**출처:http://www.androidpub.com/2398318**

**1. Google Map Android V2?**

하나. SupportMapFragment :

일단 가장 중요한 것은 일반적인 View가 아닌 Fragment로 변경된 점인데요.

다양한 화면 구성을 위해 Fragment를 선보이면서 이에 맞춰 기존의 MapView(이하 V1)에서 SupportMapFragment(이하 V2)로 클래스가 변경이 되었습니다.

둘. APIKey 관리

그동안 V1을 사용 할 때는 각 PC에서 디버그용 Apikey 를 가지고 있어야 해서. PC를 옮겨서 작업할 때는 V1에 들어갈 Apikey를 계속 바꿔줘야 하는 번거로움이 있었습니다. 하지만 V2에선 API Console을 통해 키를 발급받아 좀더 효율적인 APIkey 관리를 할 수 있게 되어 다른 PC에서도 APIkey를 변경 하지 않고도 Google Map이 출력이 됩니다.

셋. Google APIs? Byebye~

기존의 V1을 사용하기 위해서는 Project build target을 Google APIs로 바꿔서 작업을 해야 했는데 V2를 사용 하기 위해서는 google-play-services\_lib 프로젝트를 불러와 사용해야 합니다. ( 저는 이점에 제일 번거롭다고 생각이 듭니다.. ㅠㅠ)

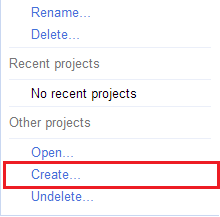
**2. 일단 띄워보자**

하나. 준비작업

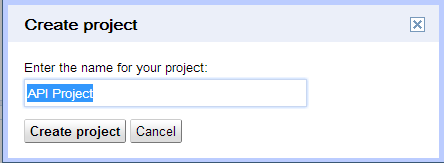
V2를 해보면서 제일 번거로웠던 점은 V1보다 번거로워진 준비 작업이 아닐까 싶은데요. 일단 스텝바이스텝으로 천천히 설명해보겠습니다.

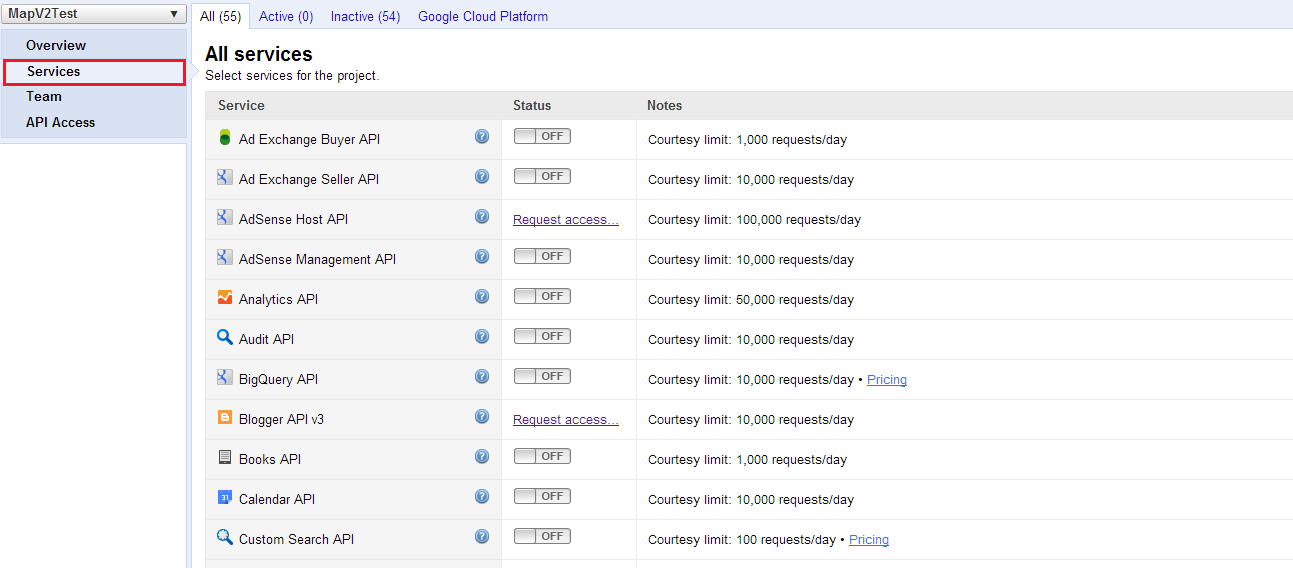
- API Console 등록

[**https://code.google.com/apis/console/**](https://code.google.com/apis/console/)API Console로 접속해 새로운 프로젝트를 생성 해야합니다.

