간 메시지 공유



Twitter

Facebook

4. 푸시 서비스를 통한
메시지 전송



FCM

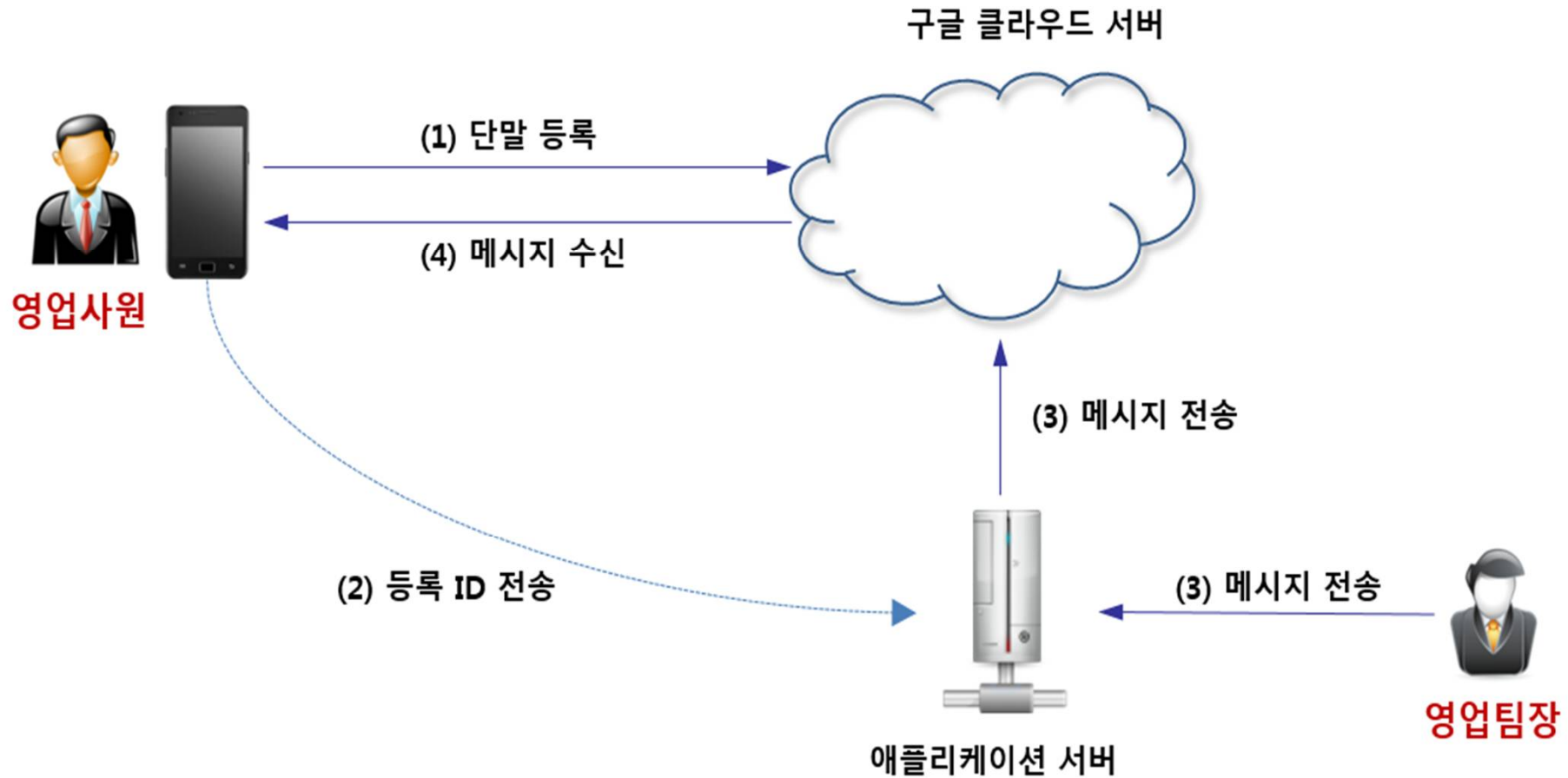


푸시 서비스 사용하기

- SMS를 이용한 알림
 - 간단하지만 비용 발생
- MMS를 이용한 알림
 - 간단하지만 비용 발생
- 폴링 알림
 - 연결을 만들어 놓고 폴링하는 과정이 필요하며, 백그라운드 서비스를 이용해야 하므로 간단하지만은 않고 단말 리소스를 많이 사용함
- FCM 알림
 - 연결을 만들어 놓고 폴링을 일부 사용하지만 구글의 클라우드 서버를 이용해 최적화한 서비스로 단말의 Play스토어에서 제공하는 기능이므로 효율적인 푸시 메시지 전송 가능



