

Daum Map API

8조

조원 : 강종찬, 정성수, 김단비, 류승렬

<목차>

- Daum map API
- Daum mobile map API
- 다른 map API와의 차이점
- 사용법
 - 사용 준비
 - API키 받기
 - 라이브러리 다운 및 추가하기
 - 예제
 - 지도를 띄우는 두가지 방법
 - 지도 띄우기(WebView)
 - 지도 띄우기(jar)
 - 자기 위치 찾기
- 관련 메서드
- 참고문헌

다음 지도 API란?

- Daum에서 누구나 사용할 수 있도록 공개한 지도 API
- 버전 : v3사용
 - Mobile 환경에서도 지도를 이용할 수 있도록 경량화
 - 사용자가 자유롭게 기능을 확장하여 이용할 수 있도록 설계

다음 지도 API란?

- Map Type



<일반지도>



<위성 뷰>



<로드 뷰>

다음 모바일 지도 API란?

- Native 앱 개발시 Daum 지도를 추가
- 웹 용 지도API와 달리 Native 수준에서 동작
- 앱과 더욱 긴밀하게 동작
- 지원 사양
 - IOS 4.0 이상
 - 안드로이드 2.1(API level7) 이상

다른 지도 API와의 차이점

다음	네이버	구글
한국형 지역정보 제공	실시간 교통정보 풍부	정보량이 매우 많음
Street view가 제공	작은 길에 대한 정보 제공	대부분의 지도 앱
자세한 지도가 제공	Pinch 동작(줌인/아웃) 안됨	지도확대 매우 어려움

사용법 - 사이트 접속

<http://dna.daum.net/apis/dashboard> 에 접속

오른 API 모두 보기 : Daum DNA - Windows Internet Explorer
http://dna.daum.net/apis/dashboard

모두 보기

- 마이피플 API**
다음 대표 모바일 메신저 마이피플의 봇 API 및 메시지 API를 통해 가상 봇 친구 운영 및 메시지 전송 등 다양한 기능을 구현 할 수 있습니다.
[시작하기](#) | [API종류](#) | [활용하기](#) | [메시지 보내기](#)
- 프로필 API**
다음 사용자의 공개 닉네임, 프로필 이미지를 통해 외부 사이트에서 소셜 로그인 및 신원 제공 서비스를 활용할 수 있는 API입니다.
[시작하기](#) | [API종류](#) | [활용하기](#)
- 검색 API**
다음 검색 내 카페, 블로그, 게시판, 동영상, 웹, 이미지, 지식, 책 서비스에 대한 검색 결과를 전달하는 API입니다.
[시작하기](#) | [API종류](#) | [활용하기](#)
- 소셜픽 API**
검색수, 조회수, 댓글수, 추천수, 트위터 리트윗 등 사용자 반응을 분석하여 중요한 이슈만을 선별하여 제공하는 소셜픽 API입니다.
[참조하기](#)
- 쇼핑 API**
Daum 쇼핑에 올라 오는 상품 검색 및 상품 정보를 블로그나 웹 서비스에 활용할 수 있습니다.
[시작하기](#) | [API종류](#) | [활용하기](#)
- 콘텐츠 API**
Daum 주요 콘텐츠인 영화 데이터 API 입니다. 향후 음악, 건강 등 추가 예정입니다.
[시작하기](#) | [API종류](#) | [활용하기](#)
- 지도뷰 API**
Daum의 대표적인 지도뷰, 스카이뷰, 로드뷰, 스토어뷰 등 다양한 지도 보기 기능을 제공합니다.
[시작하기](#) | [API종류](#) | [활용하기](#)
- 로컬 API**
다음 지도를 활용하기 위한 주소/좌표변환, 좌표계 변환, 장소 검색 등 로컬 데이터 API입니다.
[시작하기](#) | [API종류](#) | [활용하기](#)
- 모바일앱용 지도뷰 API**
모바일앱용 지도API는 iOS, Android 앱 개발시 Daum 지도를 추가할 수 있는 라이브러리를 제공합니다.
[시작하기](#) | [Android](#) | [iOS](#)
- 모바일 URLScheme API**
다음 앱, 지도 앱, 마이피플 앱 등 모바일 앱 기능을 앱으로 연결할 수 있는 URL Scheme API입니다.
[시작하기](#) | [앱 전체 보기](#) | [참고자료](#)

원로 | 인터넷 | 보호 모드: 설정 | 100%

사용법 - 라이브러리 다운

- 안드로이드 용 라이브러리 다운로드

The screenshot shows a Windows Internet Explorer browser window with the address bar displaying `http://dna.daum.net/apis/mmaps#toc-library-android`. The page content is as follows:

- Left Sidebar:**
 - 다이나믹 API
 - 검색 API
 - 로컬 API
 - 쇼핑 API
 - 콘텐츠 API
 - 소셜픽 API
 - 이용약관
 - 인증형 API
 - OAuth 살펴보기
 - 마이피플 API
 - 카페 API
 - 블로그 API
 - 캘린더 API
 - 프로필 API
 - 이용약관
 - 지도형 API
 - 지도뷰
 - 모바일앱용 지도뷰
 - 이용약관
 - API 사용 참조
 - URL Scheme
 - 표준 에러 메시지
 - 종료 API 목록
 - 로고 사용 가이드
 - FAQ
- Main Content:**
 - 모바일앱용 지도API(iOS) 시작하기 ios
 - 참조하기
 - 인쇄 | PDF | 키발급
 - SDK** 아이폰 및 안드로이드용 지도 SDK (라이브러리, 샘플코드 등) ios Android
 - SDK 다운로드**
 - 모바일용 각종 라이브러리, 샘플코드, 개발 문서를 포함한 SDK입니다.
 - Buttons: iOS **Android** (circled in red)
 - Text: 이전 버전을 다운로드 받고 싶으시나요? [여기를](#) 클릭하세요.
 - Button: **SDK 다운로드**
 - 활용하기
 - URL Scheme
 - 명예의 전당

The browser's status bar at the bottom shows "인터넷 | 보호 모드: 설정" and "100%".

사용법 - 안드로이드용 선택

- 모바일 앱용 지도 API (Android) 시작하기 클릭

The screenshot shows a web browser window displaying the Daum DNA developer network page for the Mobile Maps API. The page title is "모바일앱용 지도 API : Daum DNA". The main heading is "DNA 개발자네트워크". The navigation menu includes "나의 API", "오픈 API", "제휴 API", "개발 도구", "기술 공유", "개발자 카페", "공식 블로그", "Lens", "Labs", "위젯뱅크", and "다음앱". The main content area is titled "모바일앱용 지도 API" and includes an introduction and a list of links. The link "모바일앱용 지도 API(Android) 시작하기" is circled in red. The footer includes "SDK 다운로드" and "SDK" information.

모바일앱용 지도 API : Daum DNA

10 류진 +67

전쟁종독 | 메일 99+ | 카페 | 뉴스 | 쇼핑 더보기

DNA 개발자네트워크

나의 API | **오픈 API** | 제휴 API | 개발 도구 | 기술 공유 | 개발자 카페 | 공식 블로그 | Lens | Labs | 위젯뱅크 | 다음앱

오픈 API 소개 및 이용사례
개발성 소개

모두 보기

데이터형 API
- 검색 API
- 로컬 API
- 쇼핑 API
- 콘텐츠 API
- 소셜픽 API
- 이용약관

인증형 API
- OAuth 살펴보기
- 마이피플 API
- 카페 API
- 블로그 API

모바일앱용 지도 API

부러제라 | 1일 5만 제한 | 제휴가능
apkey | JavaScript

안내

모바일앱용 지도API는 iOS, Android 앱 개발시 Daum지도를 추가할 수 있는 라이브러리를 제공합니다. 웹용 지도API보다 Native 앱과 좀 더 긴밀하게 동작이 가능합니다.

