



## **MoPub Mediation Guide for Android**

- SUEZ-X SDK with MoPub SDK**
- For Eclipse**

문서는 TPMN 제휴 당사자에 한해 제공되는 자료로 가이드 라인을 포함한 모든 자료의 지적재산권은 TPMN이 보유합니다.

**COPYRIGHT© TPMN. ALL RIGHTS RESERVED.**

# 목차

변경이력.....	3
SDK 적용 방법.....	4
Ad Unit ID 발급 받기.....	4
MoPub SDK 적용 방법(필수 사항).....	4
SUEZ-X SDK 적용 방법(필수 사항).....	13
Android Manifest 설정 방법.....	14
Proguard 설정 방법.....	15
MoPub 배너 광고 적용 방법.....	16
MoPub 전면 광고 적용 방법.....	18
SUEZ-X Adapter 적용 방법(필수 사항).....	21
AdMob 추가 방법(필수 사항).....	22
AdMob SDK 적용 방법.....	22
CustomEvent 파일 추가.....	22
FAN(Facebook Audience Network) 추가 방법.....	23
FAN SDK 적용 방법.....	23
CustomEvent 파일 추가.....	23
MoPub Adapter 제공 네트워크.....	24
주의 사항.....	25
Q&A.....	26
Issue Case.....	28

## 변경이력

- Version 1.0

최초 작성

- Version 1.1

Issue Case 추가

# SDK 적용 방법

정상적인 광고 요청 및 노출을 위해 MoPub SDK, SUEZ-X SDK, SUEZ-X Adapter를 꼭 적용하여 주셔야 합니다.

## Ad Unit ID 발급 받기

광고 호출 시 사용할 광고 상품 ID 발급을 위해서, Mediation 기능을 사용할 App 정보를 TPMN에 전달해 주시면 App등록 후 광고 상품 ID(Ad Unit ID)를 전달해 드립니다.

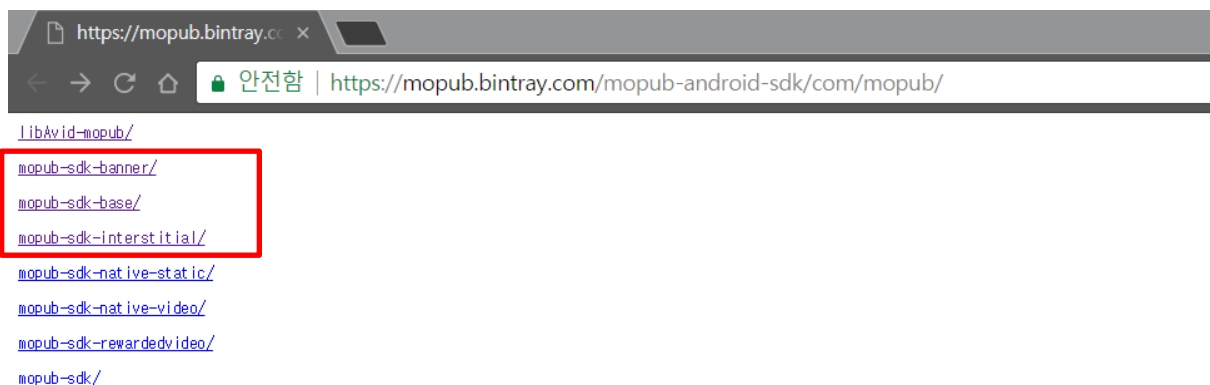
\* 진행 가능 광고 : 배너(320x50),(300x250) , 전면광고(320x480)

