

A red arrow-shaped logo pointing to the right, containing the text "Spring+Android" in white.

Spring+Android

# GCM

Several thin, curved lines in dark blue and light grey originate from the left edge and curve upwards and to the right, creating a decorative element in the bottom-left corner.

권기웅

# 1. Android 개발 환경 설치

## 1.1 Eclipse



### Eclipse IDE for Java Developers

163 MB
342,843 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder...

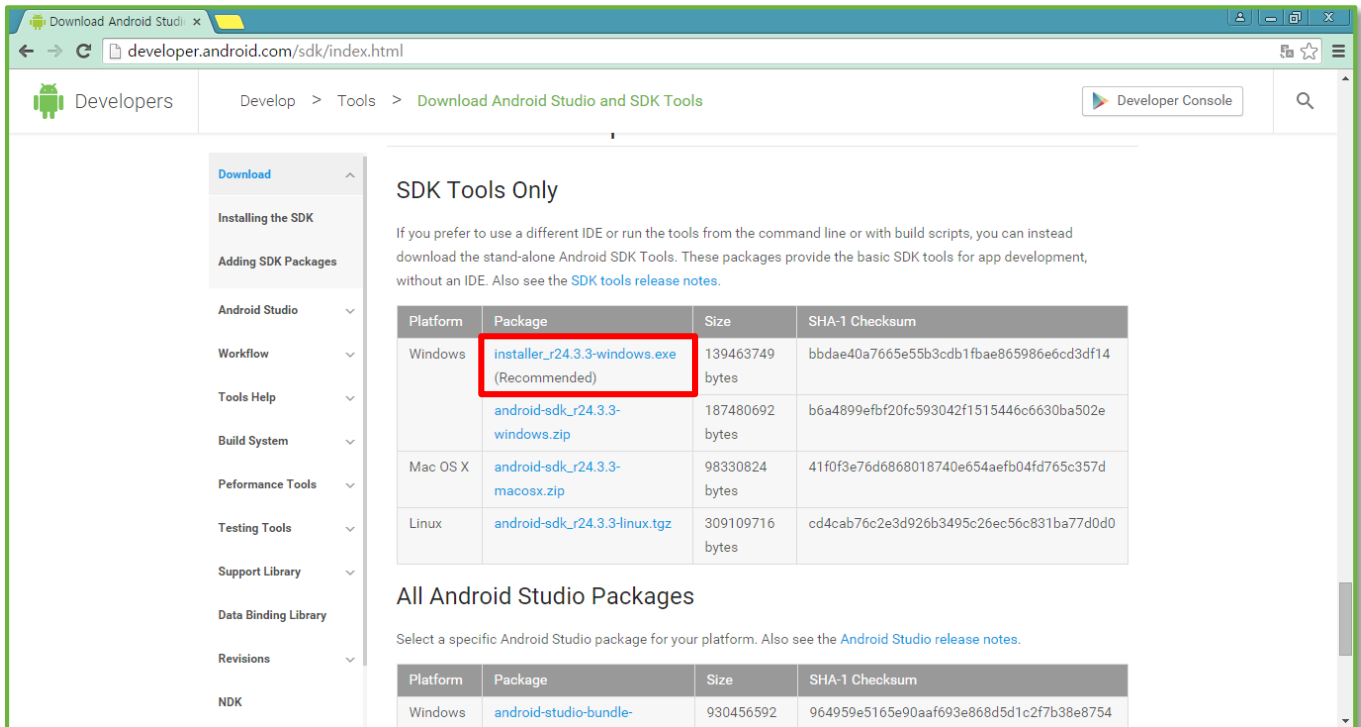
Windows


32 bit | 64 bit

## 1.2 Android SDK

### 1.2.1 SDK Tools 다운로드

<http://developer.android.com/sdk/index.html>



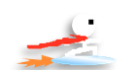
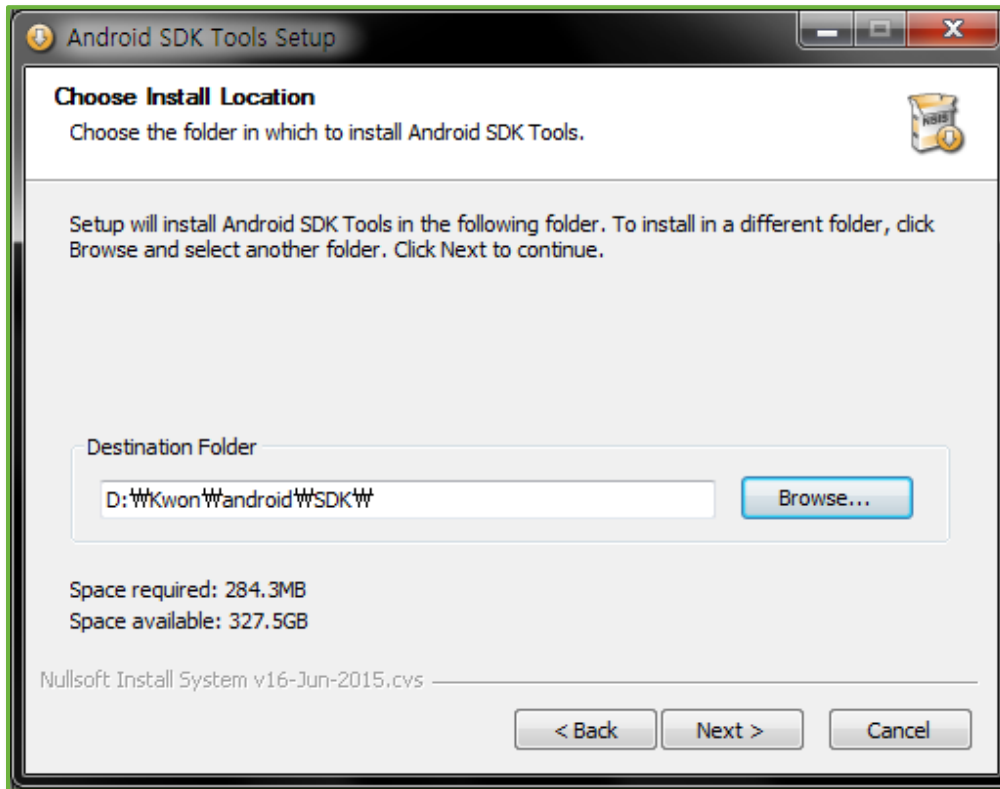
The screenshot shows the 'Download Android Studio and SDK Tools' page. The 'SDK Tools Only' section is highlighted, showing a table of available packages. The 'installer\_r24.3.3-windows.exe' package is recommended and highlighted with a red box.

Platform	Package	Size	SHA-1 Checksum
Windows	<b>installer_r24.3.3-windows.exe (Recommended)</b>	139463749 bytes	bbdae40a7665e55b3cdb1fbae865986e6cd3df14
	<a href="#">android-sdk_r24.3.3-windows.zip</a>	187480692 bytes	b6a4899efbf20fc593042f1515446c6630ba502e
Mac OS X	<a href="#">android-sdk_r24.3.3-macosx.zip</a>	98330824 bytes	41f0f3e76d6868018740e654aefb04fd765c357d
Linux	<a href="#">android-sdk_r24.3.3-linux.tgz</a>	309109716 bytes	cd4cab76c2e3d926b3495c26ec56c831ba77d0d0

Below the table, there is a section for 'All Android Studio Packages' with a similar table structure.