푸시 서비스 처리 과정





푸시 서비스 처리에 필요한 주요 정보

- (1) 애플리케이션 서버에 저장된 단말의 등록 ID
- (2) 애플리케이션 서버에서 클라우드 서버로 접속하기 위한 인증 정보



FCM 프로젝트 만들기(1)

1. FCM 콘솔에서 프로젝트 등록
2. 프로젝트를 만든 후 [Android 앱에 Firebase 추가] 아이콘 클릭
3. google-services.json 다운로드
4. 프로젝트(Project:SamplePush) build.gradle 파일 수정

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

5. 앱 모듈(Module:app) build.gradle 파일 수정

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




FCM 프로젝트 만들기(2)

- FCM 개발자 콘솔 페이지 접속하여 프로젝트 등록

Firebase 콘솔(Console) 페이지 ▶ <https://console.firebase.google.com>

Firestore에 오신
Google 도구는 뛰어난 앱을 개발
광고를 통해 더 많은 수입을 올
[자세히 알아보기](#) [문서](#)

Firestore를 사용하는 프로젝트

+
프로젝트 추가

프로젝트 추가

프로젝트 이름

FCMTest

프로젝트 ID ②

fcctest-24468

국가/지역: ②

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 [약관](#)에 동의하는 것으로 간주됩니다.

취소 프로젝트 만들기



FCM 프로젝트 만들기(3)

- 프로젝트를 만든 후 [Android 앱에 Firebase 추가] 아이콘 클릭

Firebase에 오신 것을
여기에서

iOS

iOS 앱에 Firebase
추가

Android 앱에 Firebase 추가

1

2

3

앱 등록

구성 파일 다운로드

Firebase SDK 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.

취소

앱 등록



FCM 프로젝트 만들기(4)

- google-services.json 파일 다운로드 후 프로젝트 app 폴더에 저장

Android 앱에 Firebase 추가

1

2

3

앱 등록 구성 파일 다운로드 Firebase SDK 추가

Android Studio 안내

다른 방법: [Unity](#) [C++](#)

- [다운로드 google-services.json](#)
- Android 스튜디오에서 프로젝트 보기로 전환하여 프로젝트 루트 디렉토리를 표시하세요.
- 방금 다운로드한 google-services.json 파일을 Android 앱 모듈 루트 디렉토리로 이동하세요.

google-services.json

이미 종속 항목을 추가했나요?
[콘솔로 건너뛰기](#)

계속

다른 이름으로 저장

구성 새 폴더

이름	수정한 날짜
libs	2016-11-
src	2016-11-
build	2016-11-

파일 이름(N): google-services.json
파일 형식(T): JSON 파일(.json)

폴더 숨기기 저장(S) 취소



FCM 프로젝트 만들기(5) - 앱 설정 수정

1. 프로젝트(Project:SamplePush) build.gradle 파일 수정

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

2. 앱 모듈(Module:app) build.gradle 파일 수정


```
apply plugin: 'com.android.application'  
apply plugin: 'com.google.gms.google-services'  
  
// 중략 ...  
  
dependencies {  
  
    // Firebase 라이브러리 추가  
    compile 'com.google.firebase:firebase-core:11.4.2'  
    compile 'com.google.firebase:firebase-messaging:11.4.2'  
}
```



안드로이드 앱 만들기(1)

- SamplePush 라는 이름의 새로운 프로젝트 생성

Create New Project

 **New Project**
Android Studio

Configure your new project

Application name:

Company domain:

