

# XE 홈페이지와 연동되는 GCM을 이용한 푸시 안드로이드 앱 만들기 완벽 가이드

## 기본기능

- 1) 앱 실행시 곧바로 모바일 페이지로 링크됨. 웹뷰 방식이 아니라, 폰 자체의 브라우저를 호출하여 모바일 페이지 링크하는 방식이기 때문에 첨부파일 등의 문제가 없음.
- 2) 새글, 새로운 댓글, 쪽지 푸시 알림 서비스(쪽지의 경우는 커뮤니케이션 모듈이 활성화되어 있어야 합니다)
- 3) 모바일 페이지에서 알림여부(새글, 댓글, 쪽지, 소리, 진동) 설정 가능 / 자신이 쓴 글의 댓글이나 자신이 댓글을 단 글에 새댓글이 달렸을 경우에만 알림 되도록 설정 가능 / 앱을 설치하지 않은 사람이 알림설정을 누를 경우 자동으로 구글 플레이에 접속되어 앱 설치를 유도.
- 4) 푸시 알림 서비스를 제외할 게시판 설정 가능
- 5) reg\_id 자동 정리 및 발송 결과 모듈 관리자 페이지에서 확인할 수 있도록 함. 앱을 삭제하였거나 재설치하였을 때 있게 되는 없는 reg\_id와 중복 reg\_id 삭제 처리
- 6) 등록기기 동기화 여부 목록 확인 가능
- 7) 정상작동 여부 확인을 위한 관리자용 테스트 발송기능

-----

일반적으로 앱을 만들기 위해서는 자바와 이클립스 툴을 잘 다루어야 하지만, 저는 전적으로 초보이며 자바를 잘 모르는 사람입니다. 그러나 인터넷 여기저기 떠돌아다니는 자바소스를 가지고서 이리저리 수정하여 푸시 앱을 만드는데 성공하였습니다. 그 과정에서 여러모로 삽질을 많이 했는데, 저처럼 삽질하는 사람들에게 조금이나마 도움이 되었으면 해서 앱 만드는 전 과정을 상세하게 안내해드리겠습니다. 아래의 방법을 그대로 따라하시면 초보자라도 쉽게 앱을 똑딱 만드실 수 있습니다.

## 1. jdk, 안드로이드 스튜디오 설치

먼저 jdk와 안드로이드 스튜디오를 설치합니다. 아래의 링크로 가셔서 각각 설치해주시면 됩니다.

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

이 링크에 가셔서 스크롤을 내려 보면 아래 그림의 내용이 있는데 동의한 후에 32bit 컴퓨터는 x86을 64bit컴퓨터는 x64버전을 다운받아 설치하시면 됩니다.

Visit Java CPU and PSU Releases Explained for details.

Looking for JDK on ARM?

JDK 7 for ARM downloads have moved to the JDK 7 for ARM download page.

Java SE Development Kit 7u79		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input checked="" type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	130.4 MB	jdk-7u79-linux-i586.rpm
Linux x86	147.6 MB	jdk-7u79-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u79-linux-x64.rpm
Linux x64	146.4 MB	jdk-7u79-linux-x64.tar.gz
Mac OS X x64	196.89 MB	jdk-7u79-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.79 MB	jdk-7u79-solaris-i586.tar.Z
Solaris x86	96.66 MB	jdk-7u79-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.67 MB	jdk-7u79-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u79-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140 MB	jdk-7u79-solaris-sparc.tar.Z
Solaris SPARC	99.4 MB	jdk-7u79-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24 MB	jdk-7u79-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.4 MB	jdk-7u79-solaris-sparcv9.tar.gz
Windows x86	138.31 MB	jdk-7u79-windows-i586.exe
Windows x64	140.06 MB	jdk-7u79-windows-x64.exe

<http://developer.android.com/sdk/index.html> -> 안드로이드 스튜디오 설치

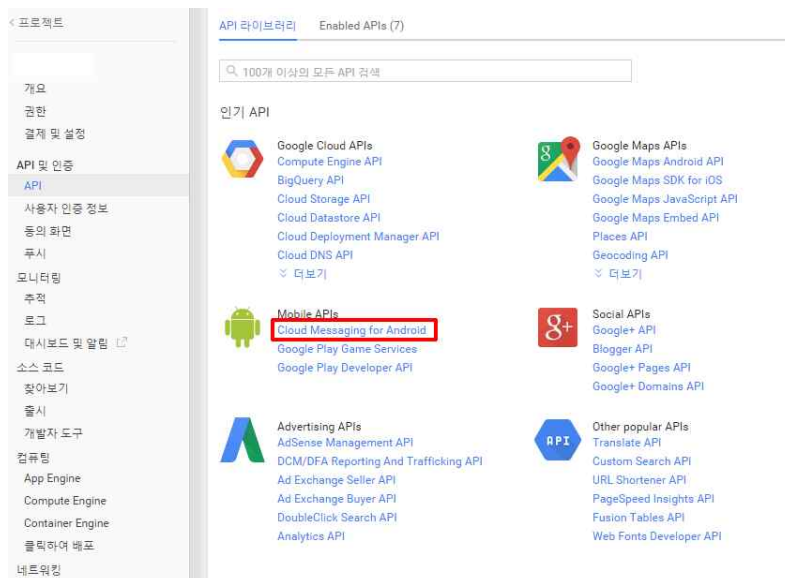
## 2. GCM API 키 받기

구글에서 무료로 무제한 GCM 서비스를 제공해주고 있습니다. 물론 무료로 푸시 서버를 제공해주는 곳이 몇몇 있지만, 우리는 GCM을 이용해보도록 하겠습니다. 우선 아래의 페이지 (google developer console)에 들어가서 구글 계정으로 로그인을 합니다.

<https://console.developers.google.com/>

거기서 create project를 클릭하여 새로운 프로젝트를 생성합니다.

그리고 프로젝트 안에 들어가서 API 및 인증 -> API 를 클릭하면 아래의 그림과 같이 나옵니다. 거기서 빨간색 표시의 Cloud Messaging for Android를 클릭하여 사용함을 클릭합니다.



그 다음에 “사용자 인증 정보”에 들어가셔서 서버 API key를 생성합니다. 여기서 OAuth 2.0이 아니라 공개 API 키로 받으셔야 하고, 공개 API 중에서 “서버 API키”를 받으셔야 합니다.

그러면 GCM 서버 설정은 끝이 납니다. 여기서 프로젝트 번호 와 API key 두 가지를 메모장에 카피해 놓으셔야 합니다. 프로젝트 번호는 아래의 그림처럼 “개요”를 누르시면 화면상에 나오는 숫자들입니다.



그리고 API key는 아래 그림처럼 API 및 인증 -> 사용자 인증 정보를 클릭하시면 나옵니다.



### 3. 안드로이드 스튜디오 시작하기

이제 안드로이드 스튜디오를 시작하기 전에 자바 path 환경설정을 해주셔야 합니다. 인터넷

에서 “자바 path 환경설정”을 검색하시거나 아래의 링크로 들어가셔서 확인하시면 됩니다.

<http://cwj3688.blogspot.kr/2013/11/path-javahome.html>

이 링크의 1번 2번을 다 해주셔야 합니다. 꼭 cmd 실행해서 javac 입력 후 반응이 있어야지만 안드로이드 스튜디오로 앱을 만들 수 있습니다. 그 다음에 안드로이드 스튜디오 기본 설정과 작업환경. 간단한 툴 사용법에 대해서는 아래 링크에 있는 강좌들을 꼭 한번 훑어보시면 도움이 되실 것입니다.