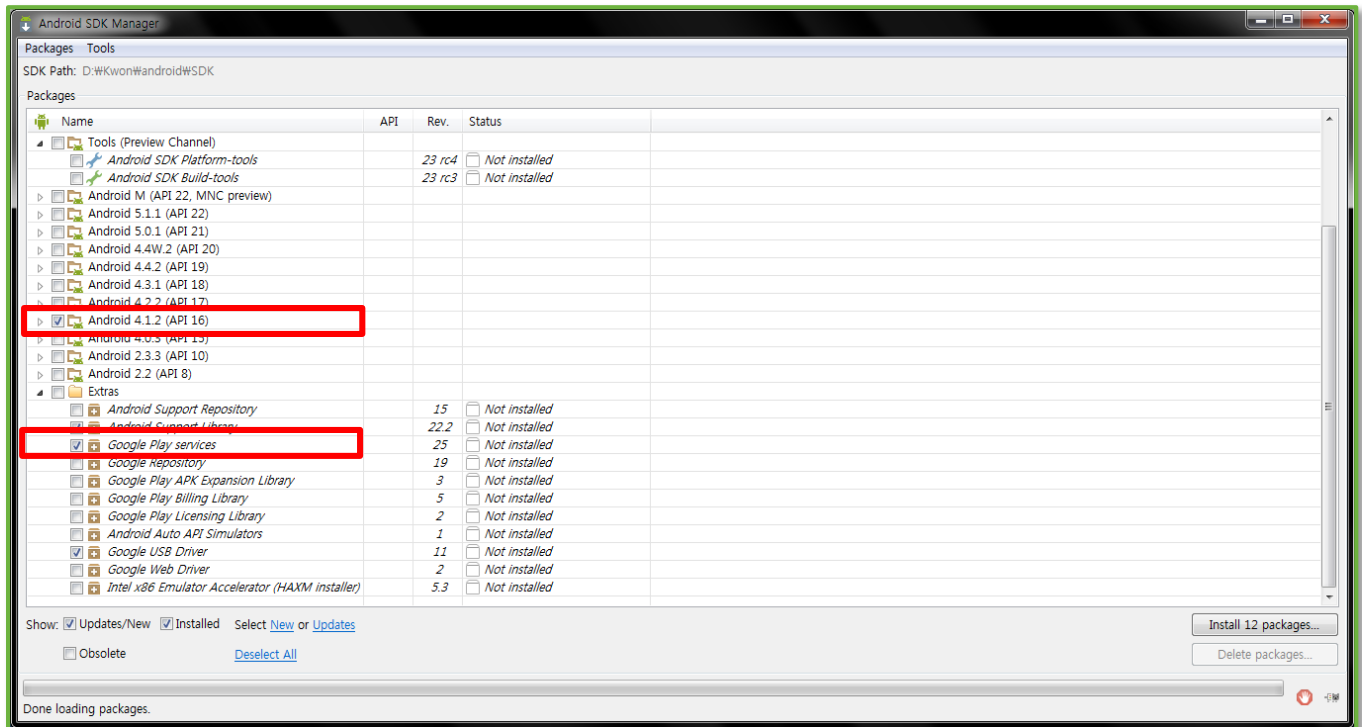


### 1.2.2 SDK Tools 설치



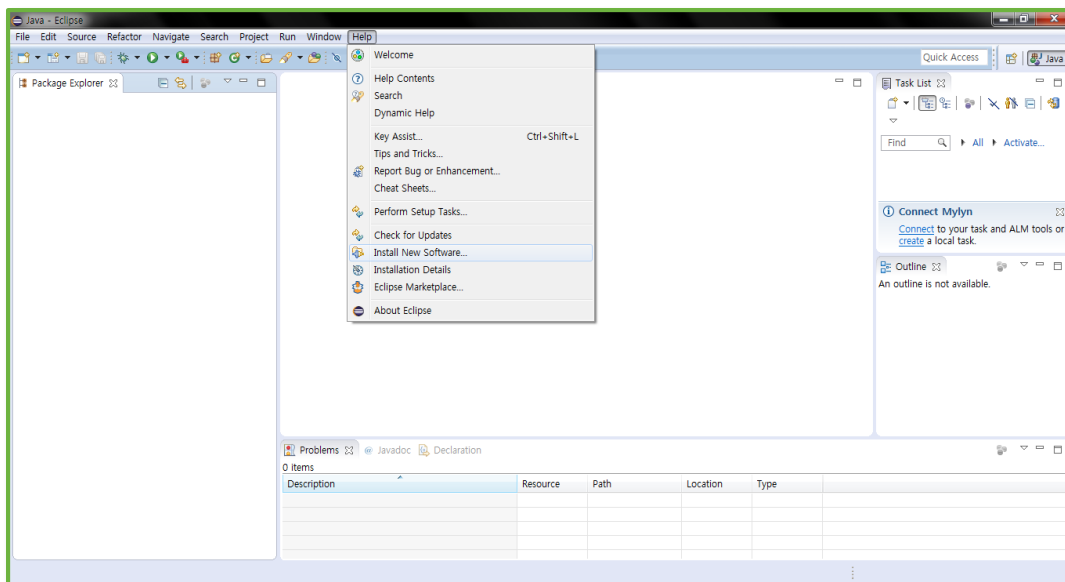


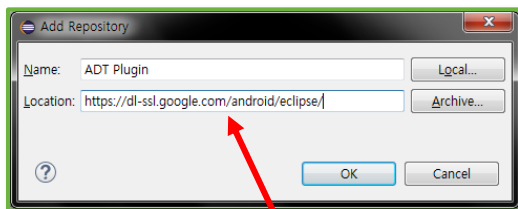
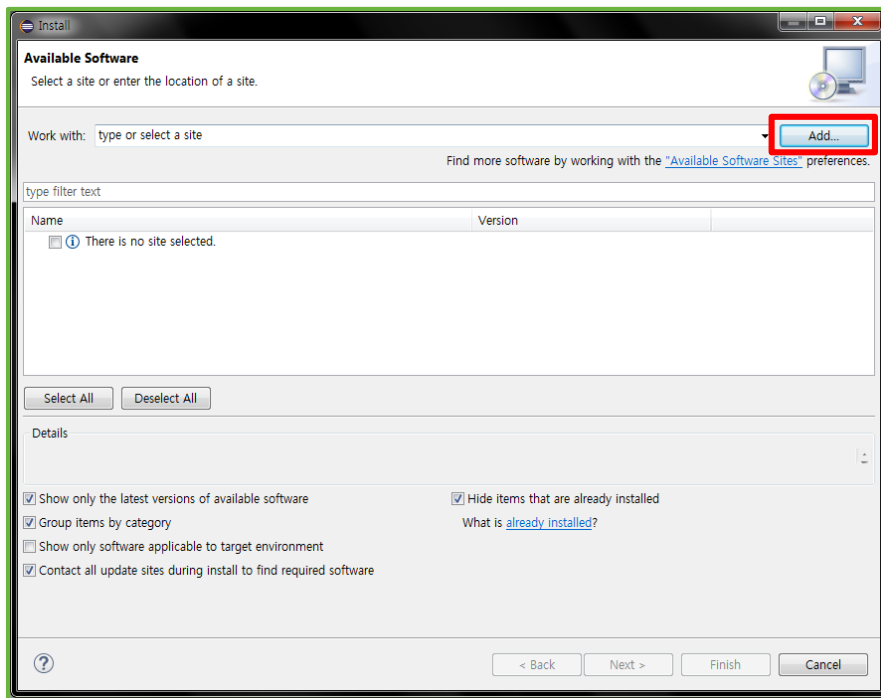
### 1.2.3 Android SDK Manager



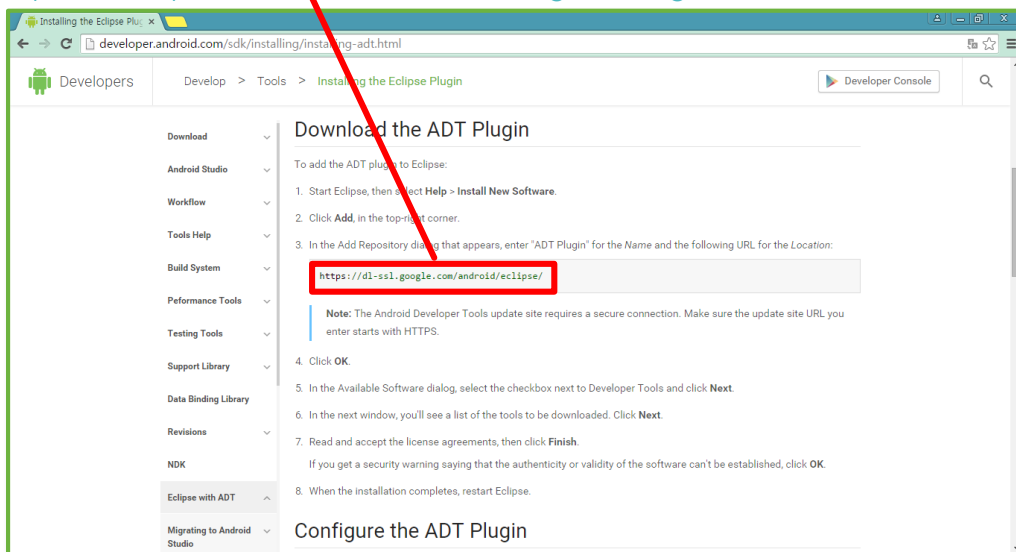
## 1.3 Eclipse ADT Plugin

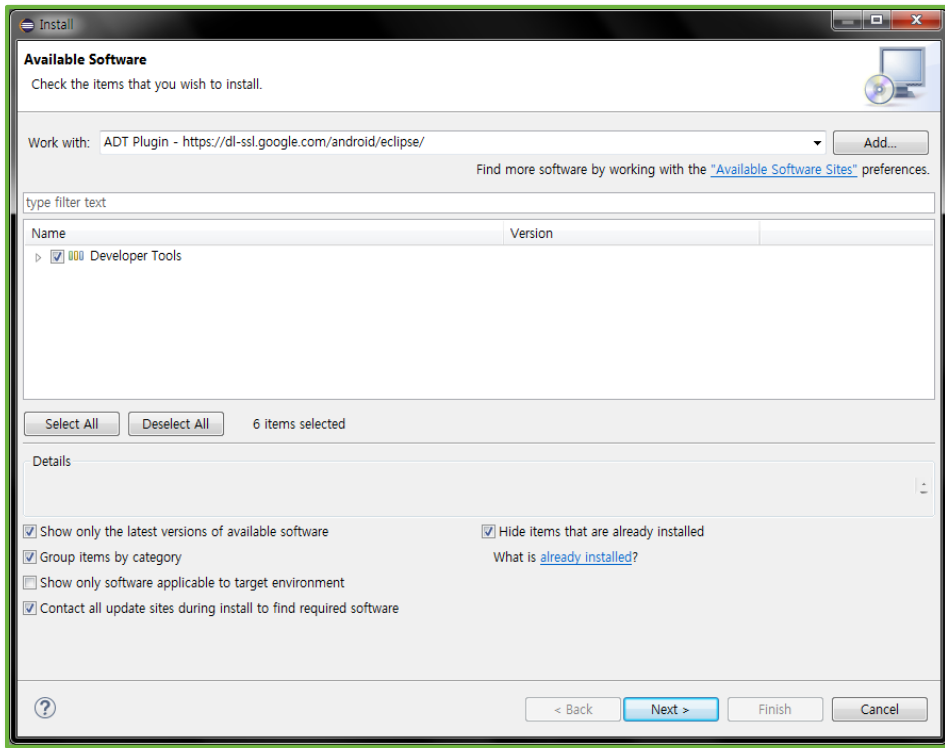
### 1.3.1 Help > Install New Software... > Add





