

NaverCafe SDK for Android

- 0. 요구사항
- 1. 네이버 카페 만들기
- 2. 네아로 어플리케이션 등록
- 3. 다운로드 SDK
- 4. 사용되는 라이브러리
- 5. SDK 설정
 - 1. Android Studio
 - 2. Eclipse
- 6. 네이버 카페 SDK 사용 방법
- 7. 리소스(이미지) 변경을 원할 경우

0. 요구사항

Android 4.2 'Jelly Bean' - API Level 17

(Android 4.1 'Jelly Bean' - API Level 16은 다음 버전에 제공할 예정입니다.)

1. 네이버 카페 만들기

<http://cafe.naver.com>

생성된 Cafe ID가 필요합니다.

2. 네아로 어플리케이션 등록

<https://nid.naver.com/devcenter/main.nhn>

생성된 Client ID, Client Secret이 필요합니다.

3. 다운로드 SDK

SDK :

1. Android Studio : cafesdk-x.x.x.aar
2. Eclipse : cafesdk-x.x.x.zip

SAMPLE :

1. Android Studio : navercafesdk-sample-android-studio.zip
2. Eclipse : navercafesdk-sample-eclipse.zip

4. 사용되는 라이브러리

다음 링크를 통해서 직접 다운 또는 maven 설정을 통하여 다운받아 SDK와 함께 컴파일が必要です.

아래 제공된 maven URL을 통하여 최신버전으로 다운로드 받거나 각 툴별 maven 참조해서 다운받으면 됩니다.

(제공되는 SDK의 library폴더에 포함되어 있습니다.)

0. 네이버 아이디 로그인 (<https://nid.naver.com/devcenter/docs.nhn?menu=Android>)

1. android support library v13
2. naver volleyer (<http://mvnrepository.com/artifact/com.navercorp.volleyextensions/volleyer>) (2.0.1 사용 중)
3. volley (<http://mvnrepository.com/artifact/com.mcxiaoke.volley/library/>) (1.0.2사용 중)
4. google gson (<http://mvnrepository.com/artifact/com.google.code.gson/gson>) (2.3.1 사용 중)
5. glide (<http://mvnrepository.com/artifact/com.github.bumptech.glide/glide>) (3.6.1 사용 중)
6. otto (<http://mvnrepository.com/artifact/com.squareup/otto>) (1.3.8 사용 중)

5. SDK 설정

1. Android Studio

1. cafeSdk-x.x.x.aar 라이브러리 파일 libs에 추가 (또는 import .JAR/.AAR Package)

2. Compile Dependencies 라이브러리 추가

라이브러리를 어플리케이션의 build.gradle에 직접 추가 또는 아래와 같이 설정하셔도 됩니다.

(네이버 아이디 로그인 JAR파일은 app의 libs에 넣어 함께 컴파일 되게 해 주세요)

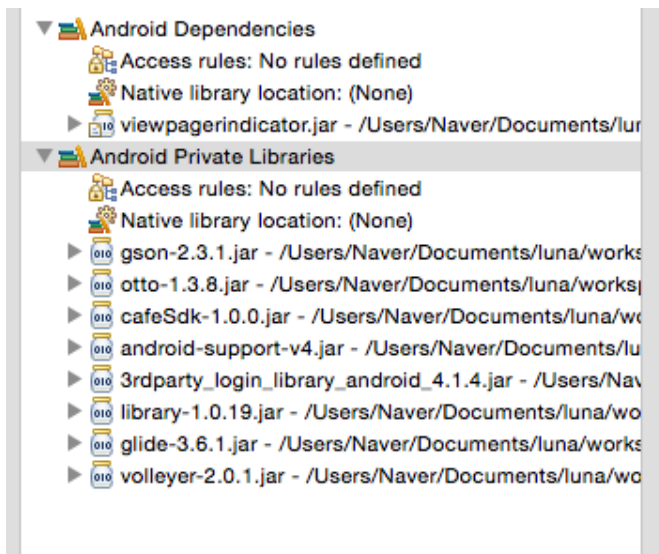
```
compile 'com.android.support:support-v13:23.1.0'
compile 'com.navercorp.volleyextensions:volleyer:2.0.1'
compile 'com.google.code.gson:gson:2.3.1'
compile 'com.github.bumptech.glide:glide:3.6.1'
compile 'com.squareup.otto:1.3.8'
```

2. Eclipse

1. cafeSdk-x.x.x.zip 파일 압축 해제

2. Compile Dependencies 라이브러리 추가

2.1. 압축 해제된 폴더의 libs에 함께 넣어 주시면 됩니다.