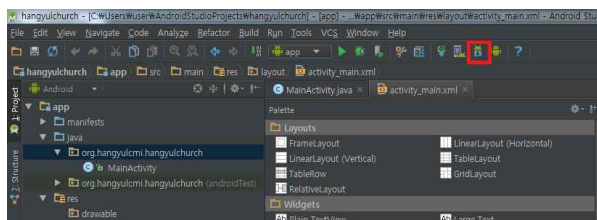
<http://blog.naver.com/eominsuk55/220214235303>

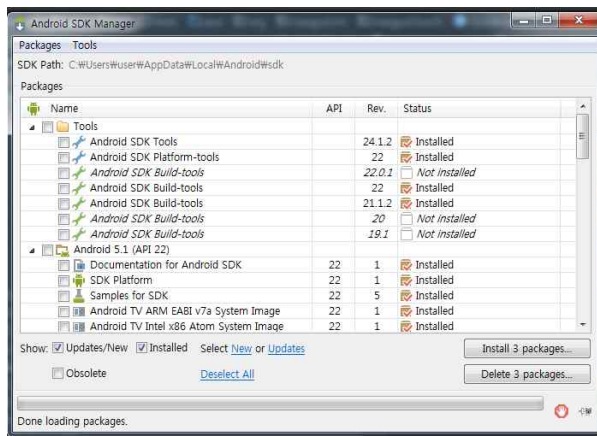
이 강좌 밑 부분에 보면 총 20개가 넘는 강좌가 연속으로 계속되는데, 대충 훑어보시면 이후 작업이 보다 수월할 수 있고, 또 이 가이드에 제시된 앱보다 더 향상된 앱을 스스로 개발하실 수 있습니다. 물론 안 보셔도 괜찮습니다. ^^

그러면 안드로이드 스튜디오를 시작해보겠습니다. 프로그램을 실행하시고 ‘새로운 프로젝트’를 클릭합니다. 어플리케이션 이름 / 도메인을 입력합니다. 어플리케이션 이름은 유니크하게 (독특하게-다른 어플과 충돌될 수가 있기 때문에) 지어주세요. 패키지명은 자동으로 설정됩니다. **패키지명도 따로 메모장에 카피해둡니다.** 그 다음 안드로이드 타겟 버전은 2.3(진저브레드)으로 하시면 됩니다. 그리고 다음에 이어지는 설정들에 대해서는 기본값 그대로 가시면 됩니다.

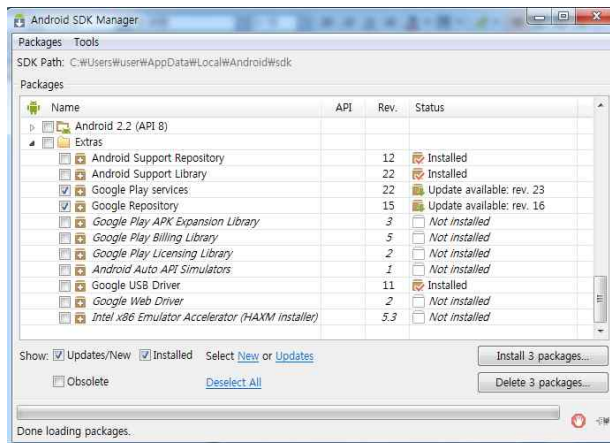
## 4. sdk 매니저 실행

한참 로딩한 후에 다음과 같은 작업 화면이 뜨면 제일 먼저 해야 할 작업은 sdk를 다운받는 것입니다. 아래 그림의 빨간 네모 부분을 클릭합니다.





여기서 기본적으로 체크되어 있는 거 그대로 두시면 되고, 가장 중요한 것은 스크롤을 내려 주시면 밑에 있는 다음을 꼭 설치해주셔야 합니다.



Google Play services <- 이게 가장 중요하구요. 다른 것은 저도 꼭 설치해야 되는지는 모르겠지만, 일단 위 그림에서 Installed 되어 있는 것(Android Support Repository / Android Support Library / Google Repository)은 다 체크하시기 바랍니다. 그다음에 Install ?? packages... 를 클릭하시고 동의하신 후에 다운로드 설치하시면 됩니다. (시간이 많이 걸립니다.)

## 5. 파일생성

SDK가 다 설치가 되었으면 아래의 파일을 만들어야 합니다. 대문자 소문자 정확하게 지켜야 합니다.

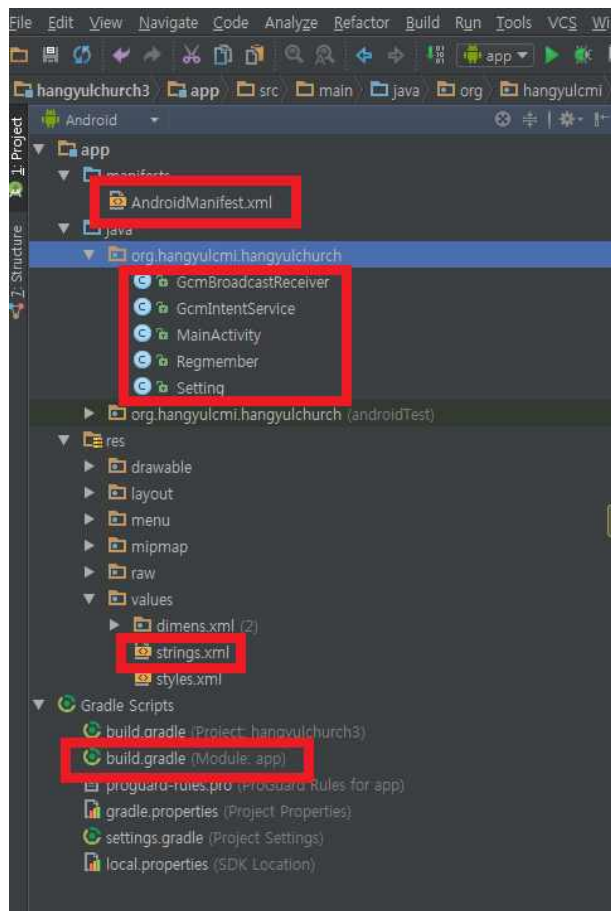
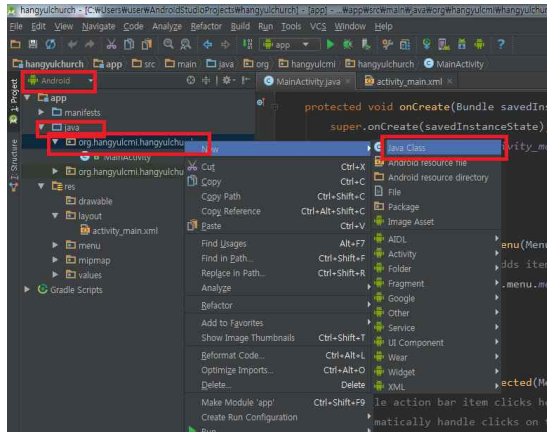
GcmBroadcastReceiver.java

GcmIntentService.java

Setting.java

Regmember.java

만드는 방법은 아래 그림처럼 하시면 됩니다. (app/java/패키지명-> 오른쪽 클릭)



그리고 만들어진 자바 파일들을 다 더블클릭해서 오른쪽 창에 띄워놓습니다. 기본적으로 MainActivity.java 와 Activity\_main.xml는 열려 있습니다. 그리고 아래의 파일들도 다 더블 클릭해서 옆에 띄워놓습니다.

GcmBroadcastReceiver.java

GcmIntentService.java

Setting.java

Regmember.java

Android\_Manifest.xml

build.gradle(Module: app)

Activity\_main.xml

MainActivity.java

strings.xml

## 6. build.gradle(Module: app) 수정

밑에

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.0.0'
}
```

이 부분에 아래의 빨간 줄 하나를 첨가합니다.

```
compile 'com.google.android.gms:play-services:7.0.0'
```

그래서 이렇게...

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:22.0.0'
    compile 'com.google.android.gms:play-services:7.0.0'
}
```

google play service 버전이 업그레이드되었을 경우, 업그레이드 된 버전을 적어주시면 됩니다(최신버전 숫자는 여기서 확인하세요.