<http://developer.android.com/sdk/installing/installing-adt.html>

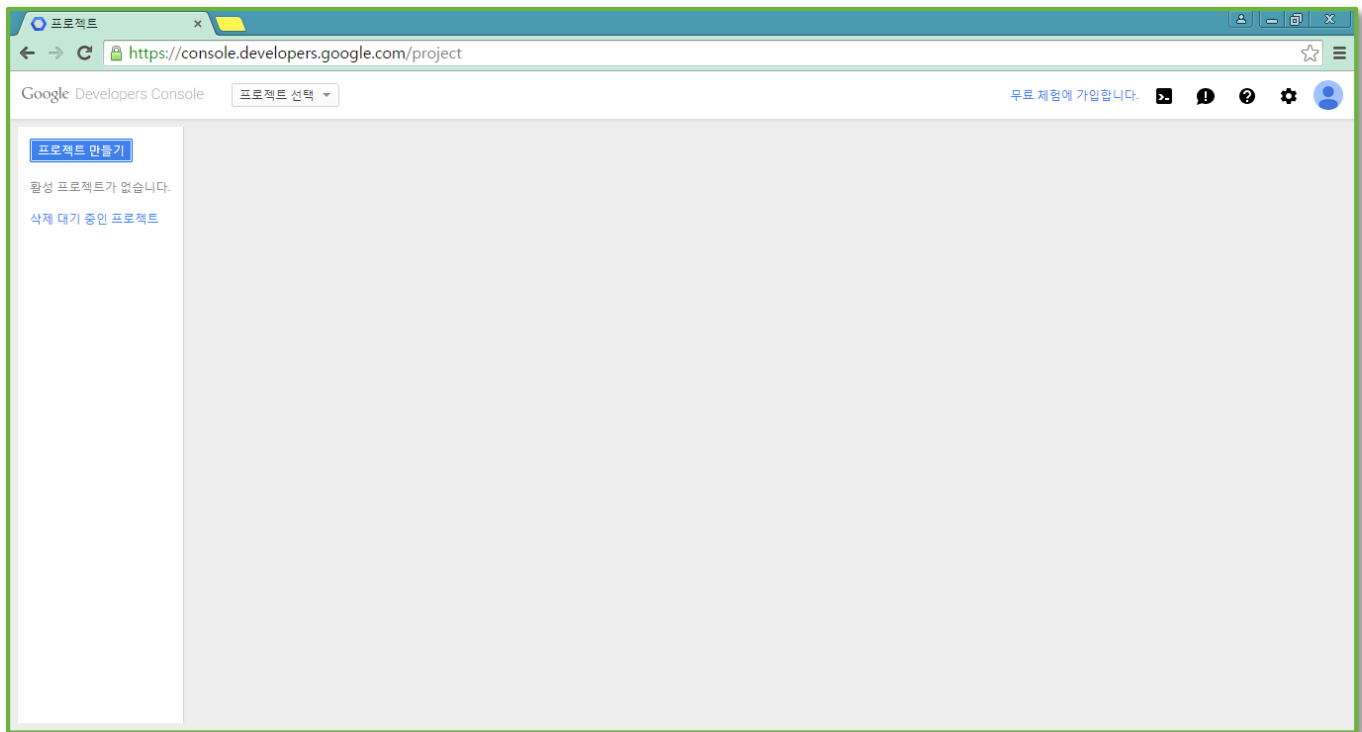




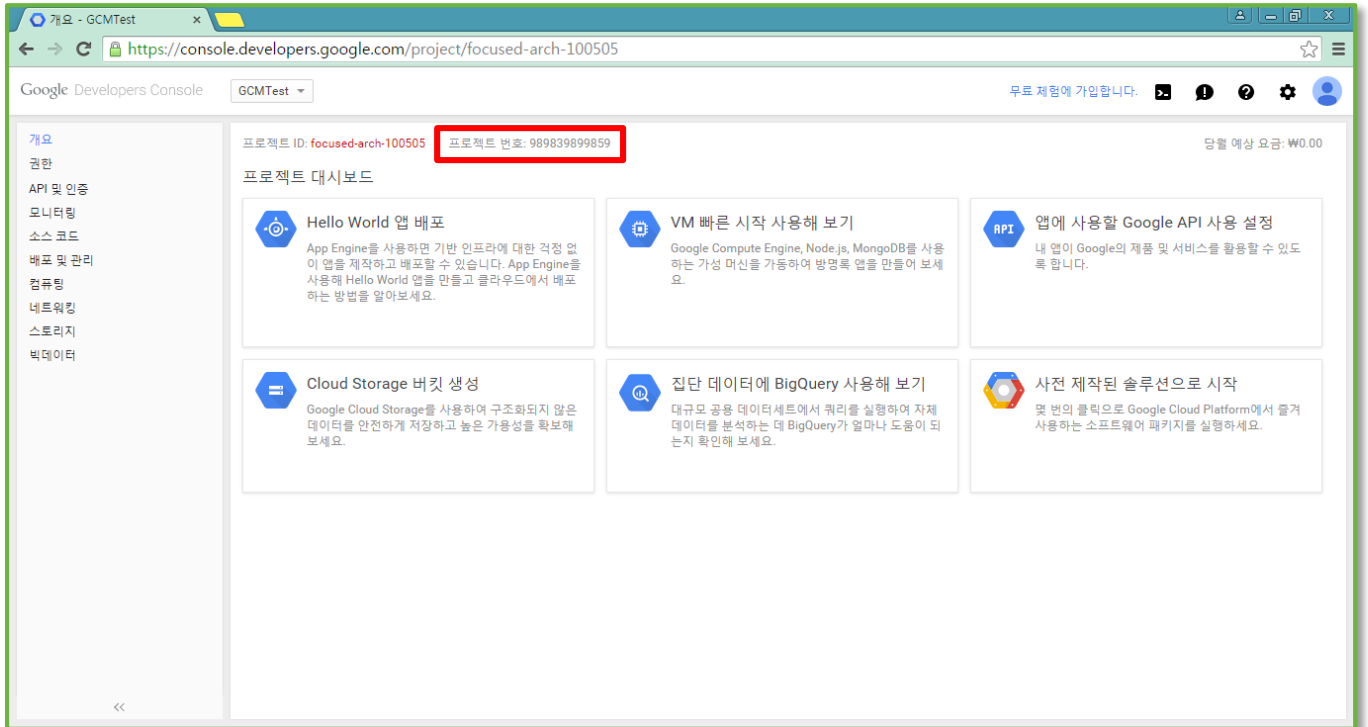
## 2. GCM 서비스 등록

### 2.1 Google 개발자 콘솔

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



## 2.1.1 프로젝트 만들기

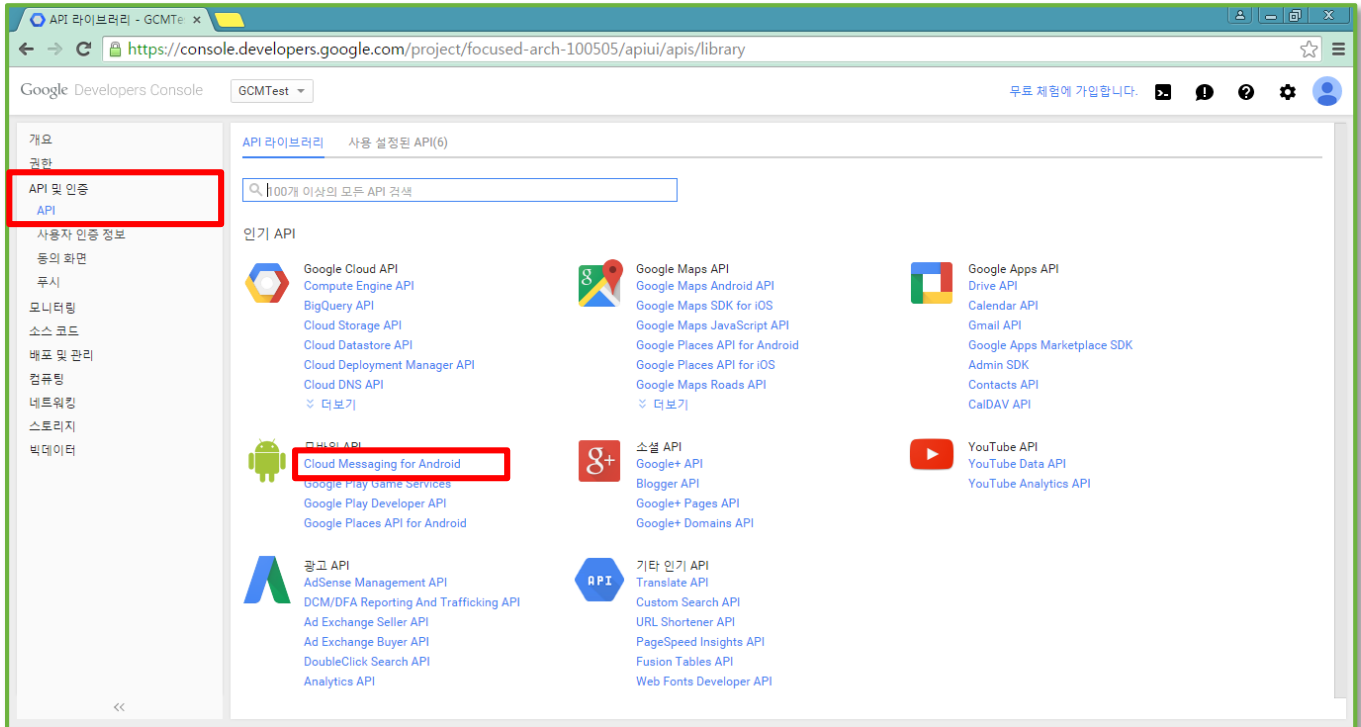


프로젝트 번호 메모





## 2.1.2 사용 API 설정



API 라이브러리 - GCMTest x

← → ↻ <https://console.developers.google.com/project/focused-arch-100505/apiui/apis/library>

Google Developers Console GCMTest 무료 체험에 가입합니다.