## MoPub SDK 적용 방법

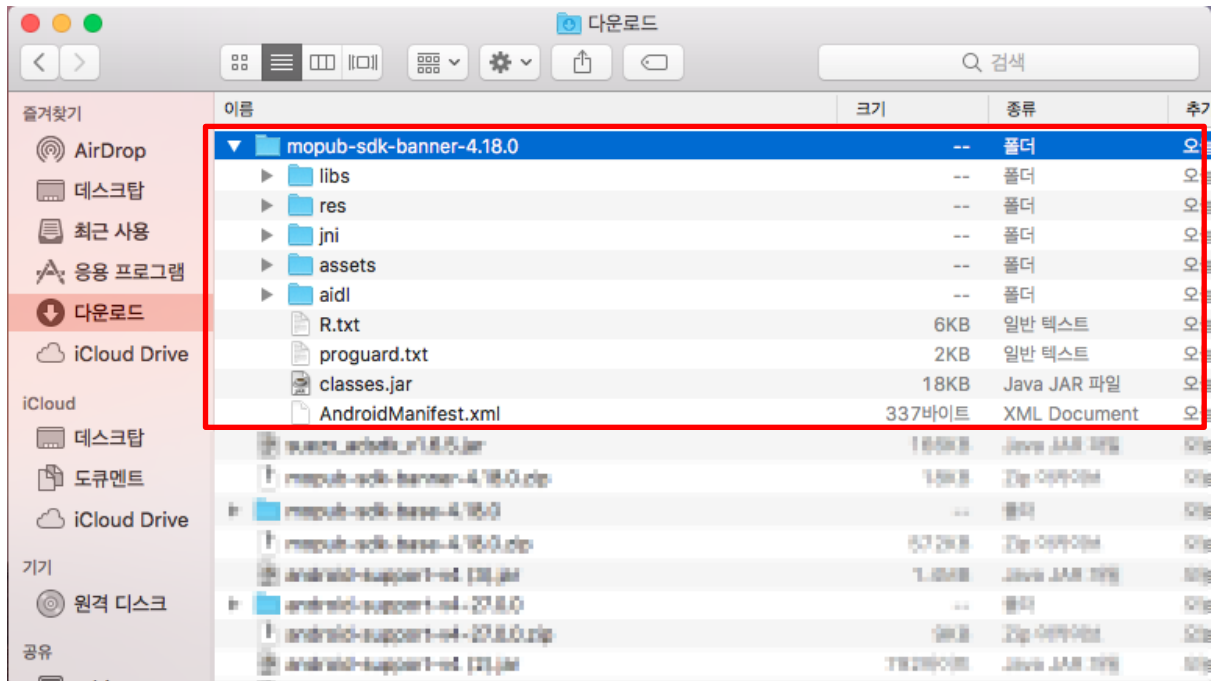
### MoPub SDK import to Eclipse Project

MoPub SDK(AAR) : <https://mopub.bintray.com/mopub-android-sdk/com/mopub/>

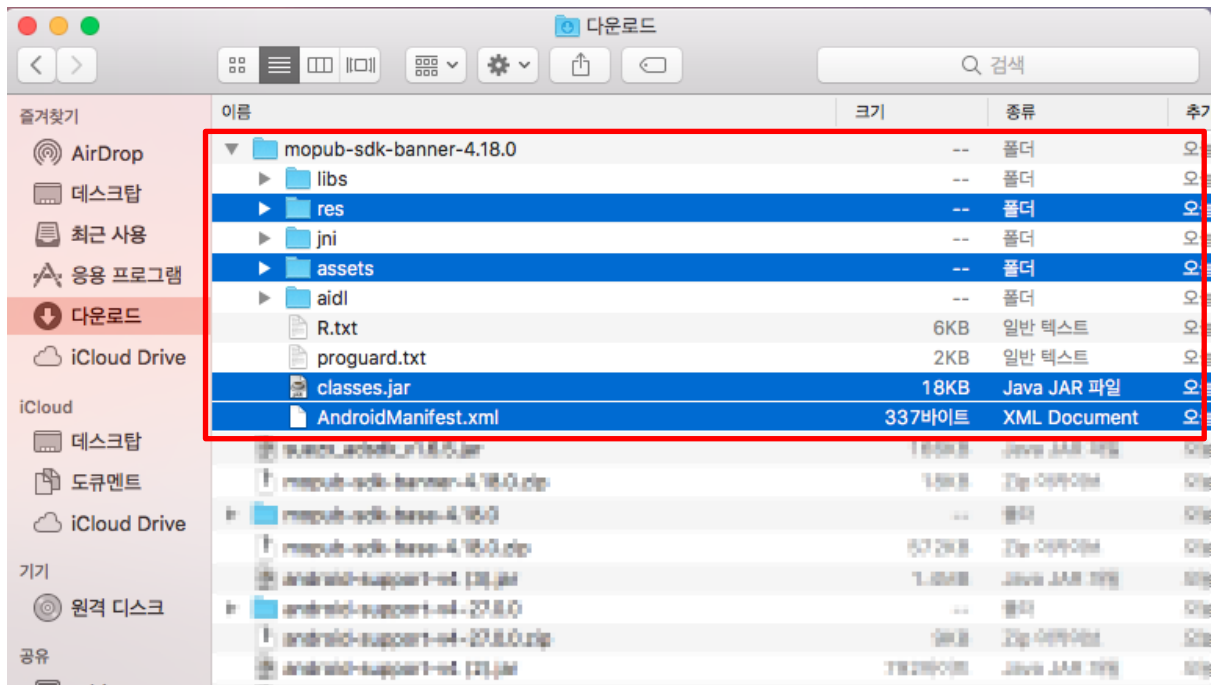