- 모바일앱용 지도 API 시작하기 **Mobile**
- 모바일앱용 지도API(Android) 시작하기 **Android**
- 모바일앱용 지도API(iOS) 시작하기 **iOS**

참조하기

인쇄 | PDF | 개발금

SDK | 아이폰 및 안드로이드용 지도 SDK (라이브러리, 샘플코드 등) **iOS** **Android**

SDK 다운로드

본 페이지를 가장 편리하게 방문하신다면, 샘플코드 개발 문서를 포함한 SDK를 다운받으실 수 있습니다.

인터넷 | 보호 모드: 설정 | 100%

사용법 - 오픈 API 키 발급

- 오픈API 키 발급 페이지 클릭

모바일앱용 지도API(Android) 시작하기 : Daum DNA - Windows Internet Explorer

http://dna.daum.net/apis/mmaps/android

즐거찾기 추천 사이트 추가 기능 더 보기

모바일앱용 지도API(Android) 시작하기 : Daum ...

- 표준 에러 메시지
- 종료 API 목록
- 로고 사용 가이드
- FAQ

API 놀이터 OAuth 놀이터

월간 스타트업 토크쇼 2014 150+명의 창업자 아이디어의 사업화를 지원합니다. 2014. 3. 31 ~ 4. 25

APIKey와 Package Name

이 API를 사용하기 위해서는 apikey를 발급받아야 합니다. apikey는 **오픈API 키 발급 페이지**에서 발급 받으실 수 있습니다. apikey는 package name 당 하나가 발급됩니다. Android 앱의 package name은 AndroidManifest.xml 파일 내 manifest 엘리먼트의 package 속성에 설정됩니다.

Library 파일 추가

1.2.0 이전 버전의 경우

Android-Daum-Map 라이브러리는 libDaumMapAndroid.jar와 libMapEngine.so 두개의 바이너리로 구성되어 있습니다. libDaumMapAndroid.jar 파일을 애플리케이션 프로젝트의 classpath로 설정하고 libMapEngine.so파일을 */libs/armeabi/*로 복사해야 합니다.

1.2.X 버전인 경우

Android Daum Map 라이브러리는 libDaumMapAndroid.jar와 아키텍처별로 libMapEngineApi.so 두개의 바이너리로 구성되어 있습니다. 주어진 압축파일 libs 에 있는 파일들을 해당 프로젝트를 */libs/*로 모두 복사해야 합니다.

- ▼ DaumMapLibrarySample_Android (trunk/MobileMap/MapAndroid/)
 - ▶ src
 - ▶ gen [Generated Java Files]
 - ▶ Android 2.2
 - ▶ Referenced Libraries
 - ▶ .settings
 - ▶ assets
 - ▶ bin
 - ▼ lib
 - libDaumMapAndroid.jar 22117 1/27/12 8:21 PM bwlim
 - ▼ libs
 - armeabi
 - libMapEngineApi.so 21771 1/11/12 1:59 PM bwlim

인터넷 | 보호 모드: 설정 100%

사용법 - 오픈 API발급페이지

지도형 API 신청 : Daum DNA - Windows Internet Explorer

주소: <https://dna.daum.net/myapi/mapsapi/new>

지도형 API 신청 : Daum DNA

지도형 API 신청

지도형 API 신규등록

- 지도형 API는 Daum의 지역 정보를 제공하며, 지도뷰 등이 제공됩니다.
- 지도형 API는 지도뷰와 모바일용 지도뷰 각각 5개의 API키를 등록할 수 있으며, 신청한 웹 사이트에 이용 결과가 외부로 공개되어야 합니다.

API 선택 지도뷰 모바일앱용 지도뷰

사용 웹사이트

- 사용할 웹 서비스 URL을 기입해 주세요. 정확하지 않으면 이용이 정지될 수 있습니다.
- 지도 API는 "레퍼러(referer) 확인용"으로 활용됩니다.(Prefix 체크)
- 웹이 아닌 데스크탑/모바일 앱 등의 경우, "앱 소개 페이지 URL"을 기입해주세요.

동작 환경

- 지도 API를 제공할 운영 체제를 선택하십시오.

모바일앱 ID

- iOS의 경우, "Buddle ID"를 의미합니다.
- Android의 경우, "Package Name"을 의미합니다.

지도 API 이용 약관

Package Name 입력

사용법 - API키 발급완료

지도형 API 관리 : Daum DNA - Windows Internet Explorer

주소: <https://dna.daum.net/myapi/mapsapi>

즐거찾기 | 추천 사이트 | 추가 기능 더 보기

지도형 API 관리 : Daum DNA

데이터형 API

- 신규 등록
- 관리 및 통계

지도형 API

- 신규 등록
- 관리 및 통계

인증형 API

- 신규 등록
- 앱 관리

공공형 API

- 신규 등록
- 관리 및 통계

제휴 센터

- 제휴 신청
- 상담 신청
- 신청 내역 및 관리

갤러리 등록

총 2개 API API 키 등록

서비스	정보	통계	관리
 모바일 지도 API	사용URL : https://dna.daum.net/myapi/mapsapi/new 발급일 : 2014.05.11 apikey : 612a367127335365215b78b01ff662295e6163f2	제한 횟수 : 50,000 오늘 요청 : 0 총 요청 : 0	등록
 모바일 지도 API	사용URL : https://dna.daum.net/myapi/mapsapi/new 발급일 : 2014.05.07 apikey : 3709130db744bdcbfdfa2dd98b0f83be03159e4e	제한 횟수 : 50,000 오늘 요청 : 0 총 요청 : 0	등록