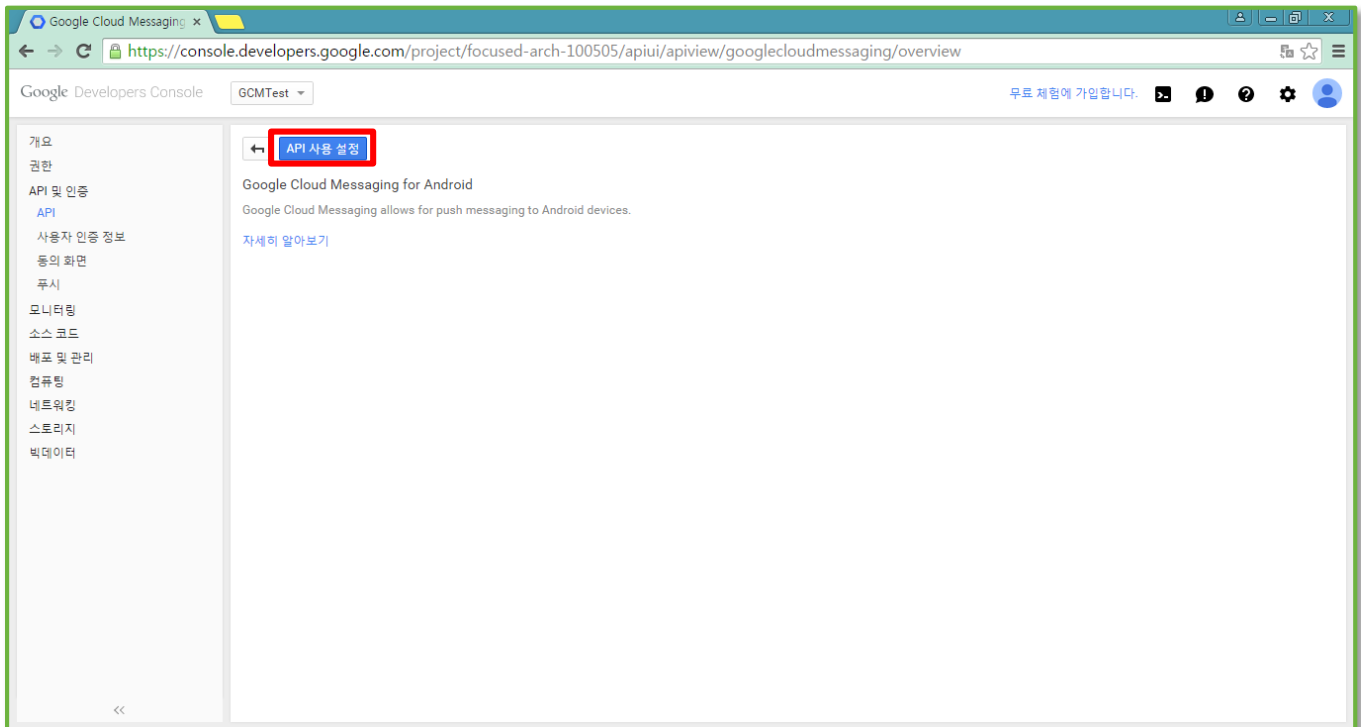
개요 권한 API 및 인증 API 사용자 인증 정보 동의 화면 푸시 모니터링 소스 코드 배포 및 관리 컴퓨팅 네트워킹 스토리지 빅데이터

API 라이브러리 사용 설정된 API(6)

100개 이상의 모든 API 검색

인기 API

- Google Cloud API
  - Compute Engine API
  - BigQuery API
  - Cloud Storage API
  - Cloud Datastore API
  - Cloud Deployment Manager API
  - Cloud DNS API
  - 더 보기
- Google Maps API
  - Google Maps Android API
  - Google Maps SDK for iOS
  - Google Maps JavaScript API
  - Google Places API for Android
  - Google Places API for iOS
  - Google Maps Roads API
  - 더 보기
- Google Apps API
  - Drive API
  - Calendar API
  - Gmail API
  - Google Apps Marketplace SDK
  - Admin SDK
  - Contacts API
  - CalDAV API
- 클라우드 메시징 API
  - Cloud Messaging for Android**
  - Google Play Game services
  - Google Play Developer API
  - Google Places API for Android
- 소셜 API
  - Google+ API
  - Blogger API
  - Google+ Pages API
  - Google+ Domains API
- YouTube API
  - YouTube Data API
  - YouTube Analytics API
- 광고 API
  - AdSense Management API
  - DCM/DFA Reporting And Trafficking API
  - Ad Exchange Seller API
  - Ad Exchange Buyer API
  - DoubleClick Search API
  - Analytics API
- 기타 인기 API
  - Translate API
  - Custom Search API
  - URL Shortener API
  - PageSpeed Insights API
  - Fusion Tables API
  - Web Fonts Developer API



Google Cloud Messaging x

← → ↻ <https://console.developers.google.com/project/focused-arch-100505/apiui/apiview/googlecloudmessaging/overview>

Google Developers Console GCMTest 무료 체험에 가입합니다.

개요 권한 API 및 인증 API 사용자 인증 정보 동의 화면 푸시 모니터링 소스 코드 배포 및 관리 컴퓨팅 네트워킹 스토리지 빅데이터

API 사용 설정

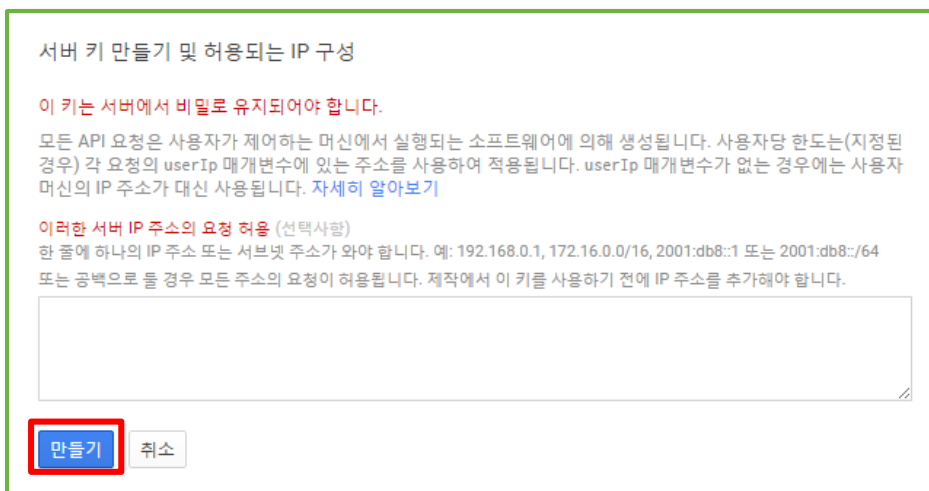
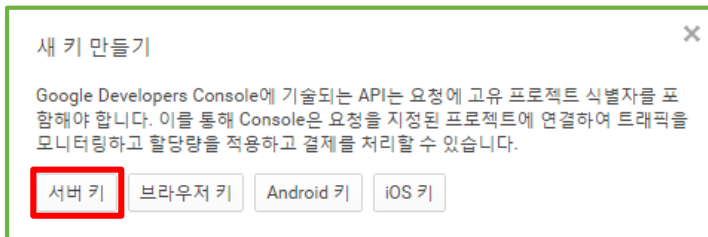
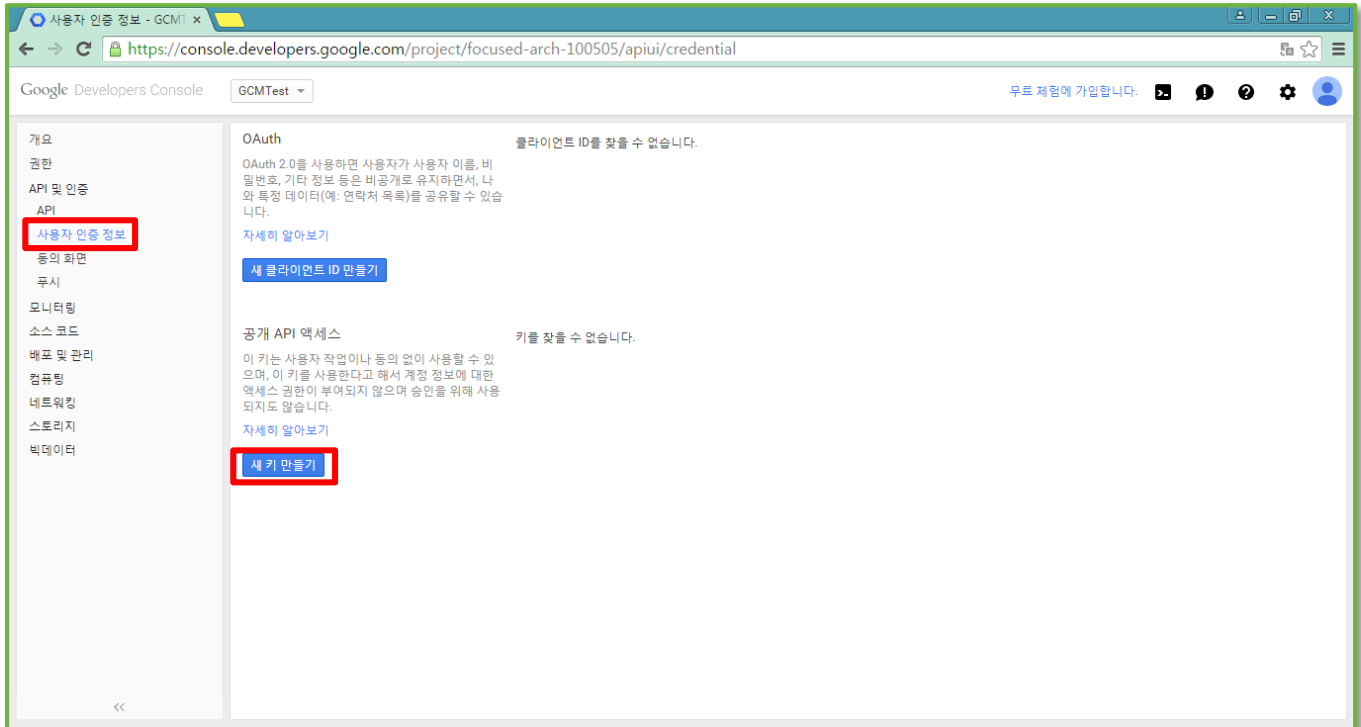
Google Cloud Messaging for Android