-><https://developer.android.com/google/play-services/index.html>). 그리고 sdk 매니저

옆에 있는  이 버튼을 한번 눌러주세요.

## 7. Android\_Manifest.xml 수정

원래 있는 모든 내용을 지우고, 아래의 내용을 완전히 덮어쓰기 하세요. 그리고 빨간 부분을 자신의 패키지명으로 수정하시면 됩니다.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="패키지명ex)com.domain.example">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />

    <permission
        android:name="com.domain.example.permission.C2D_MESSAGE"
        android:protectionLevel="signature" />
    <uses-permission android:name="com.domain.example.permission.C2D_MESSAGE" />
    <application
        android:allowBackup="true"
```



```

android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme">
<activity
    android:name="com.domain.example.MainActivity"
    android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<activity
    android:name="com.domain.example.Setting"
    android:label="@string/app_name">

    <intent-filter >
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.BROWSABLE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="example" android:host="Setting" />
    </intent-filter>

</activity>
<activity
    android:name="com.domain.example.Regmember" />

<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
<receiver
    android:name="com.domain.example.GcmBroadcastReceiver"
    android:permission="com.google.android.c2dm.permission.SEND">
    <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
        <category android:name="com.domain.example" />
    </intent-filter>
</receiver>
<service android:name="com.domain.example.GcmIntentService" />
</application>
</manifest>

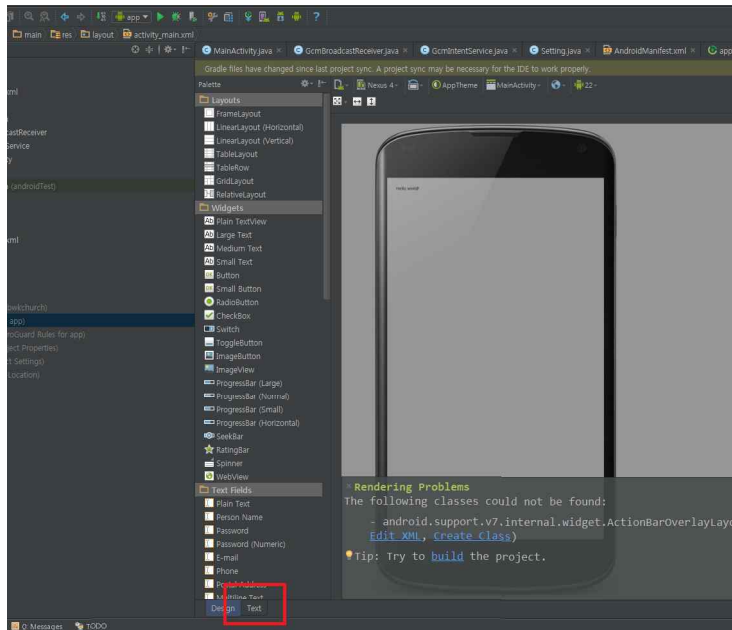
```

위 내용 중에서 특별히 “<data android:scheme="example" android:host="Setting" />” 부분에서는 패키지명이 아닌 패키지명의 끝부분(application name)만 적어주시면 됩니다.

## 8. Activity\_main.xml 수정

아래 그림에서 text를 클릭한 후에 수정합니다.





이 부분은 레이아웃 부분으로서 개인에게 맞게끔 고치시면 되는데, 일단 저는 다음과 같이 만들어보았습니다. 빨간 부분은 설정 창 배경색입니다. 마음대로 바꾸시면 됩니다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" android:background="#ffb3c2"
    android:orientation="vertical"
    android:gravity="center" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="※ 알림 설정(체크된 것만 알림)"
        android:id="@+id/textView33"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="10dp" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:background="#ffb3c2"
        android:orientation="vertical">

        <CheckBox
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="게시판 새글 알림"
            android:id="@+id/checkBoxsort1"
            android:checked="true" />
```

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="새로운 댓글 알림"
    android:id="@+id/checkBoxsort3"
    android:checked="true" />
```

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="나와 관련 있는 것만 알림받기(동기화필요)"
    android:id="@+id/checkBoxsort4"
    android:checked="false"
    android:clickable="false" />
```

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="꼭지 알림받기(동기화필요)"
    android:id="@+id/checkBoxsort5"
    android:checked="false"
    android:clickable="false" android:layout_marginBottom="10dp" />
```

```
</LinearLayout>
```

```
<TextView
    android:layout_width="170dp"
    android:layout_height="wrap_content"
    android:text="※ 소리 / 진동 설정"
    android:id="@+id/textView2"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="15dp" />
```

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="170dp"
    android:layout_height="wrap_content"
    android:gravity="left">
```

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="소리알림"
    android:id="@+id/checkBox"
    android:layout_marginTop="10dp"
    android:checked="true" />
```

```
<CheckBox
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="진동알림"
        android:id="@+id/checkBox2"
        android:layout_marginTop="5dp"
        android:checked="true" />

</LinearLayout>

<Button
    android:layout_width="300dp"
    android:layout_height="60dp"
    android:text="설정 닫기"
    android:id="@+id/button"
    android:gravity="center"
    android:layout_gravity="center"
    android:layout_marginTop="20dp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="현재 로그인된 ID로 동기화"
    android:id="@+id/button2"
    android:layout_gravity="center_horizontal" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:id="@+id/textView66"
    android:layout_gravity="center_horizontal"
    android:layout_marginBottom="10dp" />

</LinearLayout>

```

## 9. MainActivity.java 수정

그리고 가장 중요한 부분입니다. 아래의 부분을

```

public class MainActivity extends ActionBarActivity {

    ...(생략)...

}

```

아래의 내용으로 완전히 덮어쓰기 합니다. 아래의 내용 중에 꼭 빨간 부분을 수정한 후에 카피해서 덮어쓰기 하세요.

```

public class MainActivity extends ActionBarActivity {

    public static final String PROPERTY_REG_ID = "registration_id";
    private static final String PROPERTY_APP_VERSION = "appVersion";
    private static final int PLAY_SERVICES_RESOLUTION_REQUEST = 9000;

    /**
     * Substitute you own sender ID here. This is the project number you got
     * from the API Console, as described in "Getting Started."
     */

    String SENDER_ID = "앞에서 받아두었던 프로젝트 번호";

    /**
     * Tag used on log messages.
     */
    static final String TAG = "GCM Demo";

    GoogleCloudMessaging gcm;
    AtomicInteger msgId = new AtomicInteger();
    Context context;
    String regid;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        context = getApplicationContext();

        // Check device for Play Services APK. If check succeeds, proceed with GCM registration.
        if (checkPlayServices()) {

            gcm = GoogleCloudMessaging.getInstance(this);
            regid = getRegistrationId(context);

            if (regid.isEmpty()) {
                registerInBackground();
            }

        } else {
            Log.i(TAG, "No valid Google Play Services APK found.");
        }

        final SharedPreferences prefs = getGcmPreferences(context);

        if (prefs.contains("soundonoff")) {

        }else{
            initial_set();
        }
    }
}

```

```

        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://홈페이지 주소/"));
        startActivity(intent);
        finish();
    }

    private void initial_set() {
        final SharedPreferences prefs = getGcmPreferences(context);
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString("soundonoff", "true");
        editor.putString("vibonoff", "true");
        editor.putString("sort1", "true");
        editor.putString("sort3", "true");
        editor.putString("sort4", "false");
        editor.putString("sort5", "false");
        editor.commit();
    }

    @Override
    protected void onResume() {
        super.onResume();
        // Check device for Play Services APK.
        checkPlayServices();
    }