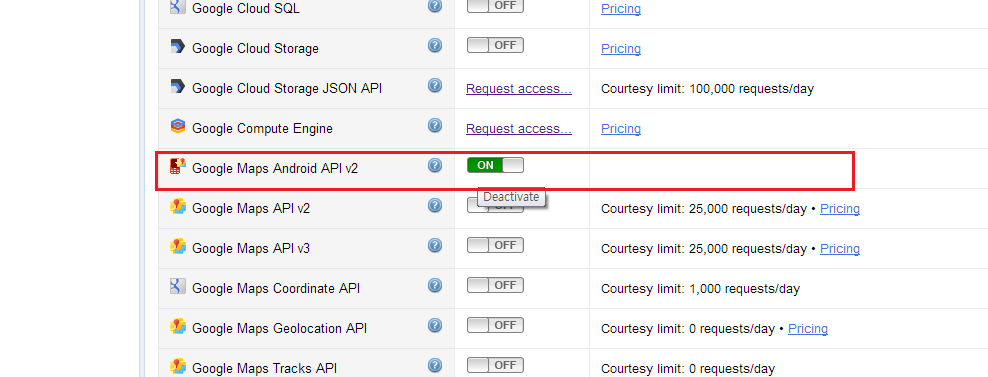


첫번째로 프로젝트 생성을 해줍니다. 이정도는 뭐 껌이죠 ~\_~



프로젝트 이름을 정해줍니다. 저는 MapV2Test라고 했네요.

프로젝트를 생성하게 되면 바로 Services 탭으로 넘어가게 됩니다.



Google Map Android V2라는 항목이 있습니다. 이걸 Off -> On으로 돌려준 뒤 API Access탭으로 넘어갑니다.



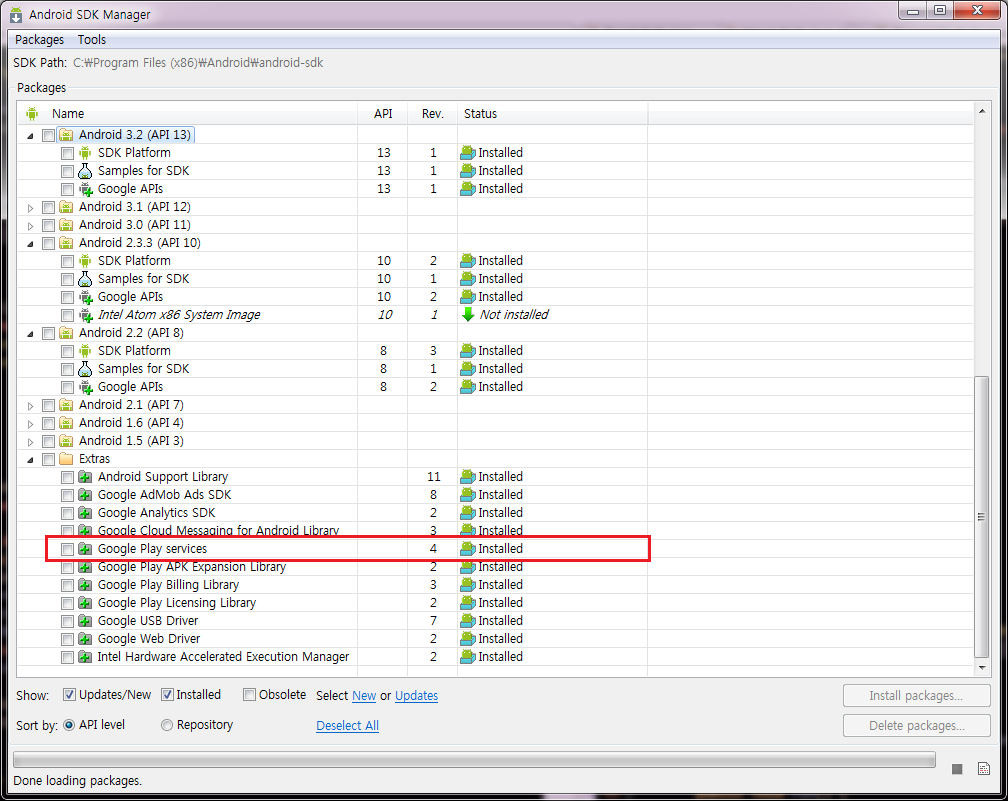
API Access탭으로 넘어 가면 바로 API Key가 보이게 되는데 이 키가 V1에서 사용되던 APIKey처럼 V2에서도 같은 역할을 해주는 키값이 되겠습니다.