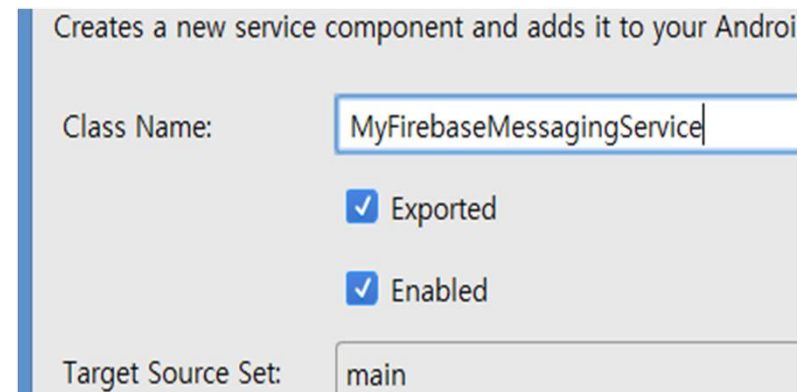
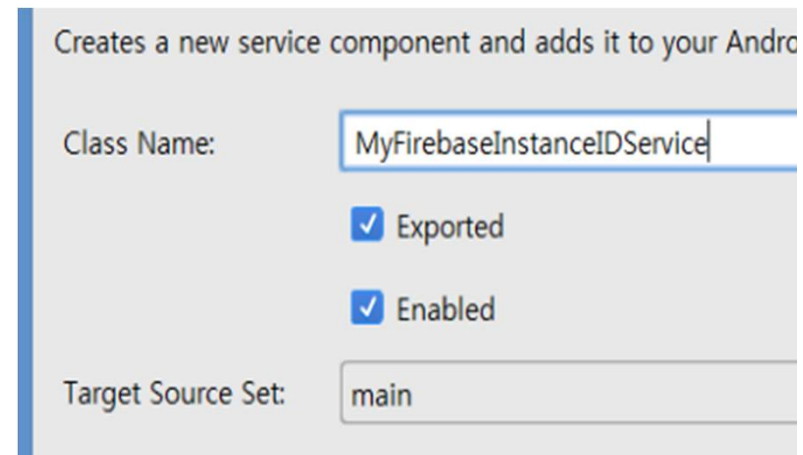
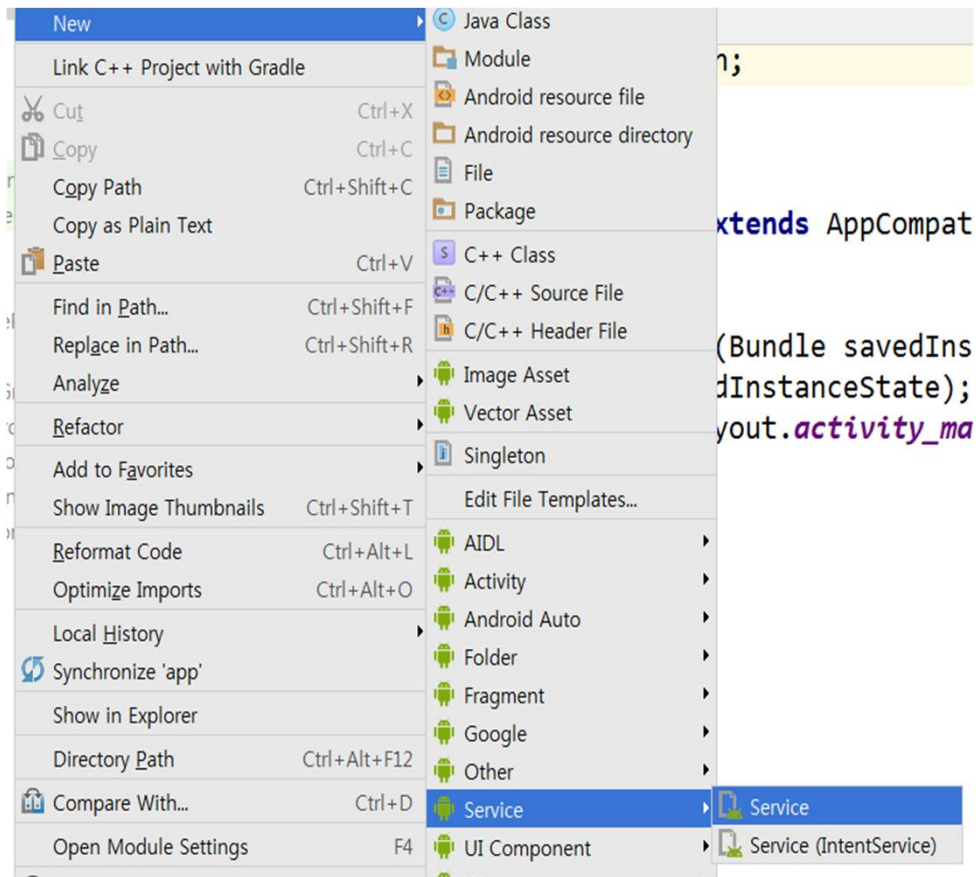
Package name: [Done](#)