    /**
     * Check the device to make sure it has the Google Play Services APK. If
     * it doesn't, display a dialog that allows users to download the APK from
     * the Google Play Store or enable it in the device's system settings.
     */
    private boolean checkPlayServices() {

        int resultCode = GooglePlayServicesUtil.isGooglePlayServicesAvailable(this);

        if (resultCode != ConnectionResult.SUCCESS) {

            if (GooglePlayServicesUtil.isUserRecoverableError(resultCode)) {

                GooglePlayServicesUtil.getErrorDialog(resultCode, this,
PLAY_SERVICES_RESOLUTION_REQUEST).show();

            } else {
                Log.i(TAG, "This device is not supported.");
                finish();
            }
            return false;
        }
        return true;
    }

    /**
     * Stores the registration ID and the app versionCode in the application's

```

```

    * {@code SharedPreferences}.
    *
    * @param context application's context.
    * @param regId registration ID
    */
    private void storeRegistrationId(Context context, String regId) {
        final SharedPreferences prefs = getGcmPreferences(context);
        int appVersion = getAppVersion(context);
        Log.i(TAG, "Saving regId on app version " + appVersion);
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString(PROPERTY_REG_ID, regId);
        editor.putInt(PROPERTY_APP_VERSION, appVersion);
        editor.commit();
    }

    /**
     * Gets the current registration ID for application on GCM service, if there is one.
     * <p>
     * If result is empty, the app needs to register.
     *
     * @return registration ID, or empty string if there is no existing
     *         registration ID.
     */
    private String getRegistrationId(Context context) {

        final SharedPreferences prefs = getGcmPreferences(context);
        String registrationId = prefs.getString(PROPERTY_REG_ID, "");
        if (registrationId.isEmpty()) {
            Log.i(TAG, "Registration not found.");
            return "";
        }
        // Check if app was updated; if so, it must clear the registration ID
        // since the existing regID is not guaranteed to work with the new
        // app version.
        int registeredVersion = prefs.getInt(PROPERTY_APP_VERSION, Integer.MIN_VALUE);
        int currentVersion = getAppVersion(context);
        if (registeredVersion != currentVersion) {
            Log.i(TAG, "App version changed.");
            return "";
        }
        return registrationId;
    }

    /**
     * Registers the application with GCM servers asynchronously.
     * <p>
     * Stores the registration ID and the app versionCode in the application's
     * shared preferences.
     */
    private void registerInBackground() {

```

```

new AsyncTask<Void, Void, String>() {
    @Override
    protected String doInBackground(Void... params) {
        String msg = "";
        try {
            if (gcm == null) {
                gcm = GoogleCloudMessaging.getInstance(context);
            }
            regid = gcm.register(SENDER_ID);
            msg = "Device registered, registration ID=" + regid;

            // You should send the registration ID to your server over HTTP, so it
            // can use GCM/HTTP or CCS to send messages to your app.
            sendRegistrationIdToBackend();

            // For this demo: we don't need to send it because the device will send
            // upstream messages to a server that echo back the message using the
            // 'from' address in the message.

            // Persist the regID - no need to register again.
            storeRegistrationId(context, regid);
        } catch (IOException ex) {
            msg = "Error :" + ex.getMessage();
            // If there is an error, don't just keep trying to register.
            // Require the user to click a button again, or perform
            // exponential back-off.
        }
        return msg;
    }

    @Override
    protected void onPostExecute(String msg) {
        Log.i(TAG, msg);
    }
}.execute(null, null, null);
}

@Override
protected void onDestroy() {
    super.onDestroy();
}

/**
 * @return Application's version code from the {@code PackageManager}.
 */
private static int getAppVersion(Context context) {
    try {
        PackageInfo packageInfo = context.getPackageManager()
            .getPackageInfo(context.getPackageName(), 0);
        return packageInfo.versionCode;
    } catch (PackageManager.NameNotFoundException e) {

```



```

        // should never happen
        throw new RuntimeException("Could not get package name: " + e);
    }
}

/**
 * @return Application's {@code SharedPreferences}.
 */
private SharedPreferences getGcmPreferences(Context context) {
    // This sample app persists the registration ID in shared preferences, but
    // how you store the regID in your app is up to you.
    return getSharedPreferences(MainActivity.class.getSimpleName(),
        Context.MODE_PRIVATE);
}

/**
 * Sends the registration ID to your server over HTTP, so it can use GCM/HTTP or CCS to send
 * messages to your app. Not needed for this demo since the device sends upstream messages
 * to a server that echoes back the message using the 'from' address in the message.
 */
private void sendRegistrationIdToBackend() {
    String url = "http://홈페이지 주소/";
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("act", "procAndroidpushappRegIn"));
    params.add(new BasicNameValuePair("reg_id", regid));
    DefaultHttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(url);
    try {
        httpPost.setEntity(new UrlEncodedFormEntity(params));
    } catch (UnsupportedEncodingException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }

    try {
        HttpResponse httpResponse = httpClient.execute(httpPost);
    } catch (ClientProtocolException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    // Your implementation here.
}
}

```

## 10. GcmBroadcastReceiver 수정

```
public class GcmBroadcastReceiver{
}
```

아래의 내용으로 덮어씹니다.

```
public class GcmBroadcastReceiver extends WakefulBroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // Explicitly specify that GcmIntentService will handle the intent.

        ComponentName comp = new ComponentName(context.getPackageName(),
GcmIntentService.class.getName());
        // Start the service, keeping the device awake while it is launching.
        startWakefulService(context, (intent.setComponent(comp)));
        setResultCode(Activity.RESULT_OK);

    }
}
```

## 11. GcmIntentService.java 수정

이 부분을

```
public class GcmIntentService {
}
```

아래의 내용으로 완전히 덮어쓰기 합니다. 역시 함수내용만 바꾸지 마시고 함수전체를 통째로 바꾸셔야 합니다. 그리고 빨간 부분을 수정한 후에 카피해서 붙여넣기 하세요.

```
public class GcmIntentService extends IntentService {
    public static final int NOTIFICATION_ID = 1;
    private NotificationManager mNotificationManager;
    NotificationCompat.Builder builder;
    String member_srl;
    boolean status1;
    boolean status2;
    String address_real;

    public GcmIntentService() {
        super("GcmIntentService");
    }
    public static final String TAG = "GCM Demo";

    @Override
    protected void onHandleIntent(Intent intent) {
```

```

Bundle extras = intent.getExtras();
String board = intent.getStringExtra("board");
String message = intent.getStringExtra("message");
String dmember = intent.getStringExtra("dmember");
String cmember = intent.getStringExtra("cmember");
String sort = intent.getStringExtra("sort");
String address = intent.getStringExtra("address");

GoogleCloudMessaging gcm = GoogleCloudMessaging.getInstance(this);
// The getMessageType() intent parameter must be the intent you received
// in your BroadcastReceiver.
String messageType = gcm.getMessageType(intent);

if (!extras.isEmpty()) { // has effect of unparcelling Bundle
    /*
     * Filter messages based on message type. Since it is likely that GCM will be
     * extended in the future with new message types, just ignore any message types you're
     * not interested in, or that you don't recognize.
     */
    if (GoogleCloudMessaging.MESSAGE_TYPE_SEND_ERROR.equals(messageType)) {
        sendNotification("Send error: " + extras.toString(), "", "", "", "", "");
    } else if (GoogleCloudMessaging.MESSAGE_TYPE_DELETED.equals(messageType)) {
        sendNotification("Deleted messages on server: " + extras.toString(), "", "", "", "", "");
        // If it's a regular GCM message, do some work.
    } else if (GoogleCloudMessaging.MESSAGE_TYPE_MESSAGE.equals(messageType)) {

        // This loop represents the service doing some work.
        for (int i = 0; i < 5; i++) {
            Log.i(TAG, "Working... " + (i + 1)
                + "/5 @ " + SystemClock.elapsedRealtime());

            try {

                Thread.sleep(5000);