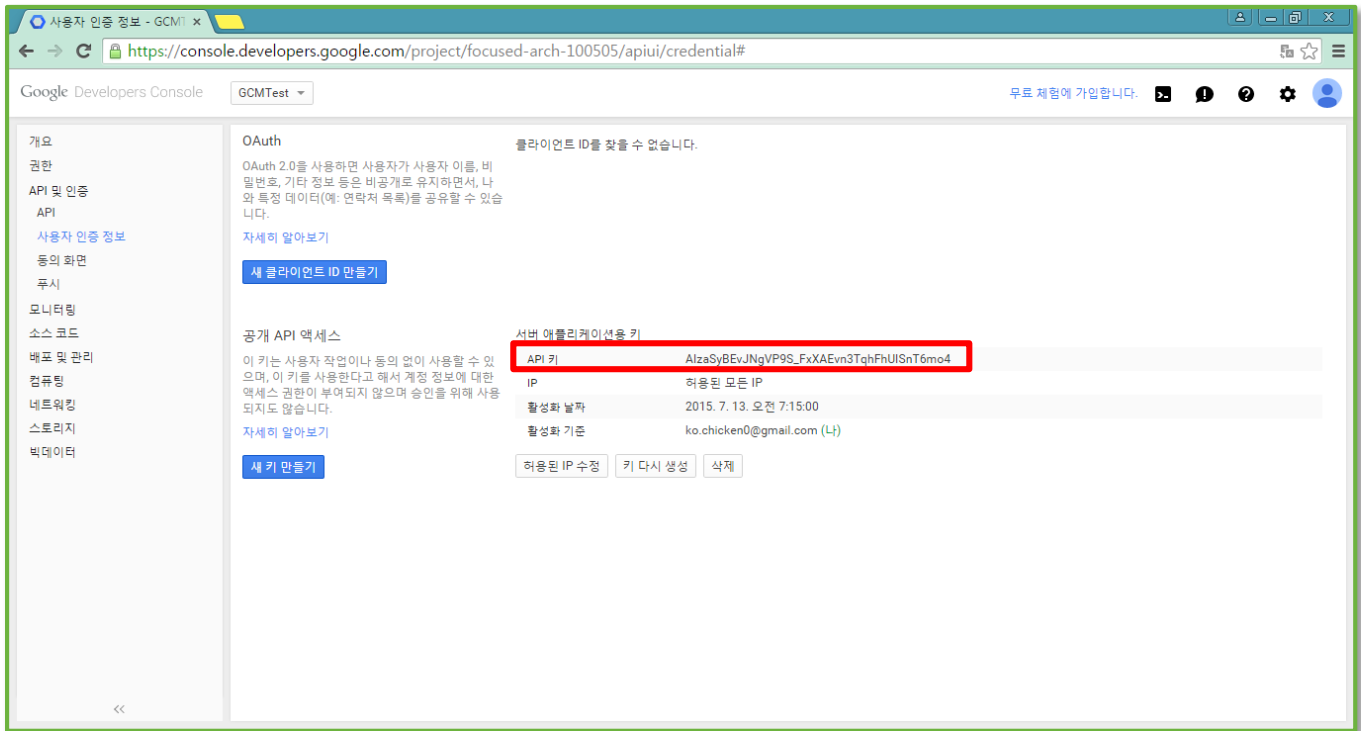
Google Cloud Messaging allows for push messaging to Android devices.

[자세히 알아보기](#)



### 2.1.3 Key 만들기





사용자 인증 정보 - GCM1 x

https://console.developers.google.com/project/focused-arch-100505/apiui/credential#

Google Developers Console GCMTest

무료 제형에 가입합니다.

개요 권한 API 및 인증 API 사용자 인증 정보 동의 화면 루시 모니터링 소스 코드 배포 및 관리 컴퓨팅 네트워킹 스토리지 빅데이터

OAuth 클라이언트 ID를 찾을 수 없습니다.

OAuth 2.0을 사용하면 사용자가 사용자 이름, 비밀번호, 기타 정보 등은 비공개로 유지하면서, 나와 특정 데이터(예: 연락처 목록)를 공유할 수 있습니다.

자세히 알아보기

새 클라이언트 ID 만들기

공개 API 액세스

이 키는 사용자 작업이나 동의 없이 사용할 수 있으며, 이 키를 사용한다고 해서 계정 정보에 대한 액세스 권한이 부여되지 않으며 승인을 위해 사용되지도 않습니다.

자세히 알아보기

새 키 만들기

서버 애플리케이션용 키

API 키	AlzaSy8EvJNgVP9S_FxXAEvn3TqhFhUISnT6mo4
IP	허용된 모든 IP
활성화 날짜	2015. 7. 13. 오전 7:15:00
활성화 기준	ko.chicken0@gmail.com (L)

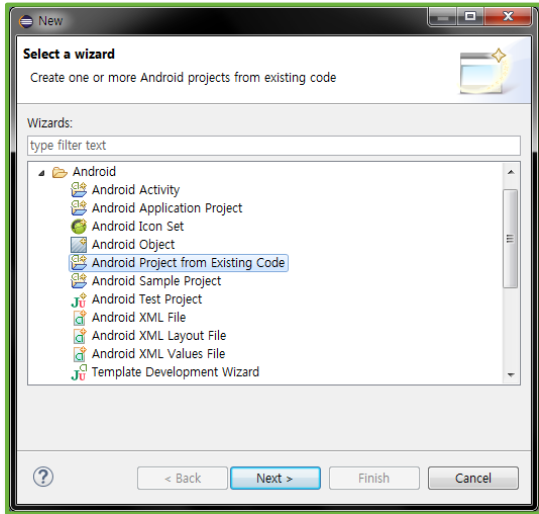
허용된 IP 수정 키 다시 생성 삭제

## API 키 메모

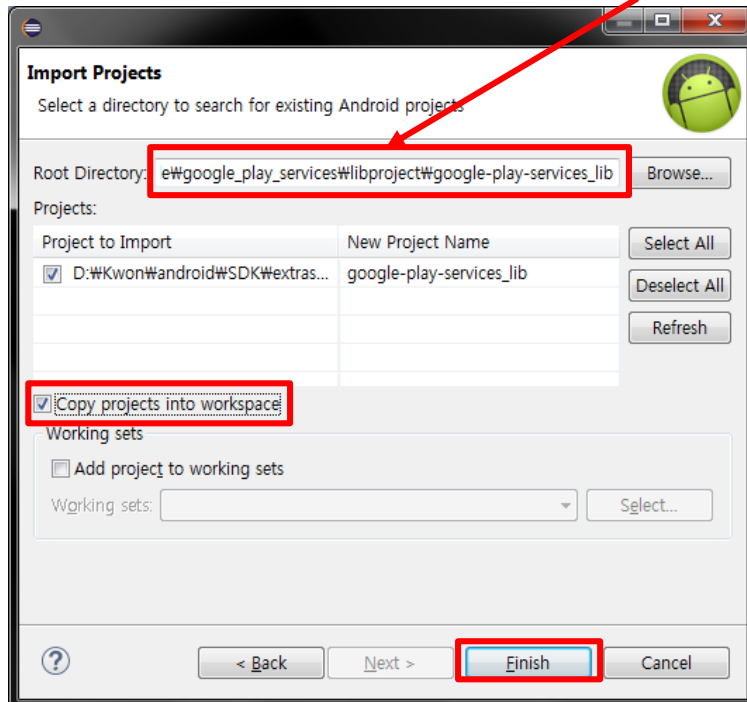


# 3. Android App

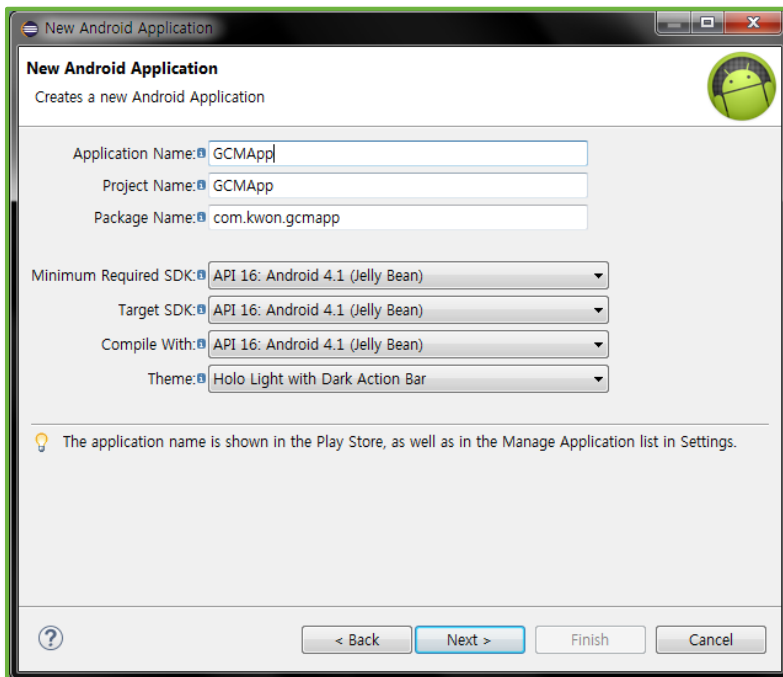
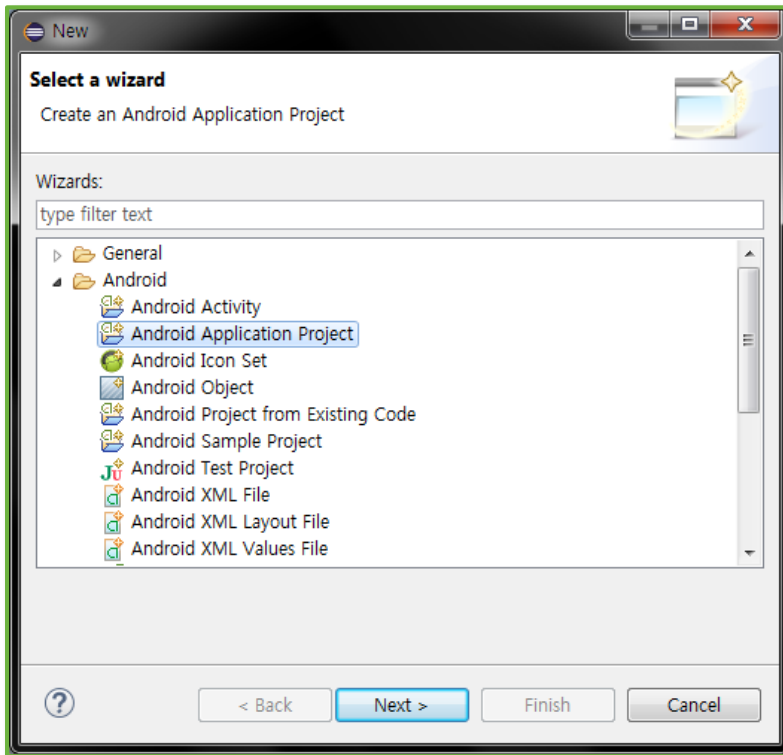
## 3.1 Google Play Service Library 추가