둘. 맵 띄워 보기

- SDK 업데이트

일단 맵을 띄우기 위해선 SDK를 최신버전으로 업데이트 해야 합니다. ( 필수 )

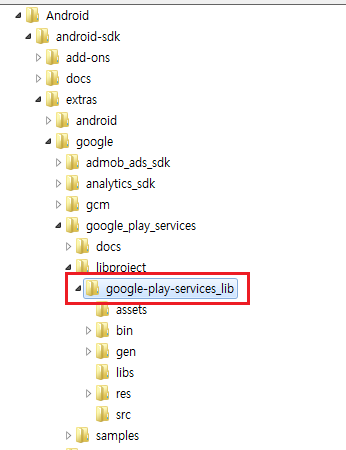
밑에 스크린샷에 보이는 Google PlayService 를 최신버전으로 업데이트 시켜줘야 합니다.



- import library project

맵을 호출 하기 위해서는 SDK폴더에 있는 라이브러리 프로젝트가 필요 합니다.

SDK 폴더 안에 있는 **sdk/extras/google/google\_play\_services/libproject/google-play-services\_lib** 를 import 해줍니다.



- 맵뷰를 띄울 새 프로젝트 만들기

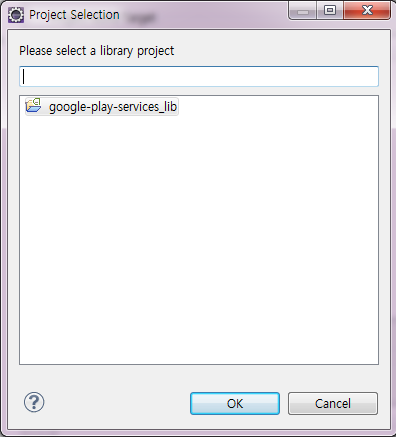
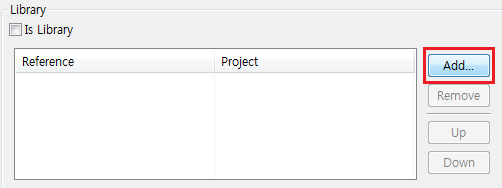
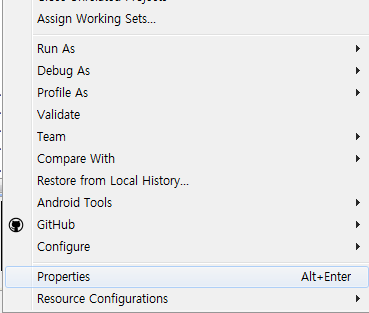
일단 샘플링을 하기 위해서는 빈 프로젝트가 필요 하겠죠? 새 안드로이드 프로젝트를 만들어 줍니다.

이때 주의할 사항은 **Target SDK를 Google APIs로 맞출 필요가 없다는 것** 입니다.

원하시는 SDK을 잡으시고 빈 프로젝트를 만들어 줍니다. ( 스크린샷을 안떠서 바로 넘어 가겠습니다 ' ㅅ';;)