            } catch (InterruptedException e) {

            }

        }
        Log.i(TAG, "Completed work @ " + SystemClock.elapsedRealtime());
        // Post notification of received message.

        sendNotification(message, board, dmember, cmember, sort, address);
        Log.i(TAG, "Received: " + extras.toString());
    }
}

// Release the wake lock provided by the WakefulBroadcastReceiver.
GcmBroadcastReceiver.completeWakefulIntent(intent);
}

private void Noti(String title, String msg, String address, String sort){
    mNotificationManager = (NotificationManager)

```

```

        this.getSystemService(Context.NOTIFICATION_SERVICE);

        if(sort.equals("4")){
            address_real="http://홈페이지/index.php?act=dispCommunicationMessages&message_srl="+address;
        }else{
            address_real="http://홈페이지/"+address;
        }
        PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
            new Intent (Intent.ACTION_VIEW, Uri.parse(address_real)), 0);

        NotificationCompat.Builder mBuilder = new
NotificationCompat.Builder(this).setSmallIcon(R.drawable.ic_stat_gcm)
            .setContentTitle(title)
            .setStyle(new NotificationCompat.BigTextStyle()
                .bigText(msg))
            .setContentText(msg);

        mBuilder.setContentIntent(contentIntent);
        mBuilder.setAutoCancel(true);
        mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());

        if(status1){
            Uri uri =Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.pop);
            Ringtone ring = RingtoneManager.getRingtone(getApplicationContext(), uri);
            ring.play();
        }

        if(status2){
            ((Vibrator) getApplicationContext().getSystemService(
                Context.VIBRATOR_SERVICE)).vibrate(800);
        }
    }

    // Put the message into a notification and post it.
    // This is just one simple example of what you might choose to do with
    // a GCM message.
    private void sendNotification(String msg,String board,String dmember,String cmember,String sort,String
address) {
        final SharedPreferences prefs = getSharedPreferences(MainActivity.class.getSimpleName(),
            Context.MODE_PRIVATE);

        boolean statusa=false;
        boolean statusc=false;
        boolean statusd=false;
        boolean statuse=false;
        status1=false;
        status2=false;
        String title;

        if (prefs.contains("soundonoff")) {
            status1=Boolean.parseBoolean(prefs.getString("soundonoff", ""));

```

```

}
if (prefs.contains("vibonoff")) {
    status2=Boolean.parseBoolean(prefs.getString("vibonoff", ""));
}
if (prefs.contains("sort1")) {
    statusa=Boolean.parseBoolean(prefs.getString("sort1", ""));
}
if (prefs.contains("sort3")) {
    statusc=Boolean.parseBoolean(prefs.getString("sort3", ""));
}
if (prefs.contains("sort4")) {
    statusd=Boolean.parseBoolean(prefs.getString("sort4", ""));
}
if (prefs.contains("sort5")) {
    statuse=Boolean.parseBoolean(prefs.getString("sort5", ""));
}
if (prefs.contains("member_srl_origin")) {
    member_srl = prefs.getString("member_srl_origin", "");
}

switch(sort){
    case "1" : title= board + "에 새 글이 올라왔습니다.";
        if(dmember.equals(member_srl)) break;
        if(statusa){
            Noti(title,msg,address,sort);
        }
        break;

    case "3" : title= board + "에 새로운 댓글이 올라왔습니다.";

        if(statusc){
            Noti(title,msg,address,sort);
        }

        if(statusd){
            String[] array = cmember.split("-");
            if(array[0].equals(member_srl)) break;

            if(dmember.equals(member_srl)){
                Noti(title,msg,address,sort);
            }else{
                boolean comment=false;
                for(int i=0; i< array.length; i++){
                    if(array[i].equals(member_srl)) comment=true;
                }
                if(comment) {
                    Noti(title,msg,address,sort);
                }
            }
        }
        break;

```

```

        case "4" : title= "쪽지가 도착했습니다.";
            if(status){
                Noti(title,msg,address,sort);
            }
            break;
        case "10" : title= "테스트";

            Noti(title,msg,address,sort);

            break;
    }
}
}

```

## 12. Setting.java 수정

```

public class Setting {
}

```

아래의 내용으로 덮어씹니다.

```

public class Setting extends ActionBarActivity {

    public static final String PROPERTY_REG_ID = "registration_id";

    Context context;
    Button btn2;
    CheckBox checkBox;
    CheckBox checkBox2;
    CheckBox checkBoxsort1;
    CheckBox checkBoxsort3;
    CheckBox checkBoxsort4;
    CheckBox checkBoxsort5;

    boolean status1 = false;
    boolean status2 = false;
    boolean statusa = false;
    boolean statusc = false;
    boolean statusd = false;
    boolean statuse = false;
    String member_srl;
    String nick_name;
    String nick_name_origin;
    String member_srl_origin;
    String registrationId;
    Intent intent;
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    context = getApplicationContext();
    this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

    TextView textView = (TextView) findViewById(R.id.textView66);

    checkBox=(CheckBox) findViewById(R.id.checkBox);
    checkBox2=(CheckBox) findViewById(R.id.checkBox2);
    checkBoxsort1=(CheckBox) findViewById(R.id.checkBoxsort1);
    checkBoxsort3=(CheckBox) findViewById(R.id.checkBoxsort3);
    checkBoxsort4=(CheckBox) findViewById(R.id.checkBoxsort4);
    checkBoxsort5=(CheckBox) findViewById(R.id.checkBoxsort5);
    btn2 = (Button) findViewById(R.id.button2);
    Button btn1 = (Button) findViewById(R.id.button);

    final SharedPreferences prefs = getGcmPreferences(context);
    registrationId = prefs.getString(PROPERTY_REG_ID, "");

    if(prefs.contains("soundonoff"))
    {
        status1 = Boolean.parseBoolean(prefs.getString("soundonoff", ""));
        if (status1) {
            checkBox.setChecked(true);
        } else {
            checkBox.setChecked(false);
        }
    }

    if(prefs.contains("vibonoff"))
    {
        status2 = Boolean.parseBoolean(prefs.getString("vibonoff", ""));
        if (status2) {
            checkBox2.setChecked(true);
        } else {
            checkBox2.setChecked(false);
        }
    }

    if(prefs.contains("sort1"))
    {
        statusa = Boolean.parseBoolean(prefs.getString("sort1", ""));
        if (statusa) {
            checkBoxsort1.setChecked(true);
        } else {
            checkBoxsort1.setChecked(false);
        }
    }
}

```



```

if(prefs.contains("sort3"))
{
    statusc = Boolean.parseBoolean(prefs.getString("sort3", ""));
    if (statusc) {
        checkBoxsort3.setChecked(true);
    } else {
        checkBoxsort3.setChecked(false);
    }
}

if(getIntent()!=null){
    Uri uri = getIntent().getData();
    if(uri != null){
        member_srl = uri.getQueryParameter("member_srl");
        nick_name = uri.getQueryParameter("nick_name");
    }
}

if(prefs.contains("member_srl_origin")&&prefs.contains("nick_name_origin"))
{
    member_srl_origin = prefs.getString("member_srl_origin", "");
    nick_name_origin = prefs.getString("nick_name_origin", "");

    Log.i("Setting",member_srl_origin+nick_name_origin+member_srl+nick_name );

    checkBoxsort4.setClickable(true);
    checkBoxsort5.setClickable(true);
    textView.setText(nick_name_origin+"님의 이름으로 동기화 완료된 상태입니다.");

    if(member_srl != null){
        if(member_srl_origin.equals(member_srl)){
            btn2.setText("다른 ID로 동기화하시려면 다른 ID로 로그인 해주세요.");
        }else{
            btn2.setText(nick_name+"님 ID로 재동기화 하기");
        }
    }
}

if(prefs.contains("sort4"))
{
    statusd = Boolean.parseBoolean(prefs.getString("sort4", ""));
    if (statusd) {
        checkBoxsort4.setChecked(true);
    } else {
        checkBoxsort4.setChecked(false);
    }
}

if(prefs.contains("sort5"))
{
    statuse = Boolean.parseBoolean(prefs.getString("sort5", ""));
    if (statuse) {
        checkBoxsort5.setChecked(true);
    }
}

```