[Android SDK 위치]\extras\google\google\_play\_services\libproject\google-play-services\_lib

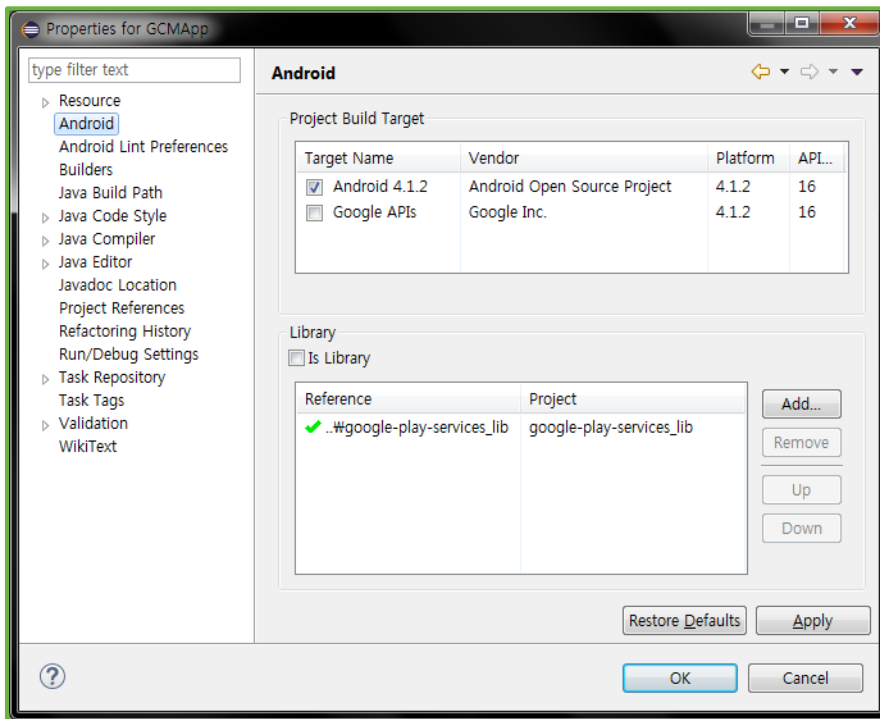
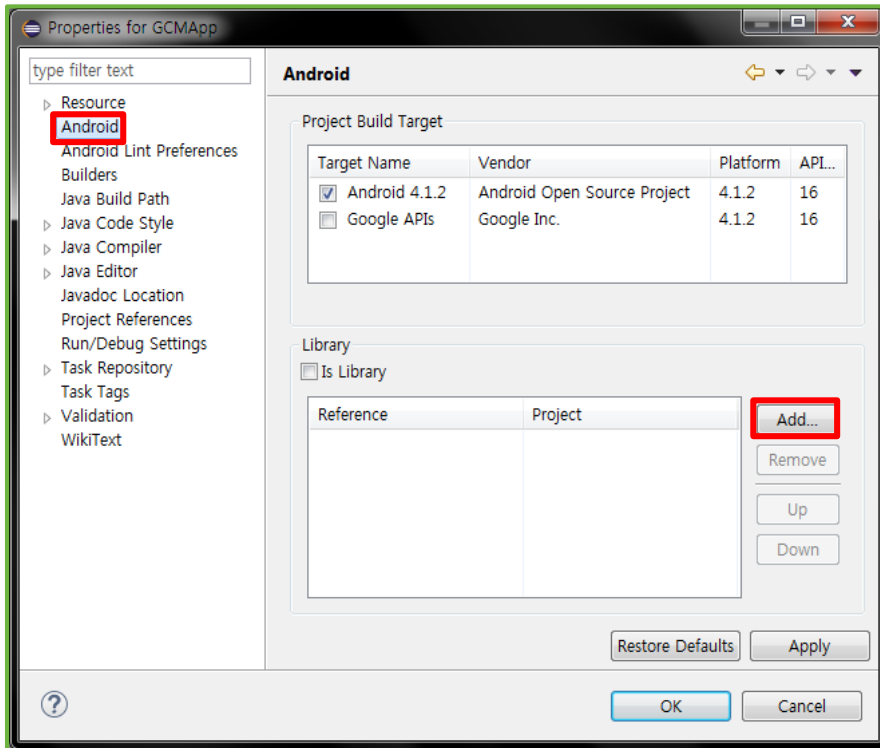


## 3.2 프로젝트 생성



### 3.3 프로젝트에 Google Play Service Library import

프로젝트 우클릭 > Properties



### 3.4 디자인

res/layout/layout\_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.kwon.gcmapapp.MainActivity"
    android:orientation="vertical">

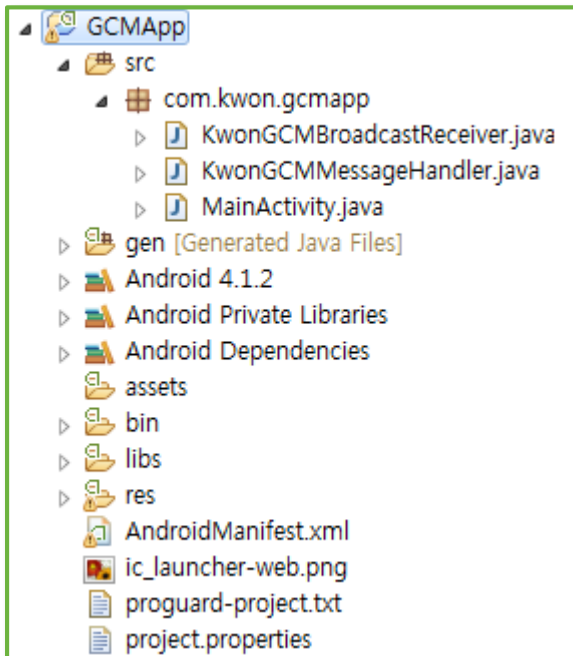
    <TextView
        android:id="@+id/tv"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="regId"/>

    <Button
        android:id="@+id/b"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="등록" />

</LinearLayout>
```



### 3.5 클래스 작성



#### 3.5.1 KwonGCMBroadcastReceiver.java

```
public class KwonGCMBroadcastReceiver extends WakefulBroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {

        // 클래스 가리킴
        ComponentName comp = new ComponentName(context.getPackageName(),
                                                KwonGCMMessageHandler.class.getName());

        // 서비스 시작
        startWakefulService(context, (intent.setComponent(comp)));
        setResultCode(Activity.RESULT_OK);
    }
}
```





### 3.5.2 KwonGCMMessageHandler.java

```
public class KwonGCMMessageHandler extends IntentService{
    private NotificationManager nm;
    private Handler h;

    public KwonGCMMessageHandler() {}

    public void onCreate() {}

    protected void onHandleIntent(Intent intent) {}
}
```

```
public KwonGCMMessageHandler() {
    super("에네르기파");
}
```

```
@Override
public void onCreate() {
    // TODO Auto-generated method stub
    super.onCreate();

    // 알림 관리자
    nm = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

    // 서버Thread 관리자
    h = new Handler();
}
```

```
@Override
protected void onHandleIntent(Intent intent) {
    // 서버에서 보낸 메시지의 형태 : {title:"제목", content:"내용"}
    String msgTitle = intent.getStringExtra("title");
    final String msgContent = intent.getStringExtra("content");
    // Activity를 실행시킬때 사용하는 객체
    PendingIntent pi = PendingIntent.getActivity(getApplicationContext(), 0,
        new Intent(getApplicationContext(), MainActivity.class), 0);

    // 알림
    Notification n = new Notification.Builder(getApplicationContext())
        .setVibrate(new long[] { 100, 200, 100, 500 }).setSmallIcon(R.drawable.ball)
        .setContentTitle(msgTitle).setContentText(msgContent)
        .setContentIntent(pi).setLights(Color.rgb(255, 255, 255), 500, 500)
        //.setSound(RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION))
        .setSound(Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.shoot)).build();

    // 알림 발사
    nm.notify(1, n);
    // 서버Thread -> 메인Thread로 변화
    h.post(new Runnable() {
        public void run() {
            // 메시지창
            Toast.makeText(getApplicationContext(), msgContent, Toast.LENGTH_LONG).show();
        }
    });
    // 설정 끝
    KwonGCMBroadcastReceiver.completeWakefulIntent(intent);
}
```



### 3.5.3 MainActivity.java