- 라이브러리 불러오기

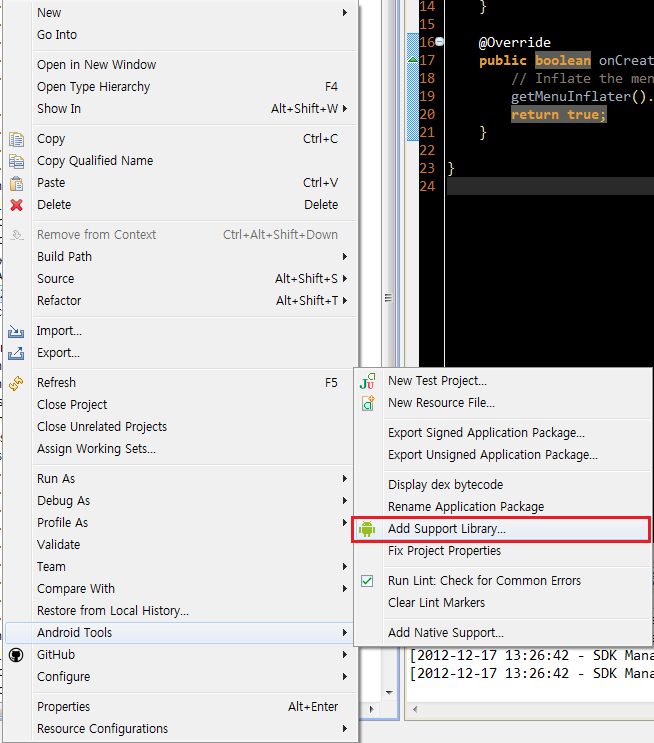
새로 만든 프로젝트에 import 해온 **google-play-services\_lib를 추가**해 줍니다.



참 쉽죠?

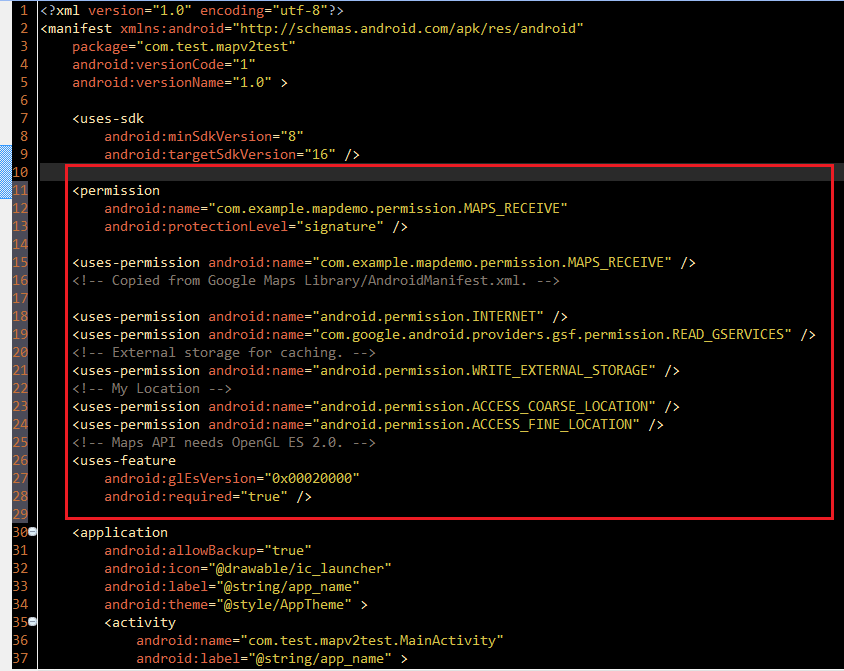
- support-v4 라이브러리 추가

SupportMapFragment를 넣기 위해서는 서포트 라이브러리가 꼭!!!! 필요 합니다.



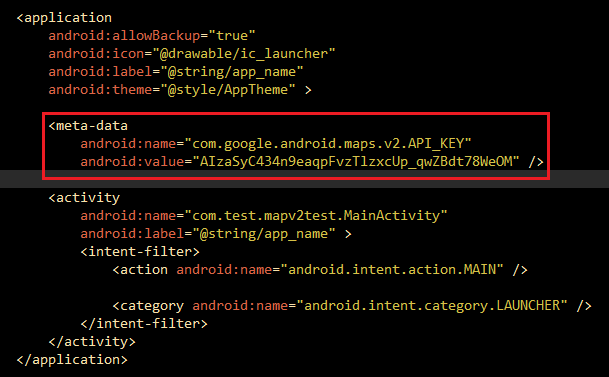
-Manifest 설정

매니패스트 설정 부분입니다. 첫번째 스크린샷은 application 밖에서 설정 해주는 부분입니다.



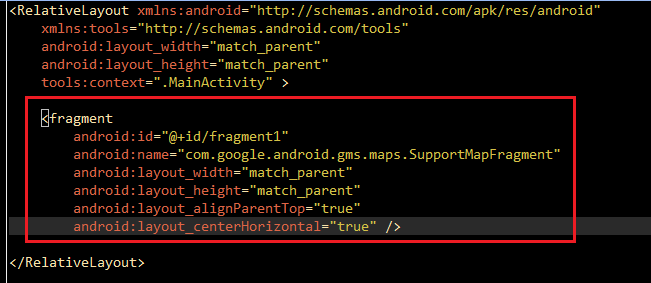
meta-data 설정을 하는 부분입니다.