```

        } else {
            checkBoxsort5.setChecked(false);
        }
    }
}

intent = new Intent(this, Regmember.class);
intent.putExtra("member_srl",member_srl);
intent.putExtra("registrationId",registrationId);

checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
{
    @Override
    public void onCheckedChanged (CompoundButton
buttonView,
                                boolean isChecked){
        // TODO Auto-generated method stub
        if (isChecked) {
            Toast.makeText(getApplicationContext(), "소리 ON",
                Toast.LENGTH_LONG).show();
            status1 = true;
            SharedPreferences.Editor editor = prefs.edit();
            editor.putString("soundonoff", status1 + "");
            editor.commit();
        } else {
            Toast.makeText(getApplicationContext(), "소리 OFF",
                Toast.LENGTH_LONG).show();
            status1 = false;
            SharedPreferences.Editor editor = prefs.edit();
            editor.putString("soundonoff", status1 + "");
            editor.commit();
        }
    }
});

checkBox2.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
{
    @Override
    public void onCheckedChanged (CompoundButton
buttonView,
                                boolean isChecked){
        // TODO Auto-generated method stub
        if (isChecked) {
            Toast.makeText(getApplicationContext(), "진동 ON",
                Toast.LENGTH_LONG).show();
            status2 = true;
            SharedPreferences.Editor editor = prefs.edit();
            editor.putString("vibonoff", status2 + "");
            editor.commit();
        } else {

```



```

        statusc = true;
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString("sort3", statusc + "");
        editor.commit();
        checkBoxsort4.setChecked(false);

    } else {
        Toast.makeText(getApplicationContext(), "댓글 알
        림 OFF",

        Toast.LENGTH_LONG).show();
        statusc = false;
        SharedPreferences.Editor editor = prefs.edit();
        editor.putString("sort3", statusc + "");
        editor.commit();
    }
}

);

checkBoxsort4.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
{
    @Override
    public void onCheckedChanged (CompoundButton
buttonView,

        boolean isChecked){
        // TODO Auto-generated method stub
        if (isChecked) {
            Toast.makeText(getApplicationContext(), "내가 쓴
            글에 한정하여 알림 ON",

            Toast.LENGTH_LONG).show();
            statusd = true;
            SharedPreferences.Editor editor = prefs.edit();
            editor.putString("sort4", statusd + "");
            editor.commit();
            checkBoxsort1.setChecked(false);
            checkBoxsort3.setChecked(false);

        } else {
            Toast.makeText(getApplicationContext(), "내가 쓴
            글에 한정하여 알림 OFF",

            Toast.LENGTH_LONG).show();
            statusd = false;
            SharedPreferences.Editor editor = prefs.edit();
            editor.putString("sort4", statusd + "");
            editor.commit();
            checkBoxsort1.setChecked(true);
            checkBoxsort3.setChecked(true);
        }
    }
}

);

```

```

checkBoxsort5.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener()
{
    @Override
    public void onCheckedChanged (CompoundButton
buttonView,
                                boolean isChecked){
        // TODO Auto-generated method stub
        if (isChecked) {
            Toast.makeText(getApplicationContext(), "쪽지 알
림 ON",
                                Toast.LENGTH_LONG).show();
            statuse = true;
            SharedPreferences.Editor editor = prefs.edit();
            editor.putString("sort5", statuse + "");
            editor.commit();

        } else {
            Toast.makeText(getApplicationContext(), "쪽지 알
림 OFF",
                                Toast.LENGTH_LONG).show();
            statuse = false;
            SharedPreferences.Editor editor = prefs.edit();
            editor.putString("sort5", statuse + "");
            editor.commit();
        }
    }
});

btn1.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick (View v){
        finish();
    }
});

btn2.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick (View v){
        if(member_srl != null){
            if(member_srl_origin != null){
                if(member_srl_origin.equals(member_srl)){
                    Toast.makeText(getApplicationContext(),
nick_name_origin+"님으로 이미 동기화되어 있습니다. 다른 id로 다시 동기화하시려면 다른 아이디 로그인해주세요.",
                                Toast.LENGTH_LONG).show();
                }else{
                    startActivity(intent);
                    storemember_srl(context, member_srl,nick_name);
                }
            }
        }
    }
});

```

```

        checkBoxsort4.setClickable(true);
        checkBoxsort5.setClickable(true);
        Toast.makeText(getApplicationContext(), "동기화가 완료되었습니다.",
        Toast.LENGTH_LONG).show();
    }
    }else{
        startActivity(intent);
        storemember_srl(context, member_srl,nick_name);
        checkBoxsort4.setClickable(true);
        checkBoxsort5.setClickable(true);
        Toast.makeText(getApplicationContext(), "동기화가 완료되었습니다.",
        Toast.LENGTH_LONG).show();
    }
    }else{
        if(member_srl_origin != null){
            Toast.makeText(getApplicationContext(), nick_name_origin+"님
으로 이미 동기화되어 있습니다. 다른 id로 다시 동기화하시려면 다른 아이디 로그인해주세요.",
            Toast.LENGTH_LONG).show();
        }else{
            Toast.makeText(getApplicationContext(), "먼저 로그인을 해주세요.",
            Toast.LENGTH_LONG).show();
        }
    }
    }
    }
    }
    );
}

private SharedPreferences getGcmPreferences(Context context) {
    // This sample app persists the registration ID in shared preferences, but
    // how you store the regID in your app is up to you.
    return getSharedPreferences(MainActivity.class.getSimpleName(),
        Context.MODE_PRIVATE);
}

private void storemember_srl(Context context, String member_srl, String nick_name) {
    final SharedPreferences prefs = getGcmPreferences(context);
    SharedPreferences.Editor editor = prefs.edit();
    editor.putString("member_srl_origin", member_srl + "");
    editor.putString("nick_name_origin", nick_name + "");
    editor.commit();
}
}

```

### 13. Regmember.java 수정

```
public class Regmember {  
  
}
```

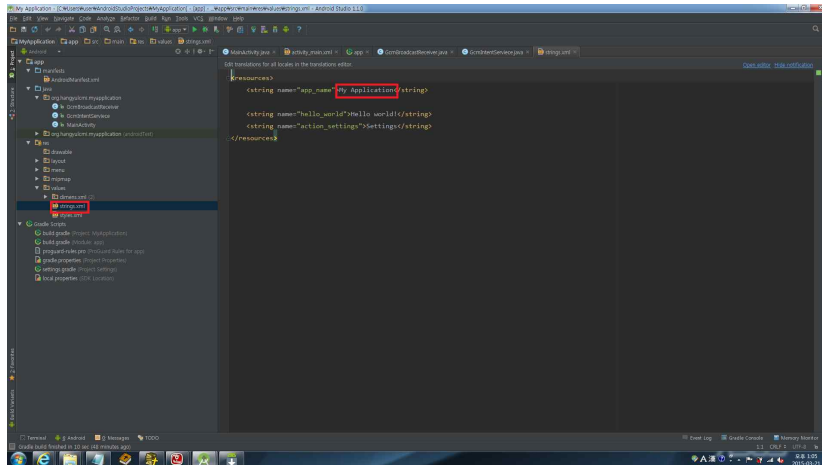
이 부분을 아래의 내용으로 덮어쓰기 합니다. 홈페이지를 수정해주셔서 덮어쓰기 해주세요.