```
public class MainActivity extends Activity {  
    private String projectNo, regId;  
    private GoogleCloudMessaging gcm;  
  
    private TextView tv;  
    private Button b;  
    private Handler h;  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        private class RegTask extends AsyncTask<Void, Void, Void> {  
    }  
}
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    h = new Handler();  
    // 프로젝트 번호  
    projectNo = "989839899859";  
  
    tv = (TextView) findViewById(R.id.tv);  
    b = (Button) findViewById(R.id.b);  
    b.setOnClickListener(new OnClickListener() {  
  
        @Override  
        public void onClick(View v) {  
            // TODO Auto-generated method stub  
  
            // 백그라운드 작업 시작  
            new RegTask().execute();  
        }  
    });  
}
```



```
private class RegTask extends AsyncTask<Void, Void, Void> {
    @Override
    protected Void doInBackground(Void... params) {
        // TODO Auto-generated method stub
        gcm = GoogleCloudMessaging.getInstance(getApplicationContext());

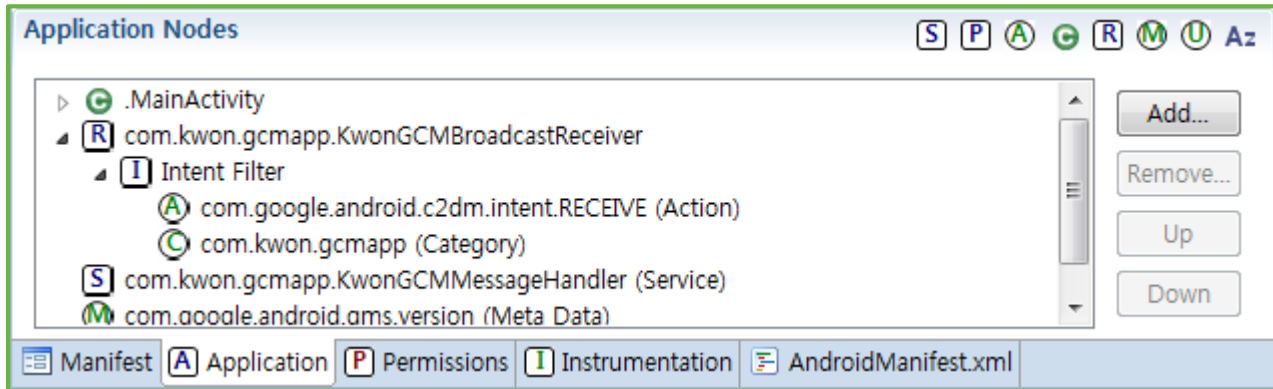
        try {
            regId = gcm.register(projectNo);
            // gcm.unregister();
            Log.i("regId", regId);
            h.post(new Runnable() {

                @Override
                public void run() {
                    // TODO Auto-generated method stub
                    tv.setText(regId);
                }
            });
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return null;
    }
}
```



## 3.6 AndroidManifest.xml

### 3.6.1 Application



```
<receiver
    android:name="[WakefulBroadcastReceiver클래스]"
    android:permission="com.google.android.c2dm.permission.SEND" >
    <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />

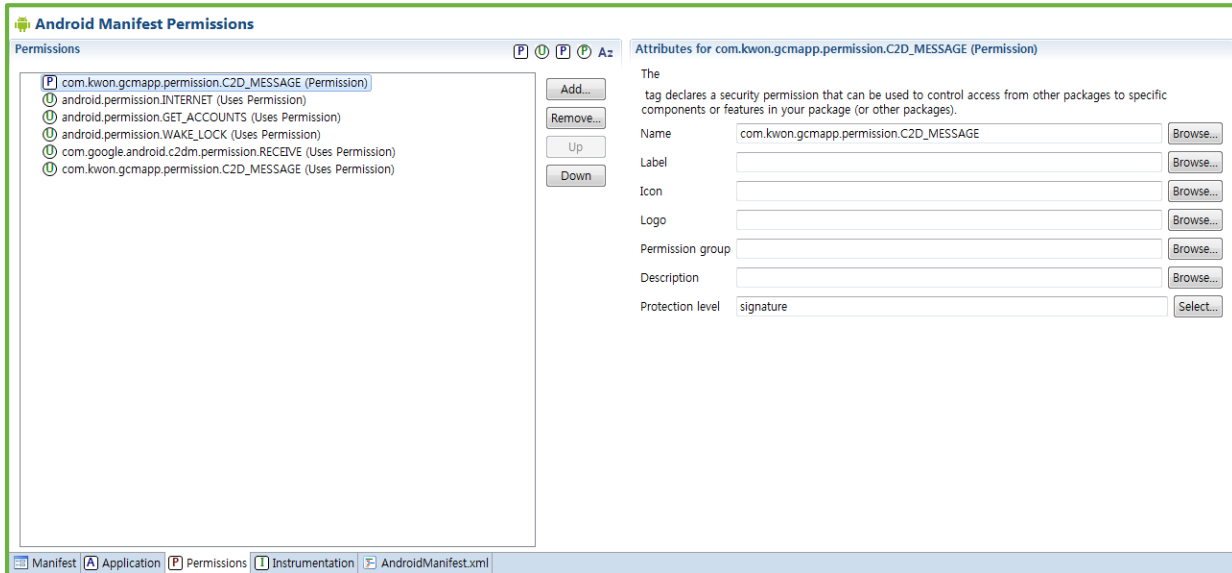
        <category android:name="[패키지명]" />
    </intent-filter>
</receiver>

<service android:name="[IntentService클래스]" />

<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```



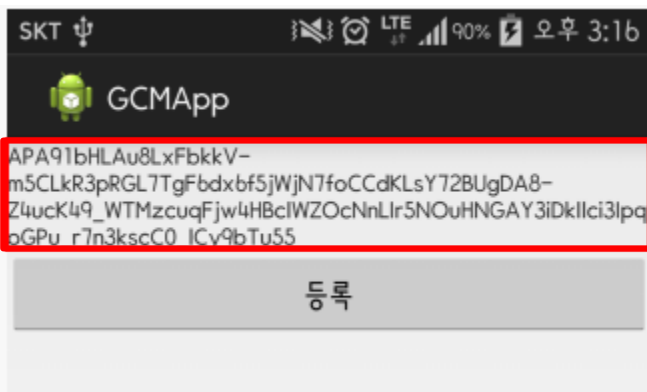
### 3.6.2 Permissions



```
<permission
    android:name="[패키지명].permission.C2D_MESSAGE"
    android:protectionLevel="signature" >
</permission>

<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="com.google.android.c2dm.permission.RECEIVE" />
<uses-permission android:name="[패키지명].permission.C2D_MESSAGE" />
```

### 3.7 실행

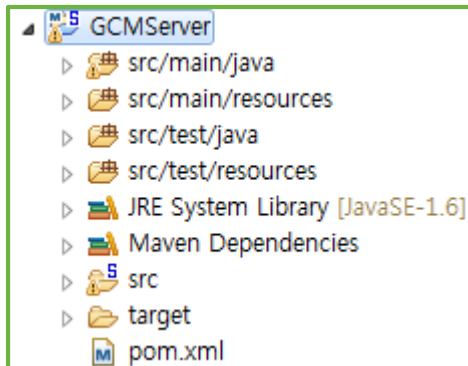


regid 메모




# 4.Spring Server

## 4.1 Spring MVC Project 작성



## 4.2 Maven 으로 gcm-server 추가

### Google GCM Library » 1.0.2



**Google GCM Library**

3 usages

[com.ganyo » gcm-server » 1.0.2](#)

POM created to enable maven dependency management for gcm-server

Artifact	<a href="#">Download ( JAR )</a> (21 KB)
POM File	<a href="#">View</a>
Date	(Apr 24, 2013)

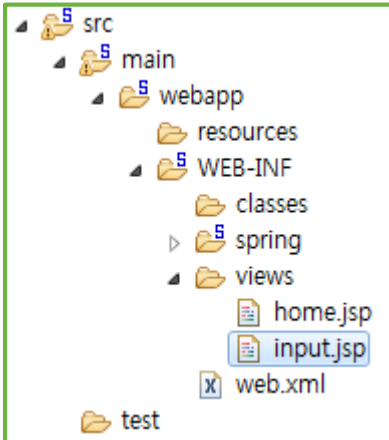
MavenIvyGrapeGradleBuildrSBTLeiningen

```
<dependency>
  <groupId>com.ganyo</groupId>
  <artifactId>gcm-server</artifactId>
  <version>1.0.2</version>
</dependency>
```



## 4.3 View

### 4.3.1 input.jsp 작성

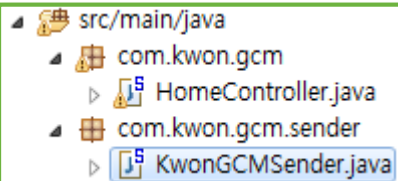


```
<body>
    <form action="GCM.send" method="post">
        제목 : <input name="title"><p>
        내용 : <input name="content"><p>
        <input type="submit" value="전송">
    </form>
</body>
```