1. 위의 링크에서 mopub-sdk-base와 사용하시는 광고 상품(Banner, Interstitial)에 알맞은 디렉토리를 선택하고 최신 버전(가이드 작성일 기준 4.18 버전)의 MoPub SDK AAR 파일들을 다운로드 합니다.



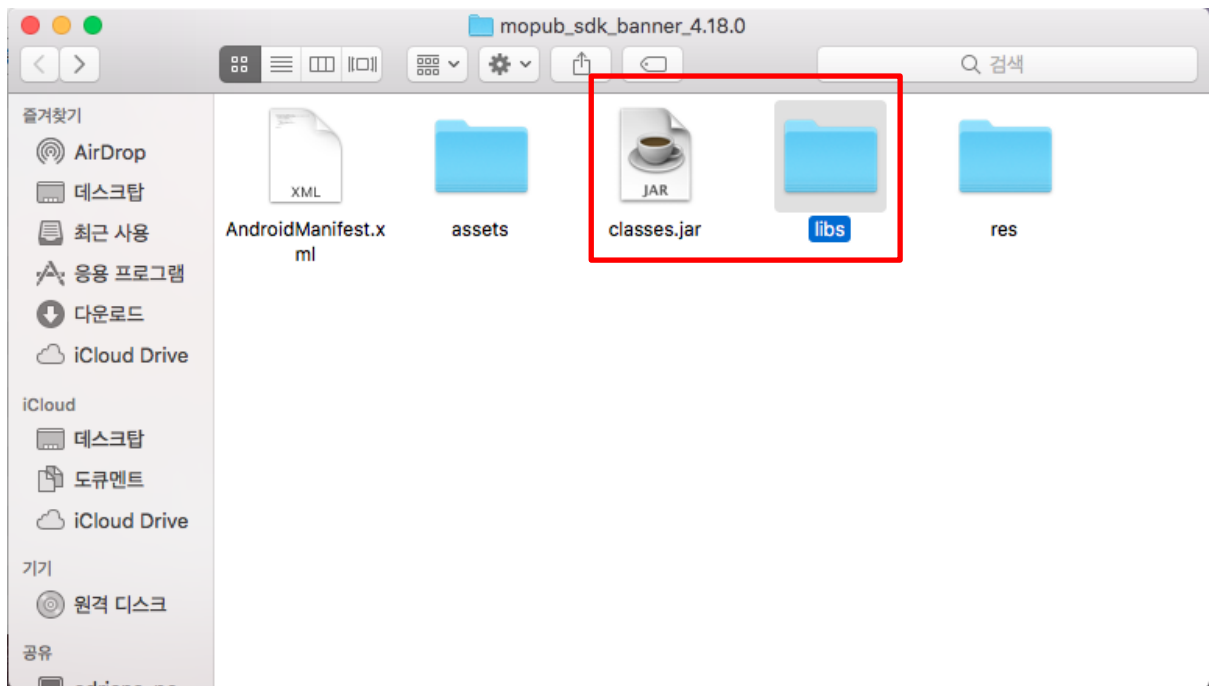
2. 다운로드 받은 .aar 파일의 확장자를 .zip으로 바꾸어 준 후 압축 해제해 줍니다.