```
public class Regmember extends Setting {  
  
    String member_srl;  
    String registrationId;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        Intent intent=getIntent();  
        member_srl = intent.getStringExtra("member_srl");  
        registrationId = intent.getStringExtra("registrationId");  
  
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();  
        StrictMode.setThreadPolicy(policy);  
        HttpPostData();  
        finish();  
    }  
  
    public void HttpPostData() {  
        String url = "http://홈페이지/";  
        List<NameValuePair> params = new ArrayList<NameValuePair>();  
        params.add(new BasicNameValuePair("act", "procAndroidpushappSync"));  
        params.add(new BasicNameValuePair("reg_id", registrationId));  
        params.add(new BasicNameValuePair("member_srl", member_srl));  
        DefaultHttpClient httpClient = new DefaultHttpClient();  
        HttpPost httpPost = new HttpPost(url);  
        try {  
            httpPost.setEntity(new UrlEncodedFormEntity(params));  
        } catch (UnsupportedEncodingException e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        }  
  
        try {  
            HttpResponse httpResponse = httpClient.execute(httpPost);  
        } catch (ClientProtocolException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        } catch (IOException e) {  
            // TODO Auto-generated catch block  
            e.printStackTrace();  
        }  
    }  
}
```



## 14. strings.xml 수정

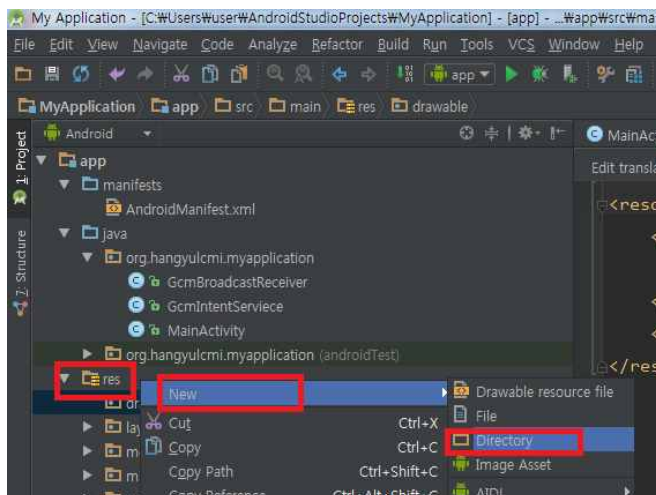
res/values/strings.xml을 더블클릭하셔서 app\_name을 자신이 원하는 명칭으로 바꿉니다.



## 15. 소리알림을 위한 음악파일 집어넣기

알림 올 때 나올 알림음 mp3 파일이 모듈 첨부파일에 포함되어 있습니다. 이 파일을 Ctrl+ C하신 후에 아래그림처럼 폴더 생성 후, 그 폴더에 붙여넣기 하시면 됩니다. 다른 알림음을 사용하시려면 이름만 똑같이 pop.mp3로 하셔서 붙여넣기 하시면 됩니다.

app/res에 오른쪽 클릭 -> new -> directory




폴더명은 raw입니다. 그리고 그 폴더를 오른쪽 클릭한 후 붙여넣기(Paste)하시면 됩니다.

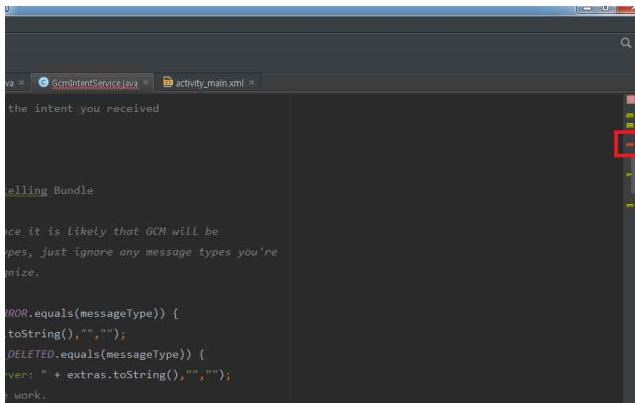
안에 pop.mp3 파일이 잘 들어가 있는지 확인해봅니다.

## 16. 아이콘 만들기

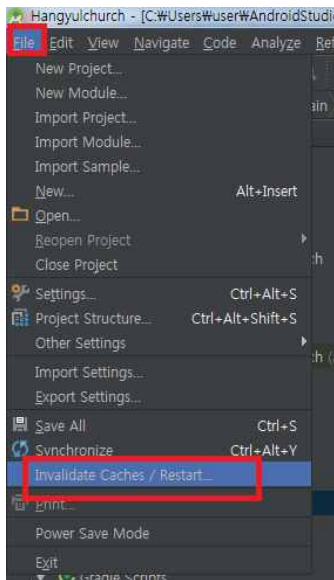
포토샵으로 아이콘을 만들어주세요. 만드실 때 크기는 우선 512x512 크기로 (png형식) 만들어주셔야 합니다. 왜냐하면 구글 플레이에 등록하기 위해서는 이 크기의 아이콘 파일이 필요하기 때문이죠. 그리고 똑같은 png 형식으로 144x144 하나, 72x72 하나 저장하십니다. 72x72크기의 파일은 이름명을 ic\_stat\_gcm.png로 바꾸시고, 144x144는 파일명을 ic\_launcher.png로 바꾸셔서 Ctrl+C를 하십니다. 그리고 음악파일 붙여넣기 하듯이 두 파일 모두 res/drawable 폴더 안에 붙여넣기 합니다.

## 17. 최종 작업


모든 작업을 완료하신 후에  이거 다시 한 번 눌러주시고, 모든 수정한 java파일들이나 xml 파일의 소스에 빨간 줄이 그어져 있는 부분이 있나 없나 확인해봅니다. 빨간 줄이 있거나 글자가 빨갱거나 하는 부분이 있으면 오류가 있는 것이기 때문에 앱 생성이 안 됩니다. 빨간 부분이 있는 확인하는 방법은 스크롤 있는 부분이 빨간색 표시가 있는지 확인하시면 됩니다.

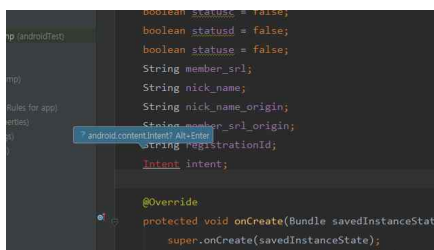


위 그림의 빨간 네모부분이 있으면 안 됩니다. 빨간 부분을 찾아가셔서 잘못 수정하셨는지 다시 확인해주시고, 잘못이 없는데도 빨간 부분이 있다면, 아래의 메뉴를 눌러줍니다.



그리고 Invalidate and restart를 눌러줍니다.

그 후에 다시 보시면 아마도 빨간 부분이 없을 것입니다. 또 있으면 한번 더 를 눌러줍니다. 그리고 빨간 부분 주변에 커서를 갖다 놓고 기다리면 파란 안내문이 뜨는데, Alt+ Enter를 눌러주면 빨간 줄이 없어지기도 합니다.



앞의 과정을 몇 번 반복해서 빨간줄을 다 없애야 합니다. 제발 없기를 기도합니다(있으면 미침 ㅠ,,ㅠ). 만일 그래도 있으면 SDK에 뭔가가 설치가 안 되었다는 말인데, SDK매니저 다시 들어가서 체크되어 있는 부분 다시 다운로드 설치하시기 바랍니다. 그래도 있으면 그냥 포기하시는 게... T..T

## 18. 안드로이드 푸시 앱 연동 모듈 설치

본 모듈을 ftp 서버에 module 폴더에 업로드 하십니다. 그리고 홈페이지로 가서 관리자 로 로그인하시고 관리자페이지에 가면 DB를 생성하는 버튼이 있는데, 그것을 클릭하면 DB가 생성됩니다. 관리자 메뉴 - 고급 - 설치된 모듈 - 안드로이드 푸시 앱 설정에 들어갑니다.

## 안드로이드 푸시 앱

환경설정

푸시 결과값 목록

등록 기기 목록

주의!

푸시 동작여부

☒ 동작

☐ 동작 안 함

API\_Key

google GCM에서 받은 API KEY를 적습니다.

글/댓글 푸시 사용여부

선택한 것만 푸시합니다.

☒ 글

☒ 댓글

☐ 쪽지(커뮤니케이션 모듈과 애드온이 활성화되어 있어야 합니다)

푸시 동작을 사용하지 않을 모듈

선택한 모듈에서는 새글이나 새댓글이 올라와도 푸시가능을 하지 않습니다.

☐ dddd (test)

여기에 API key에는 앞서 받아두었던 API KEY를 적습니다. 그리고 푸시알림을 사용하지 않을 게시판을 선택한 후에 제일 밑에 글/댓글/쪽지 푸시 여부도 설정하시면 됩니다. 그리고 제일 아래에 있는 ‘등록’버튼을 눌러 적용합니다. 푸시 알림 결과 값은 위의 그림의 빨간 표시부분에 들어가시면 확인하실 수 있고 등록기기 목록을 누르시면 현재 앱을 설치한 회원을 확인하실 수 있습니다. 그리고 아래에 “더미 데이터 생성” 버튼을 누르시면 푸시 작동이 정상적으로 되는지 확인하실 수 있습니다(앱이 설치되어 한번은 실행이 되어있어야 하고 관리자 id로 동기화가 되어 있어야 합니다.)

## 19. 모바일 페이지 layout.html 파일 수정

이제 모바일 페이지에 알림설정 버튼이 나오도록 하겠습니다. 앱이 설치된 사람은 클릭하면 설정창으로 이동되고, 앱을 설치하지 않은 사람은 구글 플레이에 접속되어 설치를 유도합니다. 모바일 웹 페이지 layout.html에 상단부분이나 하단 부분에 다음을 추가해주세요. 빨간 부분을 자신에 맞게 수정해주세요. 예를 들어서 패키지명이 com.domain.example이라고 한다면 아래와 같이 바꾸시면 됩니다.