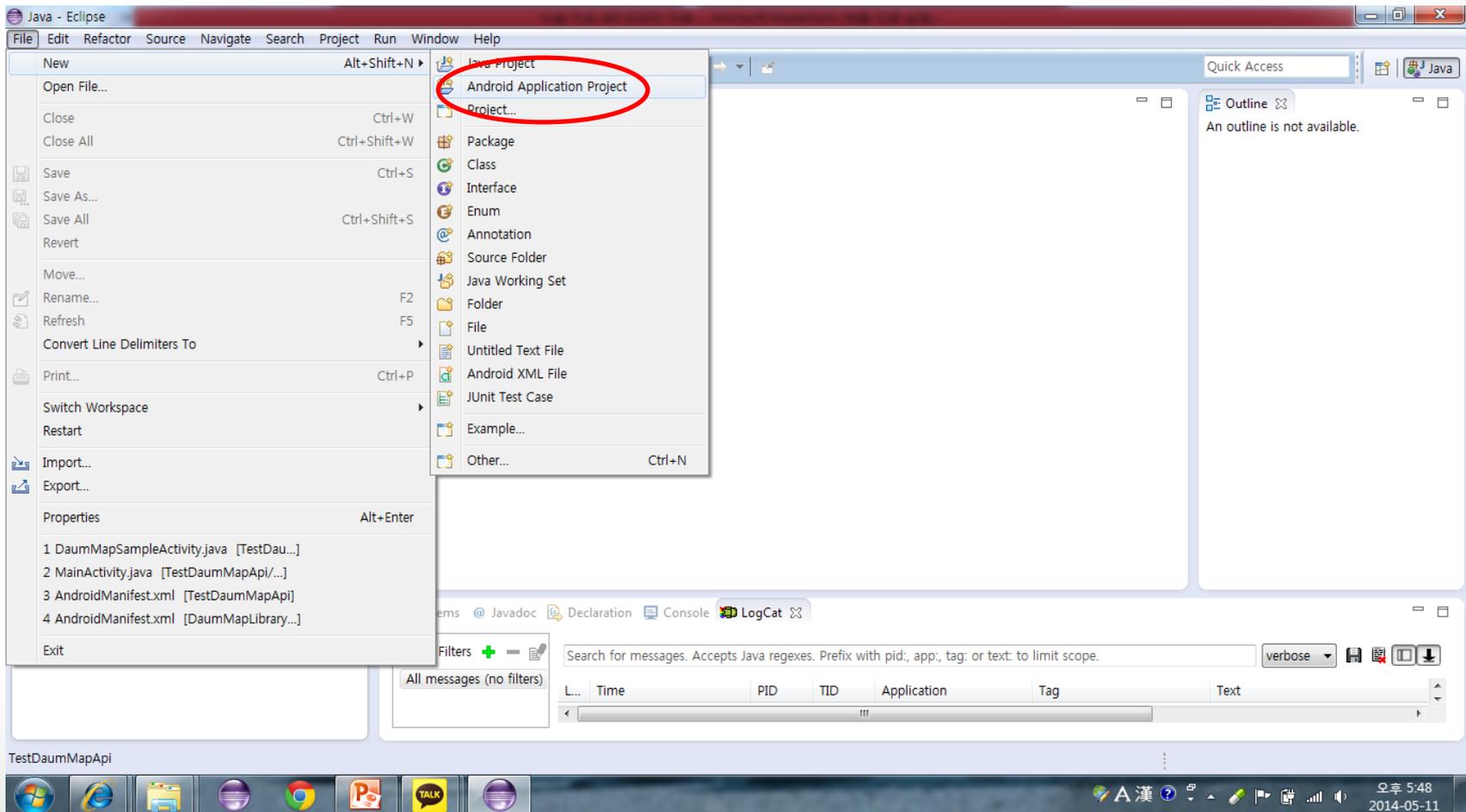
< 이전 1 다음 >

 API 놀이터

 OAuth 놀이터

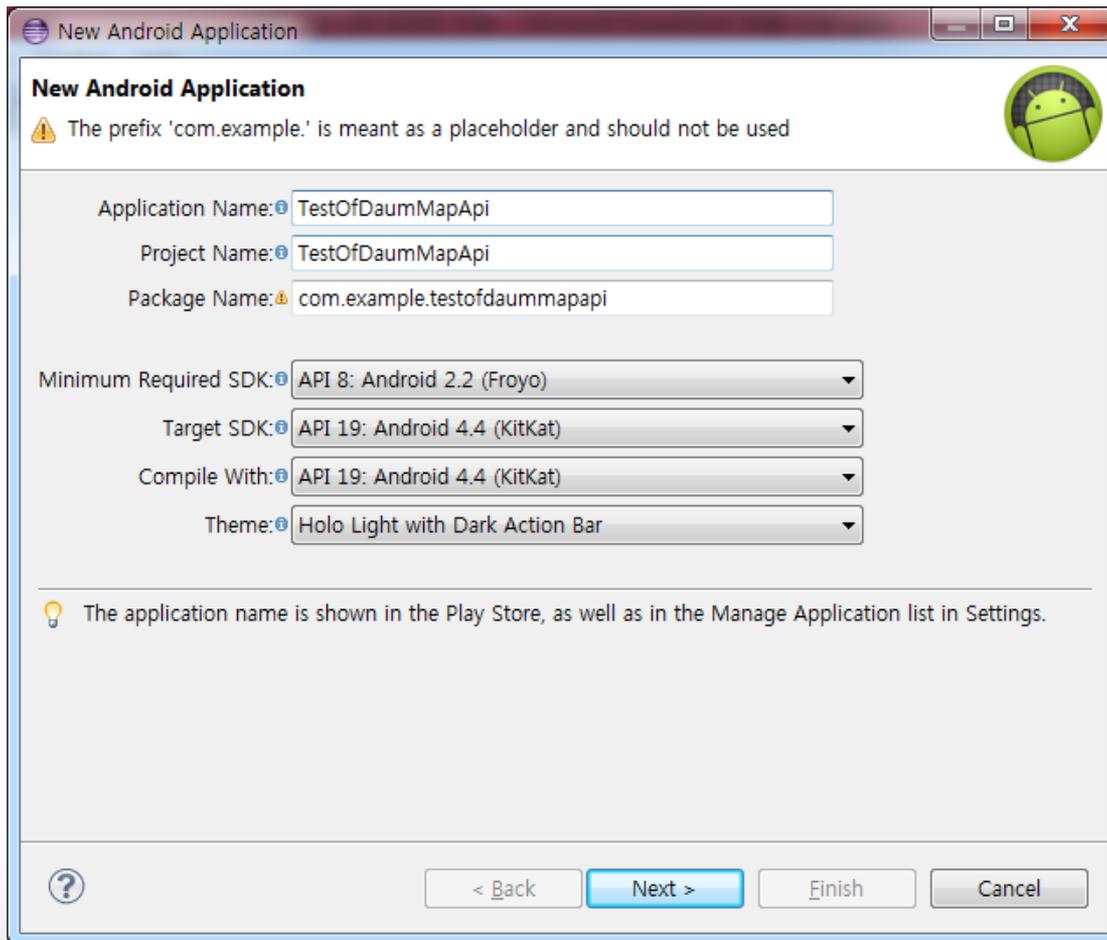
사용법 - 이클립스 실행

- 안드로이드 어플리케이션 프로젝트 생성



사용법 - 프로젝트 만들기

- 프로젝트 생성



New Android Application

New Android Application

⚠ The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

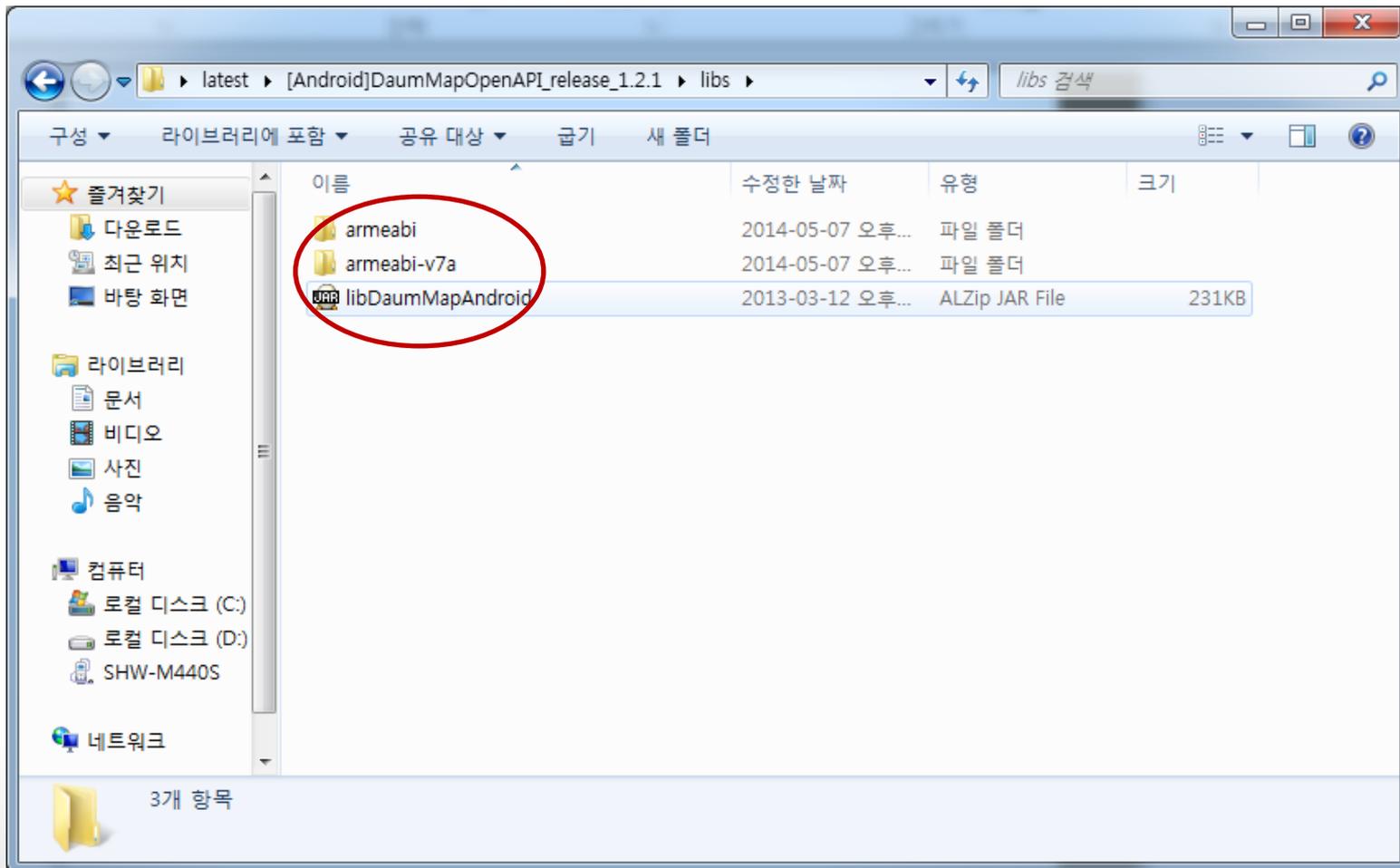
Compile With:

Theme:

💡 The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

사용법 - 라이브러리 복사

- 두 개의 폴더 및 jar파일을 복사



사용법 - 라이브러리 추가

- 프로젝트 폴더의 libs에 복사

The screenshot displays the Eclipse IDE interface for an Android project named 'TestOfDaumMapApi'. The Package Explorer on the left shows the project structure, with the 'libs' folder highlighted by a red circle. The main editor shows the MainActivity.java file, which extends ActionBarActivity and implements onCreate, onCreateOptionsMenu, and onOptionsItemSelected methods. The Outline on the right shows the class structure, including MainActivity and PlaceholderFragment. The LogCat window at the bottom is empty.

```
package com.example.testofdaummapapi;

import android.support.v7.app.ActionBarActivity;