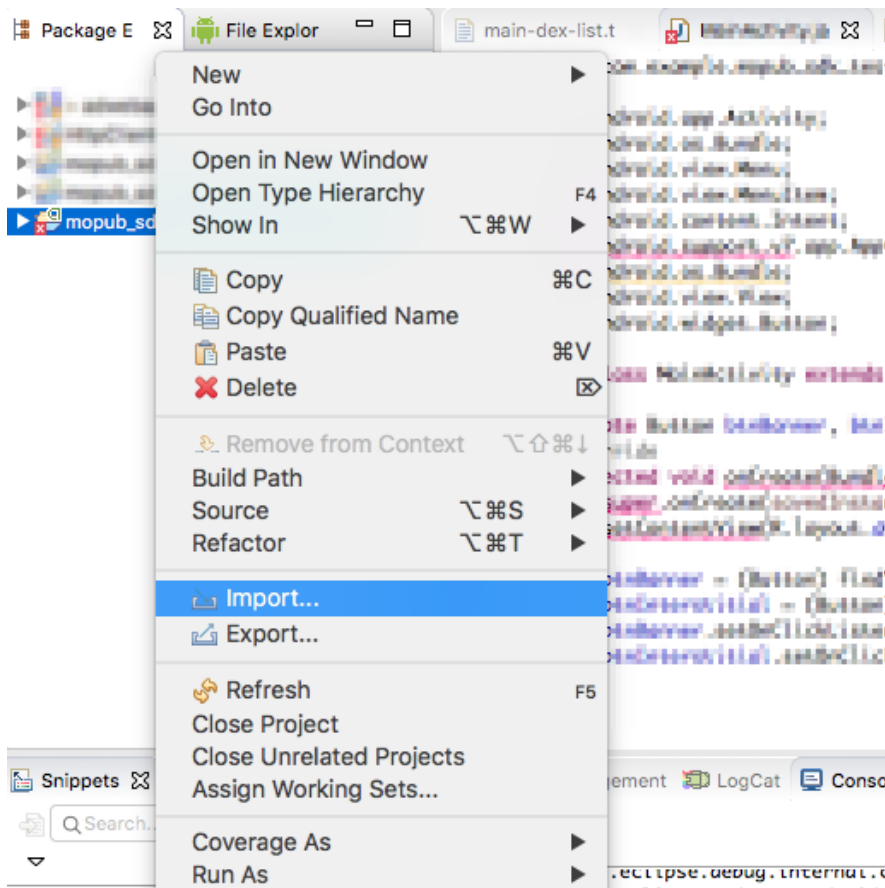
3. 압축해제한 폴더에서 res, assets 폴더와 classes.jar, AndroidManifest.xml 파일을 선택한 후 새 폴더를 만들어 그곳으로 옮겨줍니다.