3. cafeSdk 프로젝트 is library 로 체크

4. main app에 cafeSdk 라이브러리 추가

5. main app Androidmanifest.xml에 아래 activity 추가 (넣어주지 않을 경우 런타임시에 에러가 납니다)

```
<activity
    android:name="com.naver.glink.android.sdk.ui.VideoPlayActivity"
    android:screenOrientation="sensorLandscape"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />
<activity
    android:name="com.nhn.android.naverlogin.ui.OAuthLoginActivity"
    android:screenOrientation="sensorLandscape"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.nhn.android.naverlogin.ui.OAuthLoginInAppBrowserActivity"
    android:label="OAuth2.0 In-app"
    android:screenOrientation="sensorLandscape" />
```

6. 네이버 카페 SDK 사용 방법

AndroidManifest.xml 설정.

```
<!-- package이 "네아로 개발자 센터"에 등록된 "안드로이드 앱 Package명"과 동일해야 합니다. -->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.naver.glink.sample">
```

```
<!-- Glink에 필요한 permission -->
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Glink class에 SDK 초기화/시작/종료등 모든 작업을 실행할 수 있는 static method가 있습니다.

SDK 초기화와 시작하기.

```

/**
 * "네아로 개발자 센터"에서 받은 정보로 SDK를 초기화 합니다.
 * Glink의 다른 메소드를 호출하기 전에 반드시 초기화를 먼저해야 합니다.
 * 개발자 센터 주소: https://nid.naver.com/devcenter/main.nhn
 */
final int cafeld = 28266581;
final String consumerKey = "your consumer key";
final String consumerSecret = "your consumer secret";
Glink.init(consumerKey, consumerSecret, cafeld);

// 홈화면으로 시작합니다.
Glink.startHome(activity);

// 공지화면으로 시작합니다.
Glink.startNotice(activity);

// 이벤트화면으로 시작합니다.
Glink.startEvent(activity);

// 메뉴화면으로 시작합니다.
Glink.startMenu(activity);

// 프로필화면으로 시작합니다.
Glink.startProfile(activity);

SDK 화면 종료하기.

Glink 화면을 닫는 방법은 stop/popBackStack 두 가지 입니다.
stop은 화면을 바로 닫고, popBackStack은 쌓여 있는 화면을 하나씩 닫다가 화면이 없으면 종료됩니다.
(backStack으로 쌓일 수 있는 화면은 게시물/글쓰기/검색 화면입니다.)
// stop으로 종료하기.
Glink.stop(activity);

// popBackStack으로 종료하기.
Glink.popBackStack(activity);

게시글 쓰기.
// 제목, 본문
int menuId = 4; // 0이면 메뉴를 선택하지 않는다.
String text = "기본 제목, 본문을 넣어서 글쓰기 화면을 시작합니다.";
Glink.startWrite(MainActivity.this, "subject", text);

// 제목, 본문, 이미지
int menuId = 4; // 0이면 메뉴를 선택하지 않는다.
String text = "기본 제목, 본문, 이미지를 넣어서 글쓰기 화면을 시작합니다.\n이미지는 uri 형태로 넣어주시면 됩니다.";
String path = "your image uri";
Glink.startImageWrite(MainActivity.this, menuId, "subject", text, path);

// 제목, 본문, 동영상
int menuId = 4; // 0이면 메뉴를 선택하지 않는다.
String text = "기본 제목, 본문, 비디오를 넣어서 글쓰기 화면을 시작합니다.\n비디오는 uri 형태로 넣어주시면 됩니다.";
String path = "your video uri";
Glink.startVideoWrite(MainActivity.this, menuId, "subject", text, path);

홈 배너 설정하기. (앱스킴 타입)
홈 배너의 타입을 앱스킴으로 한 경우, OnClickAppSchemeBannerListener를 설정하여 유저의 터치를 리스닝할 수 있습니다.

```

```
// 앱스킴 터치 리스너 설정.
Glink.setOnClickAppSchemeBannerListener(new Glink.OnClickAppSchemeBannerListener() {
    @Override public void onClickAppSchemeBanner(String appScheme) {
        // 카페 관리에서 설정한 appScheme 문자열을 SDK에서 넘겨줍니다.
        // 각 appScheme 처리를 이곳에서 하시면 됩니다.
    }
});
```

```
GameUserId 설정
// 게임 아이디와 카페 아이디를 매핑합니다.
// 프로필 화면에서 매핑된 게임 아이디를 확인할 수 있습니다.
Glink.setGameUserId(this, "gameUserId", "게임 ID");
```

7. 리소스(이미지) 변경을 원할 경우

1. AAR Library
 - a. 다운받은 aar 라이브러리 파일을 zip을 이용하여 압축을 푼다
 - b. /res/drawable-xhdpi 폴더의 이미지들을 원하는 이미지로 수정 후 재압축
2. zip Library
 - a. /res/drawable-xhdpi 폴더의 이미지들을 원하는 이미지로 수정 후 재빌드