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (savedInstanceState == null) {
            getSupportFragmentManager().beginTransaction()
                .add(R.id.container, new PlaceholderFragment()).commit();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
    }
}
```

사용법 - 라이브러리 추가

- 프로젝트 폴더의 Properties 클릭

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with the 'src' folder expanded. A context menu is open over the 'src' folder, and the 'Properties' option at the bottom is circled in red. The main editor window shows the code for MainActivity.java, which extends ActionBarActivity and implements onCreate, onCreateOptionsMenu, and onOptionsItemSelected methods. The LogCat window at the bottom is empty.

```
com.example.testofdaummapapi;

import android.support.v7.app.ActionBarActivity;

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

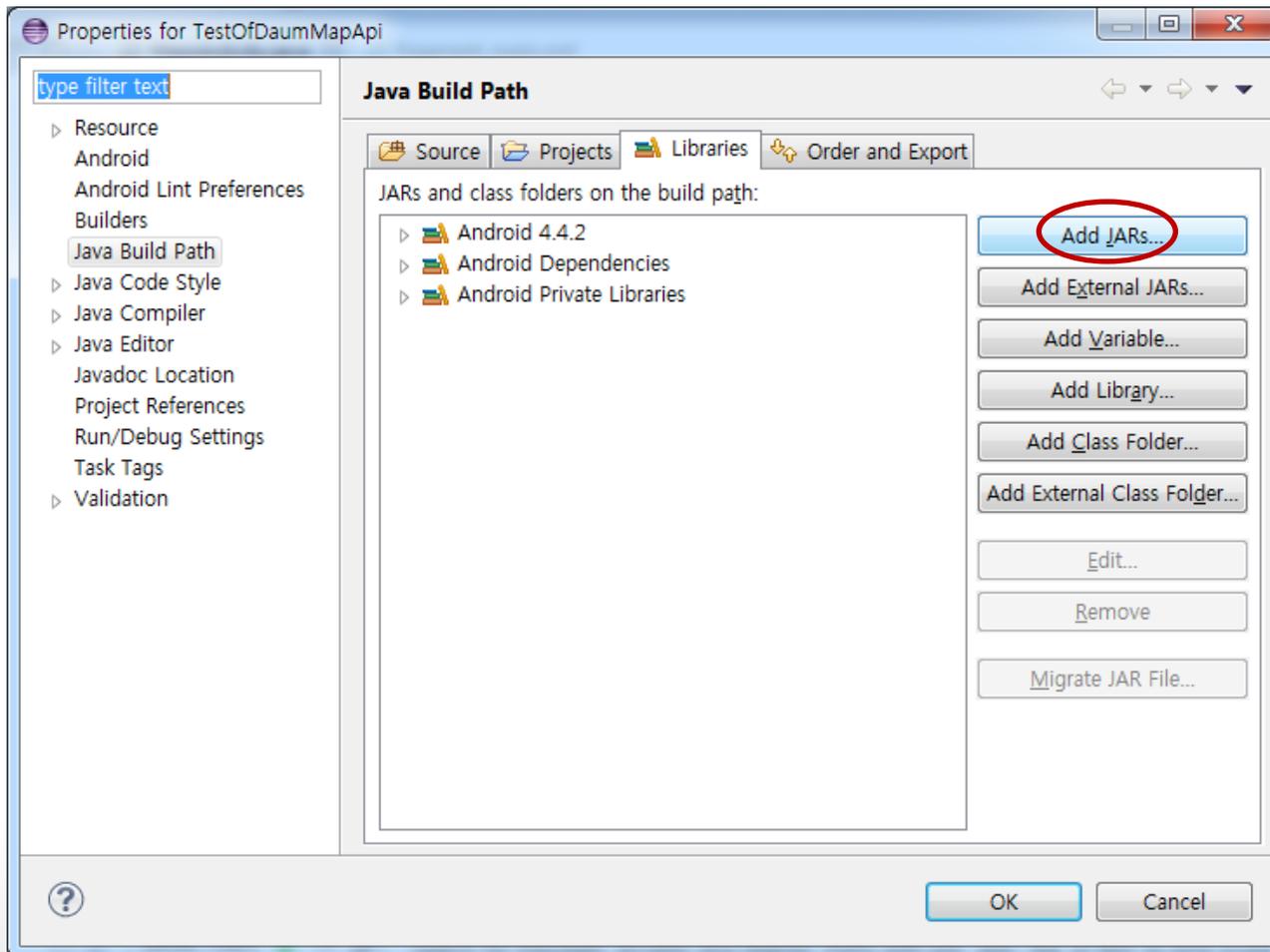
        if (savedInstanceState == null) {
            getSupportFragmentManager().beginTransaction()
                .add(R.id.container, new PlaceholderFragment()).commit();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
    }
}
```