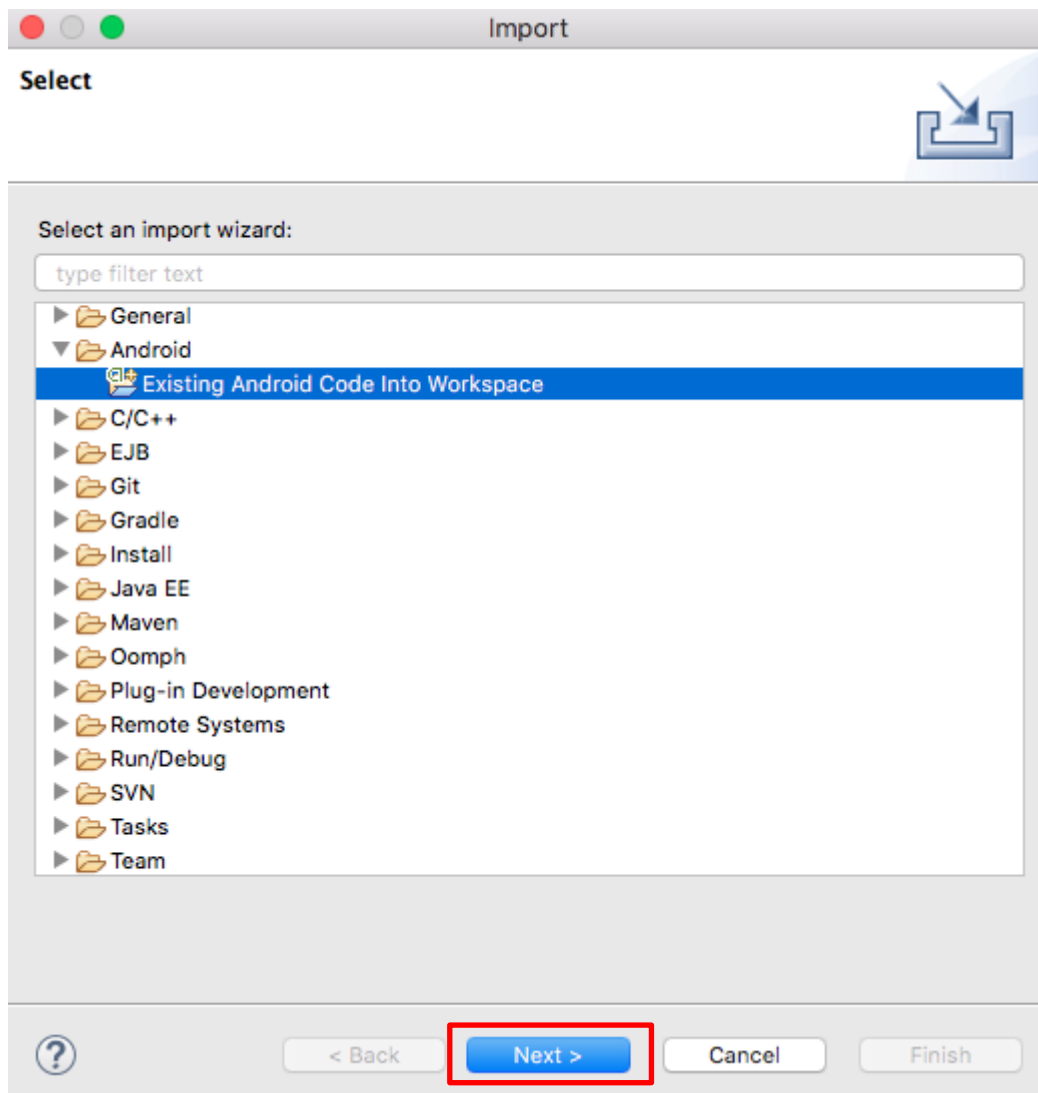
4. 새 폴더에 libs 폴더를 만든 후, classes.jar 파일의 이름을 mpub\_sdk\_xxxx 형식으로 바꾸어 준 후 libs 폴더에 넣어줍니다.