```
<!--@if($is_logged)-->

<script language="JavaScript">

function checkApplicationInstall() {

    var userAgent = navigator.userAgent;
    var visitedAt = (new Date()).getTime();

    if (userAgent.match(/Chrome/)) {
        // 안드로이드의 크롬에서는 intent만 동작하기 때문에 intent로 호출해야함
        setTimeout(function() {
            location.href
            =
            "intent://Setting?member_srl={$logged_info->member_srl}&nick_name={$logged_info->nick_name}#Intent;scheme=
            example;package=com.domain.example;end";
        }, 1000);
    }
}
```

```

    }, 100);
} else {
    // 크롬 이외의 브라우저들
    setTimeout(
        function() {
            if ((new Date()).getTime() - visitedAt < 200) {
                location.href = "market://details?id=com.domain.example";
            }
        }, 50);

    var iframe = document.createElement('iframe');
    iframe.style.visibility = 'hidden';
    iframe.src
'example://Setting?member_srl={$logged_info->member_srl}&nick_name={$logged_info->nick_name}';
    document.body.appendChild(iframe);
    document.body.removeChild(iframe); // back 호출시 캐싱될 수 있으므로 제거
}
}

</script>
<!--@else-->
<script language="JavaScript">

function checkApplicationInstall() {

    var userAgent = navigator.userAgent;
    var visitedAt = (new Date()).getTime();

    if (userAgent.match(/Chrome/)) {
        // 안드로이드의 크롬에서는 intent만 동작하기 때문에 intent로 호출해야함
        setTimeout(function() {
            location.href = "intent://Setting#Intent;scheme=example;package=com.domain.example;end";
        }, 100);
    } else {
        // 크롬 이외의 브라우저들
        setTimeout(
            function() {
                if ((new Date()).getTime() - visitedAt < 200) {
                    location.href = "market://details?id=com.domain.example";
                }
            }, 50);

            var iframe = document.createElement('iframe');
            iframe.style.visibility = 'hidden';
            iframe.src = 'example://Setting';
            document.body.appendChild(iframe);
            document.body.removeChild(iframe); // back 호출시 캐싱될 수 있으므로 제거
        }
    }
}

```

```
</script>

<!--@end-->
```

그리고 아래의 내용을 사이트의 원하는 곳에 첨가합니다.

```
<a href="#" onclick="checkApplicationInstall()">알림설정</a>
```

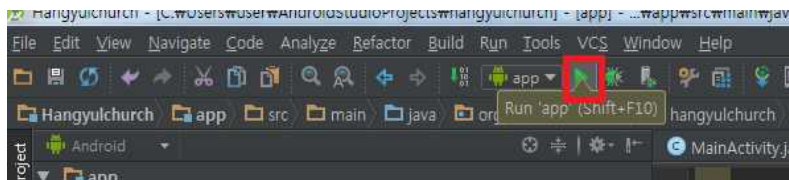
(이 버튼이 제대로 동작하기 위해서는 앱이 구글플레이에 출시되어 있어야만 합니다.)

## 20. 핸드폰 연결 테스트

이제 모든 제작과정이 완료되었습니다. 그러면 테스트 해보겠습니다. 일단 자신의 핸드폰을 PC에 연결합니다. 연결방법은 아래의 링크를 참조하세요.

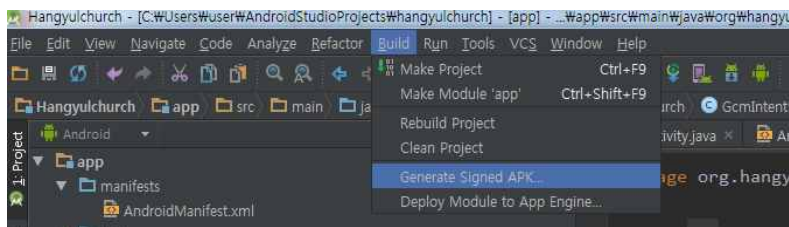
<http://blog.naver.com/eominsuk55/220215691712>

연결하신 후 아래의 빨간 네모부분의 버튼을 눌러주고 연결된 device를 선택하여 실행하시면 됩니다.



제대로 작동되는지 홈페이지에 글이나 댓글을 적으신 후에 기다려 봅니다. 알림이 오는데 약 10~15초 정도 걸립니다. 테스트하신다음에 제대로 작동이 되면 이제 앱을 생성하고 구글 플레이에 등록하겠습니다.

## 21. key 생성, 서명 app 만들기



key생성 서명과정과 앱 생성 과정은 인터넷 검색하시면 자세하게 나옵니다. 생성되는 key 파일은 앞으로 업그레이드할 때 꼭 필요하기 때문에 절대로 분실하시면 안 됩니다. 그리고 구글에 출시한 상태에서 업그레이드판을 다시 출시하려면 build.gradle(Module: app)파일에

서 버전을 올려주셔야 합니다.

```
defaultConfig {  
    applicationId "org.hangyulcmi.hangyulchurch"  
    minSdkVersion 9  
    targetSdkVersion 22  
    versionCode 3 <- 한번 업그레이드 할 때마다 +1해주면 됩니다.  
    versionName "1.2" <- 1.1, 1.2, 1.3....  
}
```

아래의 링크를 참조해주세요.

<http://blog.naver.com/docko/220263331374>

## 22. 구글 플레이 등록

앱을 구글에 등록하려면 일단 개발자 등록을 위해 2~3만원정도 결제해야만 합니다.  
구글플레이 등록방법은 인터넷에 검색하시면 자세히 나와 있습니다.

<http://progagmer.blog.me/204497805>

수고하셨습니다.~!!