## 4.4 Model

### 4.4.1 GCMSender 클래스 작성

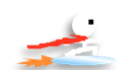
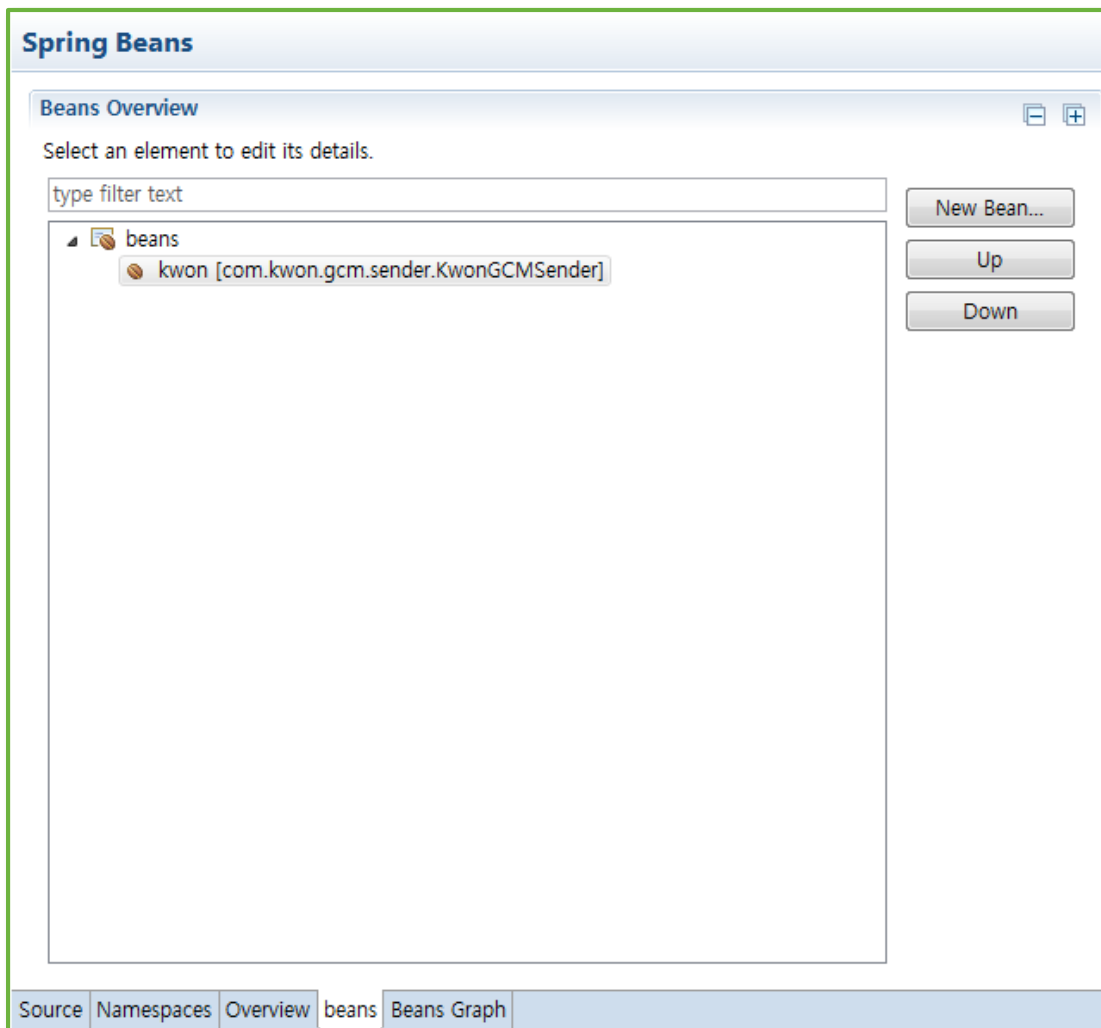
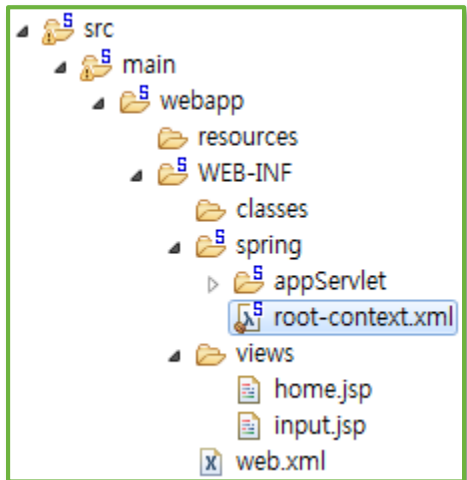


```
public class KwonGCMSender {  
  
    public void send(String title, String content) {  
        // Server API Key  
        Sender sender = new Sender("AIzaSyBEvJNgVP9S_FxXAEvn3TqhFhUISnT6mo4");  
  
        Builder builder = new Builder();  
        builder.addData("title", title);  
        builder.addData("content", content);  
  
        Message msg = builder.build();  
  
        try {  
            // 메세지, 안드로이드앱 regId, 재시도횟수  
            sender.send(msg,  
                "APA91bFSxMli7PX0ymu63sU03NV-fU_OP7qqkx5vKTMoV3QAGHRa6PQsV",  
                5);  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```





#### 4.4.2 root-context.xml 에 등록



## 4.5 Controller

### 4.5.1 HomeController 수정

```
@Controller
public class HomeController {

    @Autowired
    private KwonGCMSender kwon;

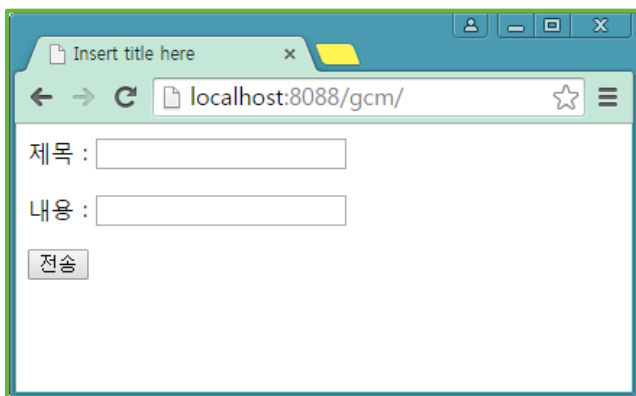
    @RequestMapping(value = "/GCM.send", method = RequestMethod.POST)
    public String send(
        @RequestParam(value = "title") String title,
        @RequestParam(value = "content") String content) {

        kwon.send(title, content);

        return "input";
    }

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home() {
        return "input";
    }
}
```

## 4.6 실행



Insert title here x

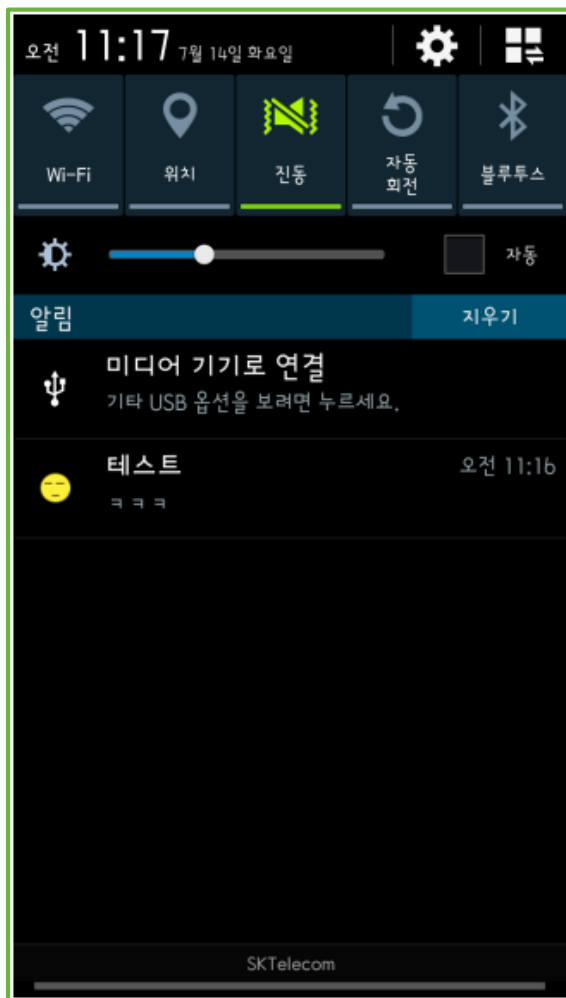
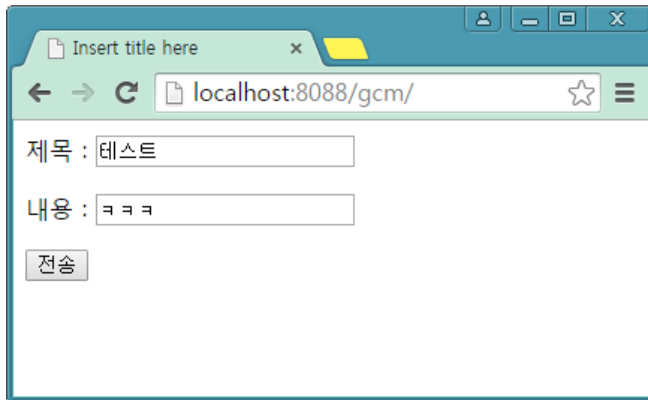
localhost:8088/gcm/

제목 :

내용 :



## 5. 실행결과



# 6. Notification 수정

## 6.1 KwonGCMMessageHandler.java Notification 부분 수정

```
Builder b = new Builder(getApplicationContext());

// 로컬 파일
// Bitmap bigPicture = BitmapFactory.decodeResource(getResources(), R.drawable.new_ball);

// 서버
Bitmap bigPicture
    = BitmapFactory.decodeStream(
        (InputStream)new URL("http://i.ytimg.com/vi/xAFOP73EzOI/maxresdefault.jpg").getContent());

BigPictureStyle bigStyle = new BigPictureStyle(b);
bigStyle.setBigContentTitle(msgTitle);
bigStyle.setSummaryText(msgContent);
bigStyle.bigPicture(bigPicture);

Notification n = b
    .setVibrate(new long[] { 100, 200, 100, 500 }).setSmallIcon(R.drawable.ball)
    .setContentTitle(msgTitle).setContentText(msgContent)
    .setContentIntent(pi).setLights(Color.rgb(255, 255, 255), 500, 500)
    .setSound(Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.shoot))
    .setTicker("여기 좀 봐봐").setStyle(bigStyle).build();
```



## 6.2 실행결과