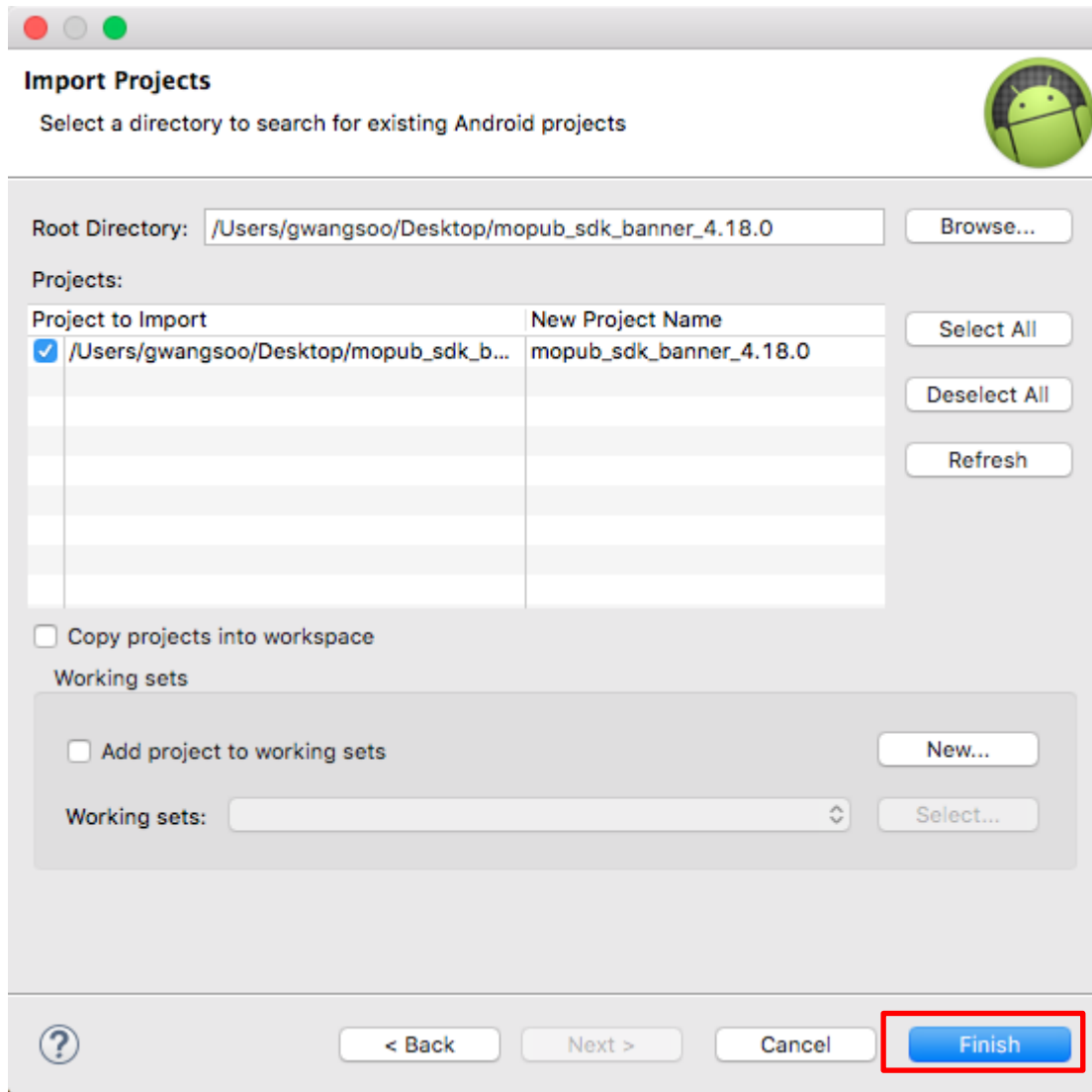
5. Eclipse에서 package 패널에서 마우스 오른쪽쪽을 누른 후 Import를 선택합니다.