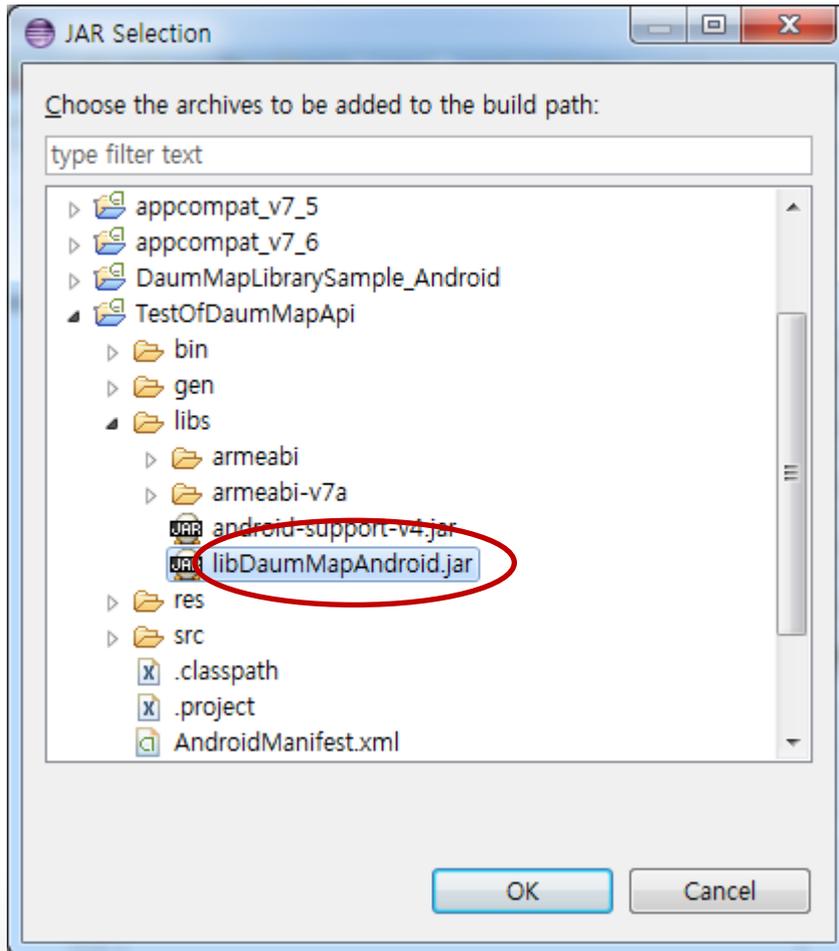
사용법 - 라이브러리 추가

- Java Build Path에 라이브러리 추가



사용법 - 라이브러리 추가

- libs폴더의 libDaumMapAndroid.jar 를 추가



여기까지 하면 다음 모바일 맵 api 사용을 위한 준비가 끝난다.

지도를 띄우는 방법

- 어플리케이션에 지도를 넣는 두 가지 방법
 1. 웹 이용
 2. jar파일 이용

예제 - 지도 띄우기 (WebView)

1. 레이아웃에 WebView 추가
2. WebView 설정 수정, Activity를 만들 자바 클래스 구현
3. AndroidManifest.xml 에 인터넷 권한 설정

예제 - 지도 띄우기 (WebView)

- fragment_main.xml 에 WebView 추가

```
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.testofdaumwebview.MainActivity$PlaceholderFragment" >
    <WebView
        android:id="@+id/webview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" >
    </WebView>
</RelativeLayout>
```

예제 - 지도 띄우기(WebView)

- MainActivity.java 에 코드 추가

```
public class MainActivity extends Activity {  
    private WebView webView;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.fragment_main);  
  
        String url="http://dna.daum.net/examples/maps/maps3/mobile_simple.html";  
  
        webView = (WebView) findViewById(R.id.webview);  
        webView.getSettings().setJavaScriptEnabled(true);  
        webView.setVerticalScrollBarEnabled(true);  
        webView.setHorizontalScrollBarEnabled(true);  
        webView.loadUrl(url);  
    }  
}
```

True로 하게 되면 액정 우측에 스크롤바가 들어갈 부분을 남기게 된다.
False로 하게 되면 공백 없이 지도로 가득채워진다.

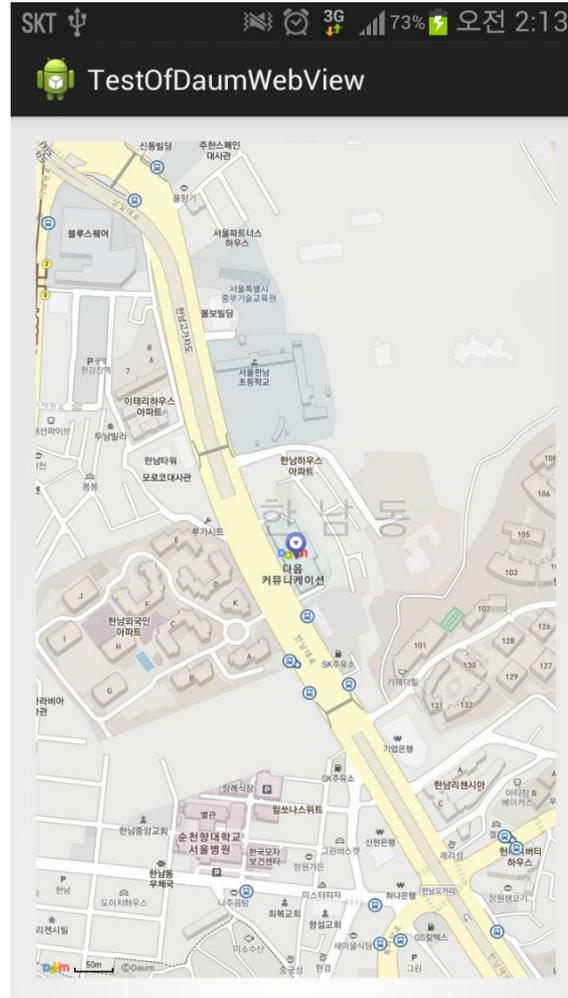
예제 - 지도 띄우기 (WebView)