해당 엘리먼트는 **API Console 에서 발급받은 API key를 입력**을 해야 올바른 맵을 볼 수 있으니 꼭 넣어 주셔야 합니다 ( 제 테스트 키 값은 이러네요.. )



- xml : SupportMapFragment 추가

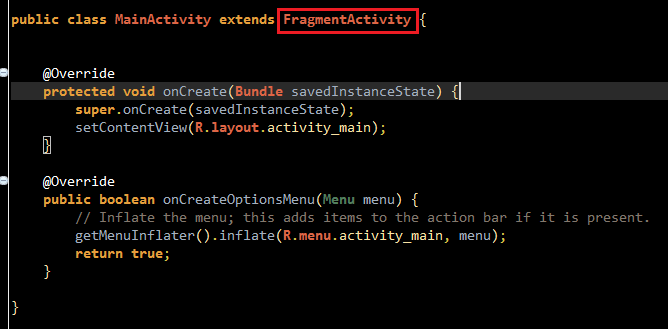
위에서 말씀드렸듯이 V1에서는 View형태로 사용 했지만 V2에서는 Fragment 형태로 변경이 되었습니다.



- Activity - > FragmentActivity로 변경

SupportMapFragment는 서포트 라이브러리의 FragmentActivity에서 동작을 합니다.

MainActivity가 상속한 Activity를 Fragment Activity 로 바꿔줍니다.



- Run as Application

일단 실행 시켜보겠습니다.



**3. 그밖의 기능**

**﻿**

﻿하나. 마커

V2 샘플을 만들면서 가장 편했던 기능은 V1의 Overlays 관리, V2에서는 Market 관리인것 같습니다.

V1에서는 Overlays를 MapController에 넣어 줘야 하는 번거로움이 있었지만 V2에서는 MapController와 같은 Controller 클래스는 찾아 볼수가 없었다는 점인데요.

아래 소스 캡쳐 화면을 보시면 좀더 간결하게 진행 할 수 있다는것을 보실 수 있습니다.



- 위에서 설명이 안됐지만 MapFragment를 선언 해주는 부분과 마커를 추가 하는 소스입니다.

<1> 은 **V1의 MapView** 클래스에서 **V2의 GoogleMap** 클래스로 클래스명이 변경되었습니다. 좀더 직관적으로 바꼈네요.

<2>은 MapFragment를 불러오는 부분입니다. Fragment를 불러오는 형식이 비슷해 보이네요

<3>은 MapFragment에 마커를 추가 하는 부분인데요. **MarketOptions**(추후 설명 업데이트)  클래스와 **LatLng**(Double타입, Double타입) 클래스가 눈에 띕니다.

둘. animate & move

마커를 찍어봤으니 이제 카메라를 이동 시켜봐야겠죠.

일단 제일 간단한 **animateCamera**와 **moveCamera**를 소개해 드리겠습니다.

- animateCamera

**animateCamera** 함수는 이전에 V1에서 사용 했던 **animateTo** 함수와 같은 기능을 합니다.

[view source](http://www.androidpub.com/2398318#viewSource)



[print](http://www.androidpub.com/2398318#printSource)[?](http://www.androidpub.com/2398318#about)

1.void setAnimate(){

2.    googleMap.animateCamera(CameraUpdateFactory.newLatLng(new LatLng(37, 123)));

3.}

V2에서 바뀐점은 **CameraUpdate**라는 클래스를 사용하는 것인데 해당 클래스를 사용 할 때는 **CameraUpdateFactory**를 가져와 사용 합니다.  여기서도 중요한것은 LatLng 클래스를 사용 합니다.

- moveCamera

**moveCamera** 함수는 animateCamera처럼 **애니메이션이 적용 되지 않고**, 바로바로 맵이 이동되는 함수 입니다.

[view source](http://www.androidpub.com/2398318#viewSource)



[print](http://www.androidpub.com/2398318#printSource)[?](http://www.androidpub.com/2398318#about)

1.void setMove(){

2.    googleMap.moveCamera(CameraUpdateFactory.newLatLng(new LatLng(37, 123)));

3.}