6. Import에서 Existing Android Code Into Workspace 를 선택합니다.

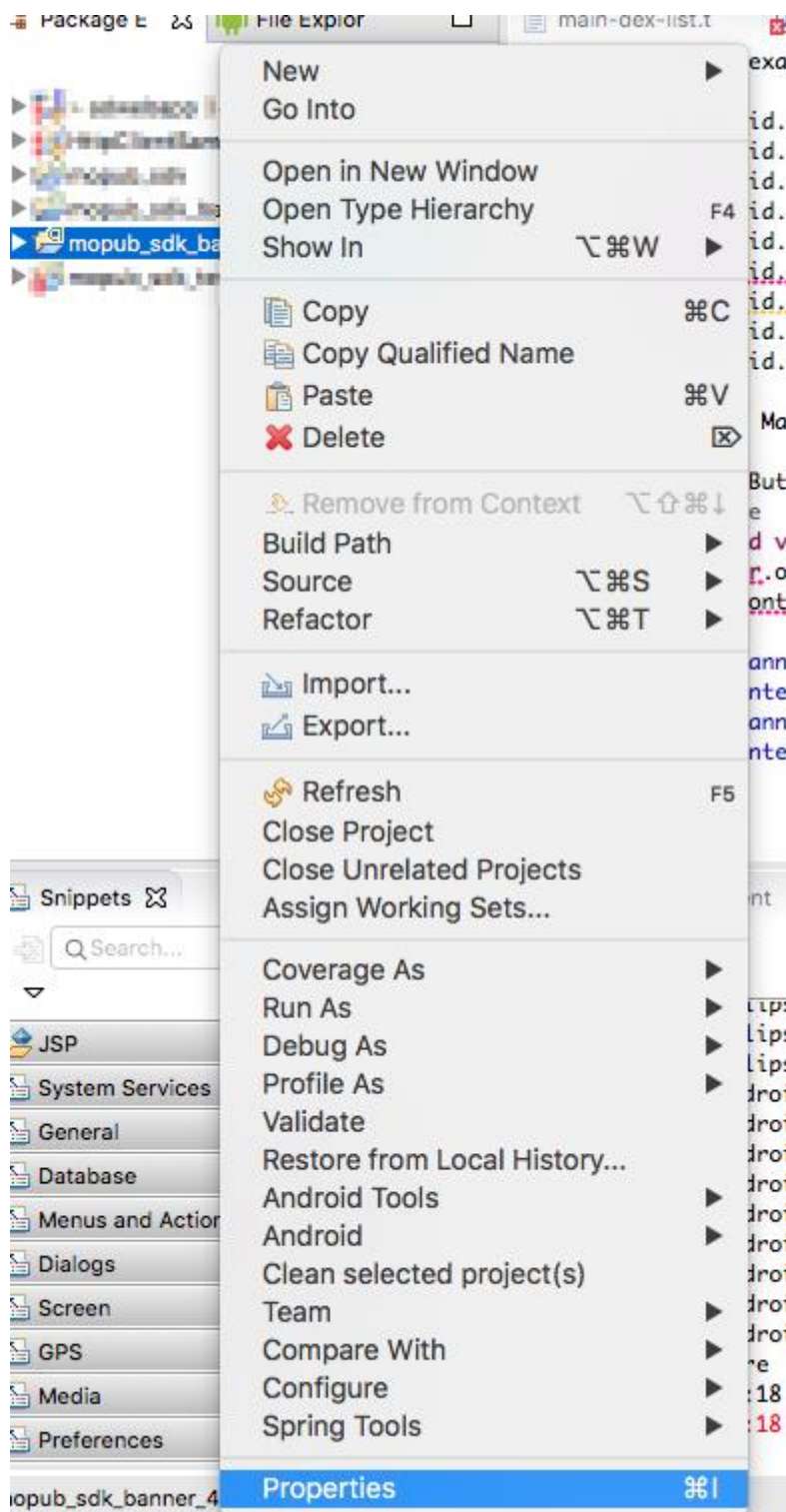


7. 3.에서 새로 만든 폴더의 경로를 찾아서 Import해줍니다.