☐ Include C++ support



안드로이드 앱 만들기(2) - 서비스

- 앱에 두 개의 서비스를 추가해야 함
 - 등록ID를 위한 MyFirebaseInstanceIdService 서비스 추가
 - 등록ID를 위한 MyFirebaseMessagingService 서비스 추가
 - Manifest.xml에 서비스 태그 안에 인텐트 필터 정보 추가





안드로이드 앱 만들기(2) - 서비스

```
public class MyFirebaseInstanceIdService extends FirebaseInstanceIdService {  
    private static final String TAG = "MyIID";  
  
    @Override  
    public void onTokenRefresh() {  
        Log.d(TAG, "onTokenRefresh() 호출됨.");  
  
        String refreshedToken = FirebaseInstanceId.getInstance().getToken();  
        Log.d(TAG, "Refreshed Token : " + refreshedToken);  
    }  
}
```



안드로이드 앱 만들기(2) - 서비스

```
public class MyFirebaseMessagingService extends FirebaseMessagingService {  
    private static final String TAG = "MyMS";  
  
    @Override  
    public void onMessageReceived(RemoteMessage remoteMessage) {  
        Log.d(TAG, "onMessageReceived() 호출됨.");  
        String from = remoteMessage.getFrom();  
        Map<String, String> data = remoteMessage.getData();  
        String contents = data.get("contents");  
        Log.v(TAG, "from : " + from + ", contents : " + contents);  
        sendToActivity(getApplicationContext(), from, contents);  
    }  
  
    private void sendToActivity(Context context, String from, String contents) {  
        Intent intent = new Intent(context, MainActivity.class);  
        intent.putExtra("from", from);  
        intent.putExtra("contents", contents);  
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_SINGLE_TOP);  
        context.startActivity(intent);  
    }  
}
```




안드로이드 앱 만들기(2) - 서비스

- Manifest.xml의 서비스 태그 안에 인텐트 필터 정보 추가
- 인터넷 권한 추가

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

<service
    android:name=".MyFirebaseInstanceIdService"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="com.google.firebase.INSTANCE_ID_EVENT" />
    </intent-filter>
</service>