- AndroidManifest.xml에 인터넷 권한 추가

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
        android:name="com.example.testofdaumwebview.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>
</application>
<uses-permission android:name="android.permission.INTERNET"> </uses-permission>
```

예제 - 지도 띄우기 (WebView)



예제 - 지도 띄우기 (WebView)

- 장점 : 빠른 리스트 로딩이 가능
- 단점 : 리스트 로딩 후에 스크롤 시 리스트가 전체적으로 느려짐

예제 - 지도 띄우기(jar)

- `net.daum.mf.map.api.MapView` 객체를 생성하여 다음 지도 화면을 보여주는 기능을 구현
- 지도화면이 보여지기 전에 API Key 발급 페이지를 통해 발급받은 Open API Key를 `MapView` 객체에 지정

예제 - 지도 띄우기(jar)

- 지도 종류는
`net.daum.mf.map.api.MapView.setMapType(MapType)` 메소드를 이용하여 선택
 - 기본지도 : `MapView.MapType.Standard`
 - 위성지도 : `MapView.MapType.Satellite`
 - 하이브리드 : `MapView.MapType.Hybrid`

예제 - 지도 띄우기(jar)

- fragment_main.xml에 버튼 추가

```
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.testofdaummapapi.MainActivity$PlaceholderFragment" >

    <Button
        android:id="@+id/ViewBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="다들 물어보기" />

</RelativeLayout>
```

예제 - 지도 띄우기(jar)

- MainActivity.java

```
package com.example.testofdaummapapi;

import android.app.Activity;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fragment_main);

        Button btn1=(Button)findViewById(R.id.ViewBtn);
        btn1.setOnClickListener(new OnClickListener(){

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent = new Intent(MainActivity.this, DaumMapView.class);
                startActivity(intent);
            }

        });
    }
}
```

DaumMapView라는 새로운 액티비티를 사용



DaumMapView.class

예제 - 지도 띄우기(jar)

- DaumMapView.java

```
public class DaumMapView extends Activity implements
    MapView.OpenAPIKeyAuthenticationResultListener
{

    private MapView mapView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        LinearLayout linearLayout = new LinearLayout(this);
        linearLayout.setLayoutParams(new LayoutParams(
            LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
        linearLayout.setOrientation(LinearLayout.VERTICAL);

        mapView = new MapView(this);
        mapView.setDaumMapApiKey("612a367127335365215b78b01ff662295e6163f2");
        mapView.setOpenAPIKeyAuthenticationResultListener(this);

        /* 맵 타입
        * mapView.setMapType(MapView.MapType.Hybrid);
        */
        mapView.setMapType(MapView.MapType.Standard);

        linearLayout.addView(mapView);

        setContentView(linearLayout);
    }
}
```

등록한 API key



예제 - 지도 띄우기(jar)

- import “MapView”(net.daum.mf.map.api.MapView) 클릭

The screenshot shows the Eclipse IDE interface. The main editor displays the `DaumMapView.java` file. The code is as follows:

```
public class DaumMapView extends Activity implements
    MapView.OpenAPIKeyAuthenticationResultListener
{
    private MapView mapView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.fragment_main);

        mapView = (MapView) findViewById(R.id.mapView);
        mapView.setMapType(MapView.MapType.Hybrid);
        mapView.setMapType(MapView.MapType.Standard);

        linearLayout.addView(mapView);

        setContentView(linearLayout);
    }
}
```

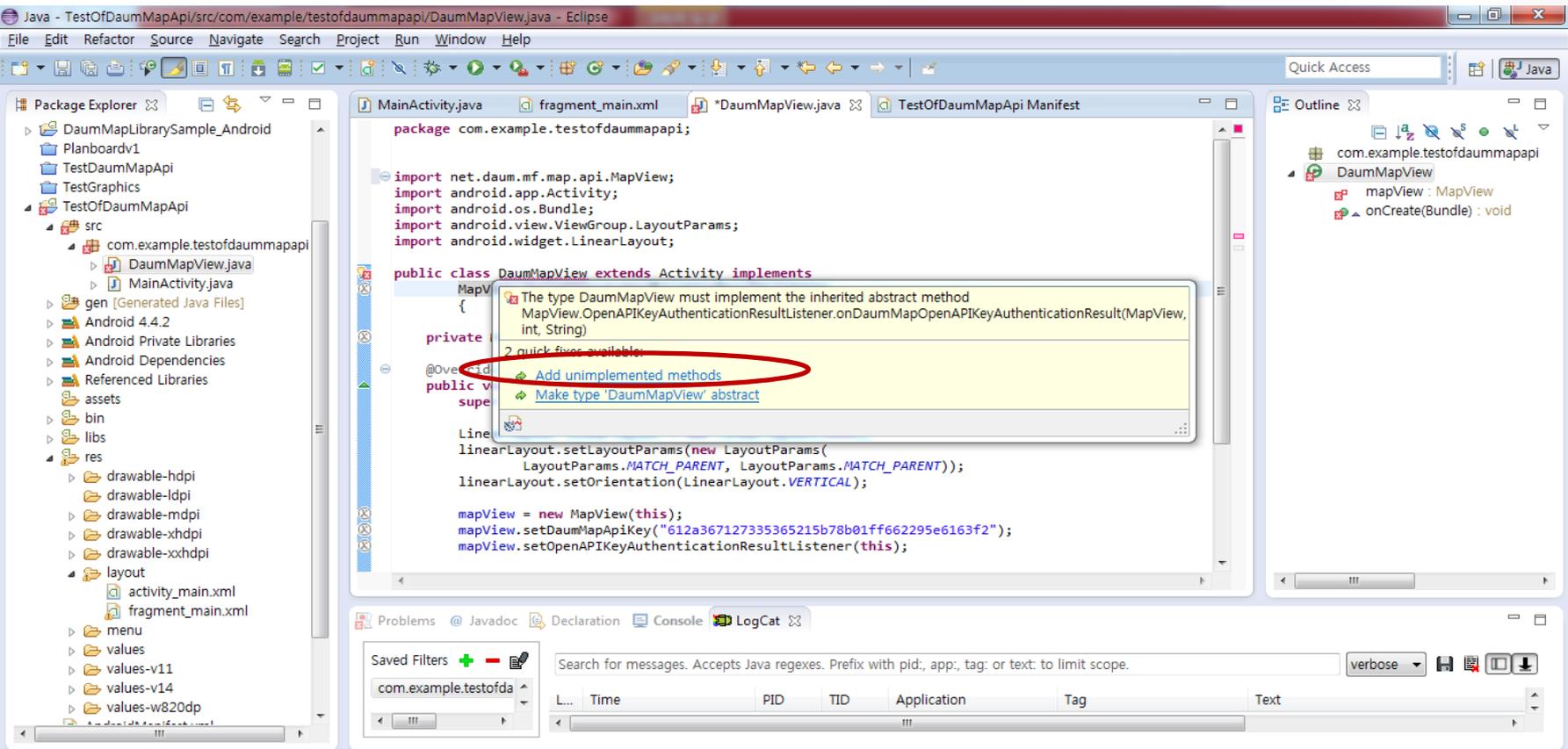
An error message is shown: "MapView cannot be resolved to a type". The quick fixes list includes:

- Import 'MapView' (net.daum.android.map)
- Import 'MapView' (net.daum.mf.map.api)** (highlighted with a red circle)
- Create class 'MapView'
- Create interface 'MapView'
- Change to 'MenuView' (android.support.v7.internal.view.menu)
- Create enum 'MapView'
- Add type parameter 'MapView' to 'DaumMapView'
- Fix project setup

The Package Explorer on the left shows the project structure, and the Outline on the right shows the class hierarchy. The bottom status bar shows the time as 17:34 and the date as 2014-05-11.

예제 - 지도 띄우기(jar)

- Add unimplemented methods 클릭



예제 - 지도 띄우기(jar)

- AndroidManifest.xml에 액티비티 및 권한 추가