<service
    android:name=".MyFirebaseMessagingService"
    android:enabled="true"
    android:exported="true" >
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT" />
    </intent-filter>
</service>
```



안드로이드 앱 만들기(3) - 화면

- 보내는 사람과 받는 사람의 기능을 한 화면에 구성





서버 키 값 입력

- Firebase 콘솔 페이지의 설정 메뉴에서 서버 키 값 확인하여 입력

Firestore Console

https://console.firebase.google.com/project/samplepush-59726/settings/cloudmessaging

SamplePush 문서로 이동

설정

일반설정 클라우드 메시징 애널리틱스 계정 연결 서비스 계정

프로젝트 사용자 인증 정보

Firebase Cloud Messaging 토큰 발신자 ID

AAAA_b17X-4:APA91bEOAH0oYbwFx2x4aeNJPz-X-i6BB
sEaKz4Khyet7Q9xTGW6SlgJec66_TBPqi4ZxUbCtJ87Zc
FNbLweg0IzubLbsfKRtcvmaFxV-6kqcZVFJVhzvDHExfeK
4rltvfhWA83MoxOPVhTUFnEQUnt9xiC6nt_Ypg

1089805705198

메시지 전송용 서버 키가 Firebase Cloud Messaging 토큰으로 대체되었습니다. 서버 키를 계속 사용할 수도 있지만 서버 키에 대한 지원은 중단되었습니다.

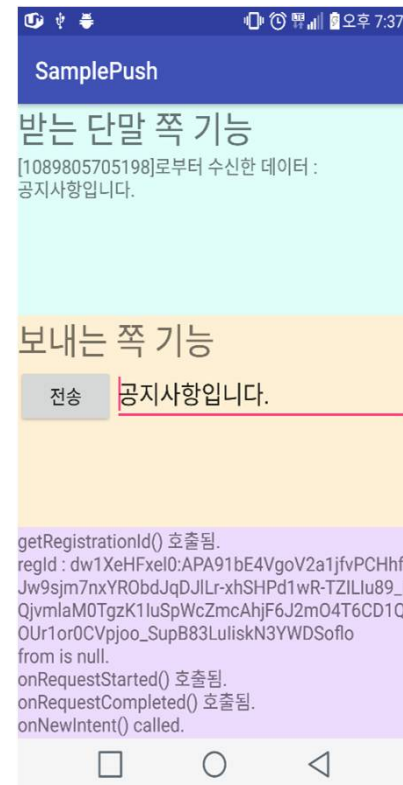
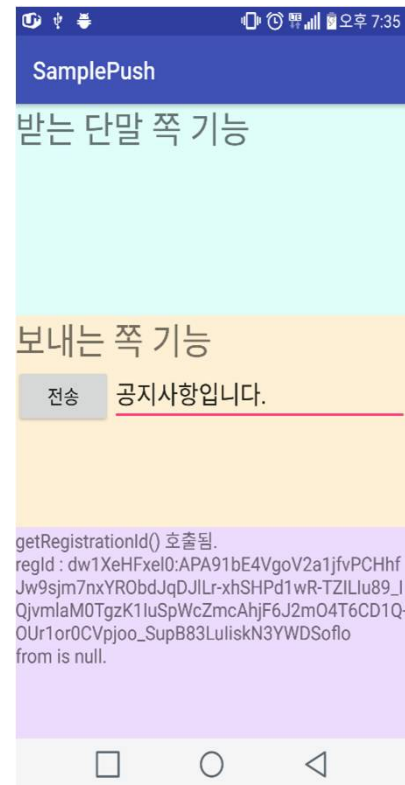
서버 키

AlzaSyAVS0SEesatCSiW0-Hx7Y5xPSW_13pTtHM

iOS 앱 구성



실행 화면

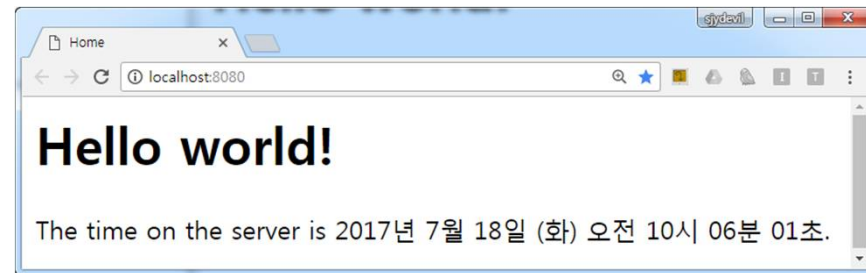


FCM ID등록 및 푸시 메시지 수신 화면



REST 서비스 만들기

- 스프링 프로젝트 생성
- gradle 프로젝트로 변경:
 - build.gradle 복사
 - Configure >> Convert to Gradle(STS) Project
 - Gradle(STS) >> refresh dependency 실행
 - Configure >> Add Gradle Nature
 - Gradle >> Refresh Gradle Project
- src/main/resources/log4j.xml 수정
 - `<priority value="warn" />` ⇒ `<priority value="debug" />` 로
- src/main/webapp/WEB-INF/web.xml 수정
 - 인코딩 필터 설정
 - JSP 파일 UTF-8 인코딩 설정
- 프로젝트 실행
- 클래스 추가
 - AndroidPushNotificationsService.java 추가
 - HeaderRequestInterceptor.java 추가
 - FcmController.java 추가





서버 키 값 입력

- Firebase 콘솔 페이지의 설정 메뉴에서 서버 키 값 확인하여 입력