```
<application  
  android:allowBackup="true"  
  android:icon="@drawable/ic_launcher"  
  android:label="@string/app_name"  
  android:theme="@style/AppTheme" >  
  <activity  
    android:name="com.example.testofdaummapapi.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>
```

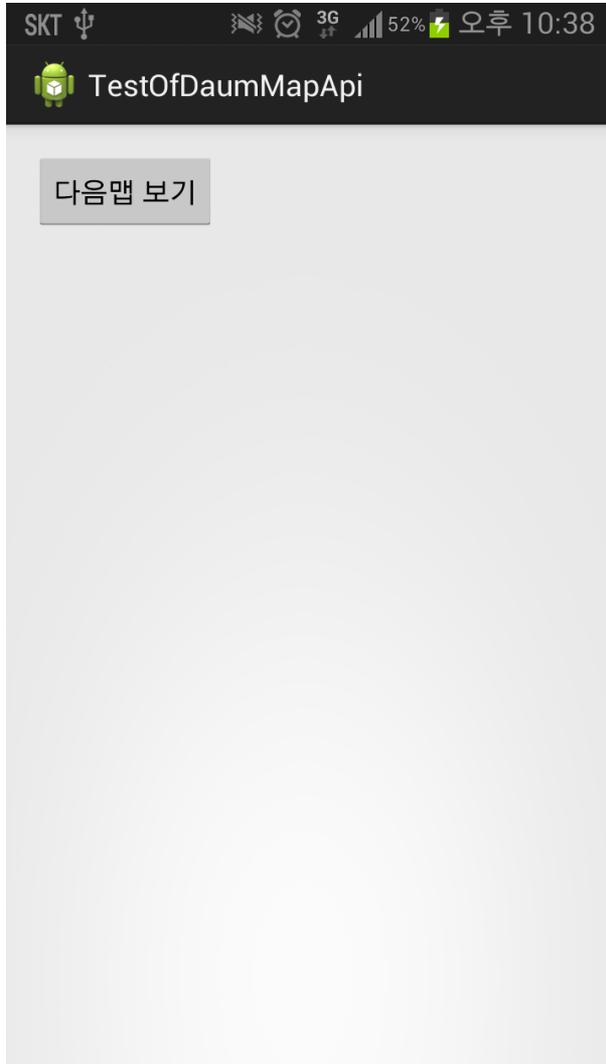
액티비티 추가

```
<activity android:name=".DaumMapView"  
  android:label="@string/app_name">  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.SAMPLE_CODE" />  
  </intent-filter>  
</activity>
```

권한 추가

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

예제 - 지도 띄우기(jar)



예제 - 자기 위치 찾기

- 단말기의 현재 위치에 따라 지도 화면이 이동하도록 현 위치 관련 기능을 사용
 - 현 위치 트래킹 모드 : 지도 화면 중심을 단말의 현재 위치로 이동
 - 나침반 모드 : 단말의 방향에 따라 지도화면이 회전됨
 - 트래킹 모드만 On 또는 트래킹 모드 나침반 모드 둘다 On 가능
 - 기능 활성화시 MapView.MapViewEventListener 객체로 좌표, 단말 각도 등이 전달됨

예제 - 자기 위치 찾기

- CurrentLocationEventListener추가

```
public class DaumMapView extends Activity implements  
    MapView.OpenAPIKeyAuthenticationResultListener, MapView.CurrentLocationEventListener  
{
```

- 추가하고 나서 앞서 했던 것과 같이 Add unimplemented methods 클릭

예제 - 자기 위치 찾기

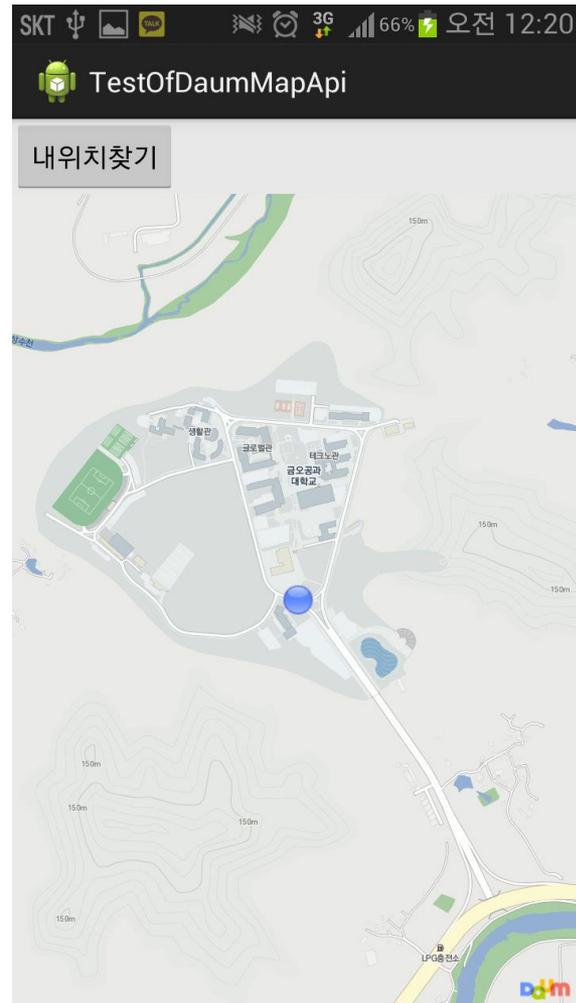
- DaumMapView.java에 버튼 추가

```
/*자기 위치 찾는 버튼 추가*/  
Button btn=new Button(this);  
btn.setText("내 위치 찾기");  
linearLayout.addView(btn, new LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT, LinearLayout.LayoutParams.WRAP_CONTENT));  
btn.setOnClickListener(new OnClickListener(){  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        mapView.setCurrentLocationTrackingMode(MapView.CurrentLocationTrackingMode.TrackingModeOnWithHeading);  
    }  
  
});
```



`mapView.setCurrentLocationTrackingMode(MapView.CurrentLocationTrackingMode.TrackingModeOnWithHeading);`
`//또는 TrackingModeOnWithoutHeading`

예제 - 자기 위치 찾기



관련 메서드

<u>addPOIItem(MapPOIItem)</u>	지도화면에 POI(point of interest) Item 마커를 추가
<u>addPOIItems(MapPOIItem[])</u>	지도화면에 POI(Point Of Interest) Item 아이콘(마커) 리스트를 추가
<u>addPoint(MapPoint)</u>	Polyline에 점 하나를 추가
<u>addPoints(MapPoint[])</u>	Polyline에 점 리스트를 추가
<u>addPolyline(MapPolyline)</u>	지도화면에 Polyline을 추가

관련메서드

<u>cancelFindingAddress()</u>	Reverse Geo-coding(Asynchronous) 서비스 요청을 취소한다
<u>clearMapTilePersistentCache()</u>	Application의 cache 디렉토리 (android.content.Context.getCacheDir())에 저장된 지도 타일 캐쉬 데이터를 모두 삭제
<u>deselectPOIItem(MapPOIItem)</u>	특정 POI(Point Of Interest) Item 을 선택 해제
<u>findAddressForMapPoint(String, MapPoint)</u>	동기(Synchronous, Blocking) 방식으로 Reverse Geo-coding 을 수행
<u>findPOIItemByName(String)</u>	지도화면에 추가된 POI(Point Of Interest) Item 들 중에서 이름이 일치하는 POI Item 객체를 찾음

관련메서드

<u>fitMapViewAreaToShowAllPOIItems()</u>	지도 화면에 추가된 모든 POI(Point Of Interest) Item들이 화면에 나타나도록 지도 화면 중심과 확대/축소 레벨을 자동으로 조정한다
<u>fitMapViewAreaToShowAllPolylines()</u>	지도 화면에 추가된 모든 Polyline 들이 화면에 나타나도록 지도 화면 중심과 확대/축소 레벨을 자동으로 조정한다.
<u>fitMapViewAreaToShowMapPoints(MapPoint[])</u>	지정한 지도 좌표들이 모두 화면에 나타나도록 지도화면 중심과 확대/축소 레벨을 자동조절한다
<u>fitMapViewAreaToShowPolyline(MapPolyline)</u>	특정 Polyline의 모든 점들이 화면에 전부 나타나도록 지도 화면 중심과 확대/축소 레벨을 자동으로 조정한다.

관련메서드

<u>getCurrentLocationTrackingMode()</u>	현재 설정된 현위치 트래킹 모드를 확인한다
<u>getCustomImageAnchorPointOffset()</u>	Custom Image Anchor Point Offset 좌표를 조회한다.
<u>getCustomImageResourceid()</u>	markerType이 net.daum.mf.map.api.MapPOIItem.MarkerType.CustomImage인 경우 설정된 POI Item 아이콘 custom image의 resource id를 조회한다.
<u>getItemName()</u>	POI Item 이름을 조회한다.
<u>getMapCenterPoint()</u>	현재 지도 화면의 중심점을 조회한다.

관련메서드

<u>isHDMapTileEnabled()</u>	고해상도 지도 타일 사용 여부를 조회 한다.
<u>isMapTilePersistentCacheEnabled()</u>	지도 타일 이미지 Persistent Cache 기능 활성화 여부를 조회한다
<u>isShowCalloutBalloonOnTouch()</u>	사용자가 POI Item 아이콘을 터치했을 때 말풍선(Callout Balloon)을 보여줄지 여부를 조회한다.
<u>isShowDisclosureButtonOnCalloutBalloon()</u>	Disclosure Button 이미지 표시 여부 조회한다.
<u>isShowingCurrentLocationMarker()</u>	현위치를 표시하는 아이콘(마커)를 화면에 표시되는지 여부를 확인한다.

관련메서드

<u>MapPOIItem()</u>	MapPOIItem 객체 생성
<u>MapPOIItem.ImageOffset(int, int)</u>	ImageOffset 객체 생성
<u>MapPoint.GeoCoordinate(double, double)</u>	Geoid 타원체 상의 한 점을 표현하는 지도 좌표 타입 (WGS84) 위도(latitude)와 경도(longitude)값으로 구성된다.
<u>MapPoint.PlainCoordinate(double, double)</u>	Geoid 상의 구면 좌표를 평면으로 프로젝션한 지도 좌표 정보를 저장하는 데이터
<u>mapPointWithGeoCoord(double, double)</u>	위경도 좌표 시스템(WGS84)의 좌표값으로 MapPoint 객체를 생성한다.

관련메서드

<u>MapPolyline()</u>	MapPolyline 객체 생성
<u>MapPolyline(int)</u>	Polyline을 구성하는 점들을 저장하는 Array의 크기를 미리 지정하여 MapPolyline 객체를 생성한다.
<u>MapReverseGeoCoder.ReverseGeoCodingResultListener, Activity)</u>	지도 좌표를 주소 정보로 변환(Reverse Geo-Coding)하는 기능을 제공하는 Class

관련메서드

<u>onCurrentLocationDeviceHeadingUpdate(MapView, float)</u>	단말의 방향(Heading) 각도값을 통보받을 수 있다.
<u>onCurrentLocationUpdateCancelled(MapView)</u>	현위치 트래킹 기능이 사용자에게 의해 취소된 경우 호출된다.
<u>onDaumMapOpenAPIKeyAuthenticationResult(MapView, int, String)</u>	<u>MapView.setDaumMapApiKey(String)</u> 메서드로 설정한 Open API Key값을 Daum Open API Key 인증 서버에 인증한 결과를 통보받을 수 있다.
<u>onDraggablePOIItemMoved(MapView, MapPOIItem, MapPoint)</u>	단말 사용자가 길게 누른 후(long press) 끌어서(dragging) 위치 이동이 가능한 POI Item의 위치를 이동시킨 경우 호출된다.
<u>onMapViewCenterPointMoved(MapView, MapPoint)</u>	지도 중심 좌표가 이동한 경우 호출된다.

관련메서드

<u>onMapViewInitialized(MapView)</u>	MapView가 사용가능 한 상태가 되었음을 알려준다.
<u>onMapViewLongPressed(MapView, MapPoint)</u>	사용자가 지도 위 한 지점을 길게 누른 경우 (long press) 호출된다.
<u>onMapViewZoomLevelChanged(MapView, int)</u>	지도 확대/축소 레벨이 변경된 경우 호출된다.
<u>onPOIItemSelected(MapView, MapPOIItem)</u>	단말 사용자가 POI Item을 선택한 경우 호출된다.

관련메서드

<u>refreshMapTiles()</u>	지도 화면에 나타나는 지도 타일들을 지도 타일 서버에서 다시 받아와서 갱신한다
<u>releaseUnusedMapTileImageResources()</u>	지도 타일 이미지 캐쉬 데이터 들을 삭제하여 메모리를 확보할 수 있다.
<u>removePOIItem(MapPOIItem)</u>	특정 POI(Point Of Interest) Item을 지도화면에서 제거한다.
<u>removePolyline(MapPolyline)</u>	특정 Polyline을 지도화면에서 제거한다.

관련메서드

<u>setCurrentLocationTrackingMode(MapView.CurrentLocationTrackingMode)</u>	현위치 트래킹 모드 및 나침반 모드를 설정한다
<u>setCustomImageResourceid(int)</u>	markerType이 net.daum.mf.map.api.MapPOIItem.MarkerType.CustomImage인 경우에만 지정한다.
<u>setDaumMapApiKey(String)</u>	Daum Map Open API Key 발급 페이지에서 발급 받은 API Key를 설정한다.
<u>setDraggable(boolean)</u>	사용자가 위치를 변경할 수 있는 POI Item을 생성하려면 draggable property를 true로 설정한다.
<u>setMapViewEventListener(MapView.MapViewEventListener)</u>	MapViewEventListener interface를 구현하는 객체를 MapView 객체에 등록하여 지도 이동/확대/축소, 지도 화면 터치(Single Tap / Double Tap / Long Press) 이벤트를 통보받을 수 있다.

관련메서드

<u>zoomIn(boolean)</u>	지도 화면을 한단계 확대한다.
<u>zoomOut(boolean)</u>	지도 화면을 한단계 축소한다.

참고문헌

- <http://simit.tistory.com/26> (2014.05.12 검색함)
- <http://dna.daum.net/apis/mmaps/android> (2014.05.12)
- <https://github.com/8thdbteam/testRepository>
 - PPT의 예제 소스 업로드