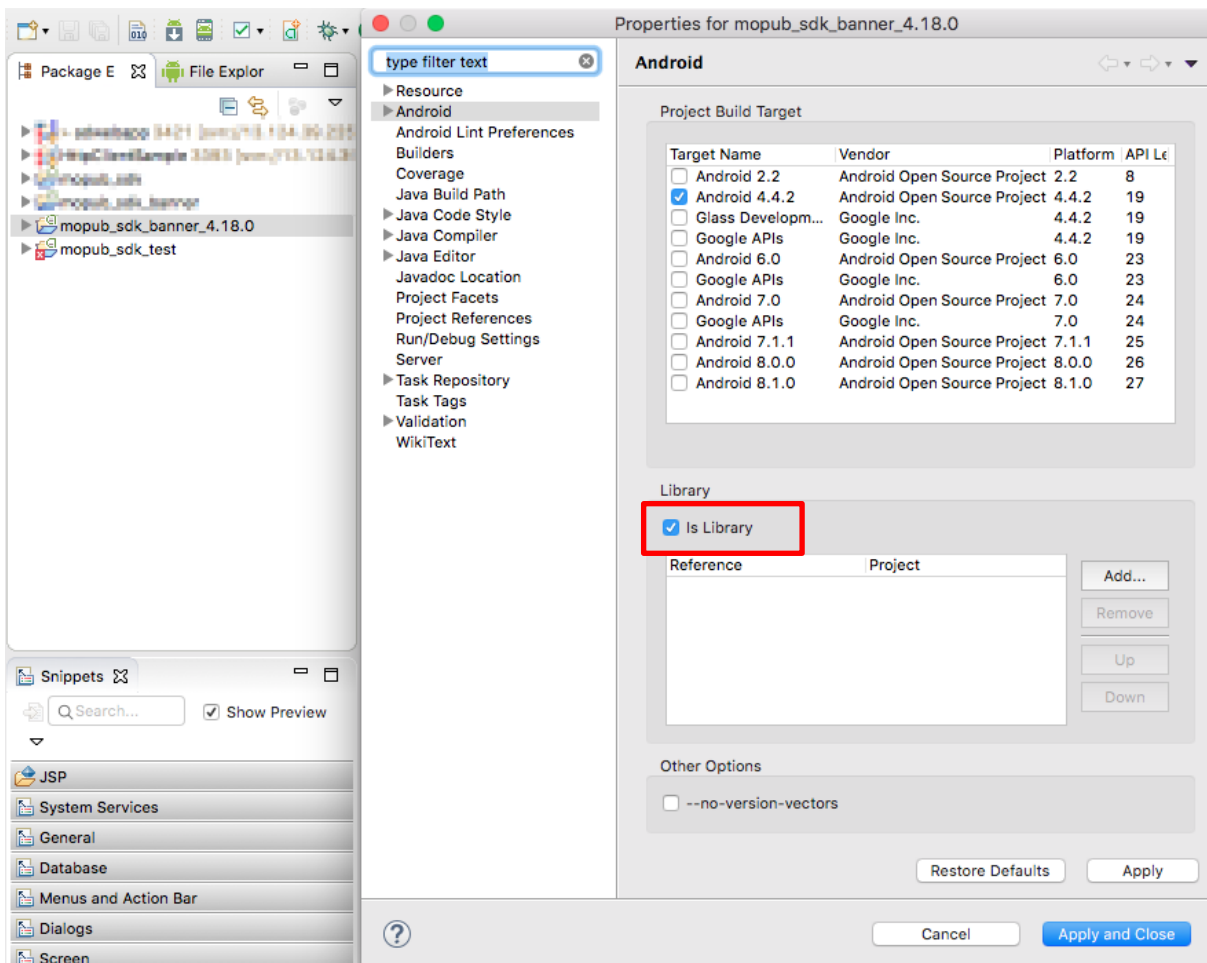




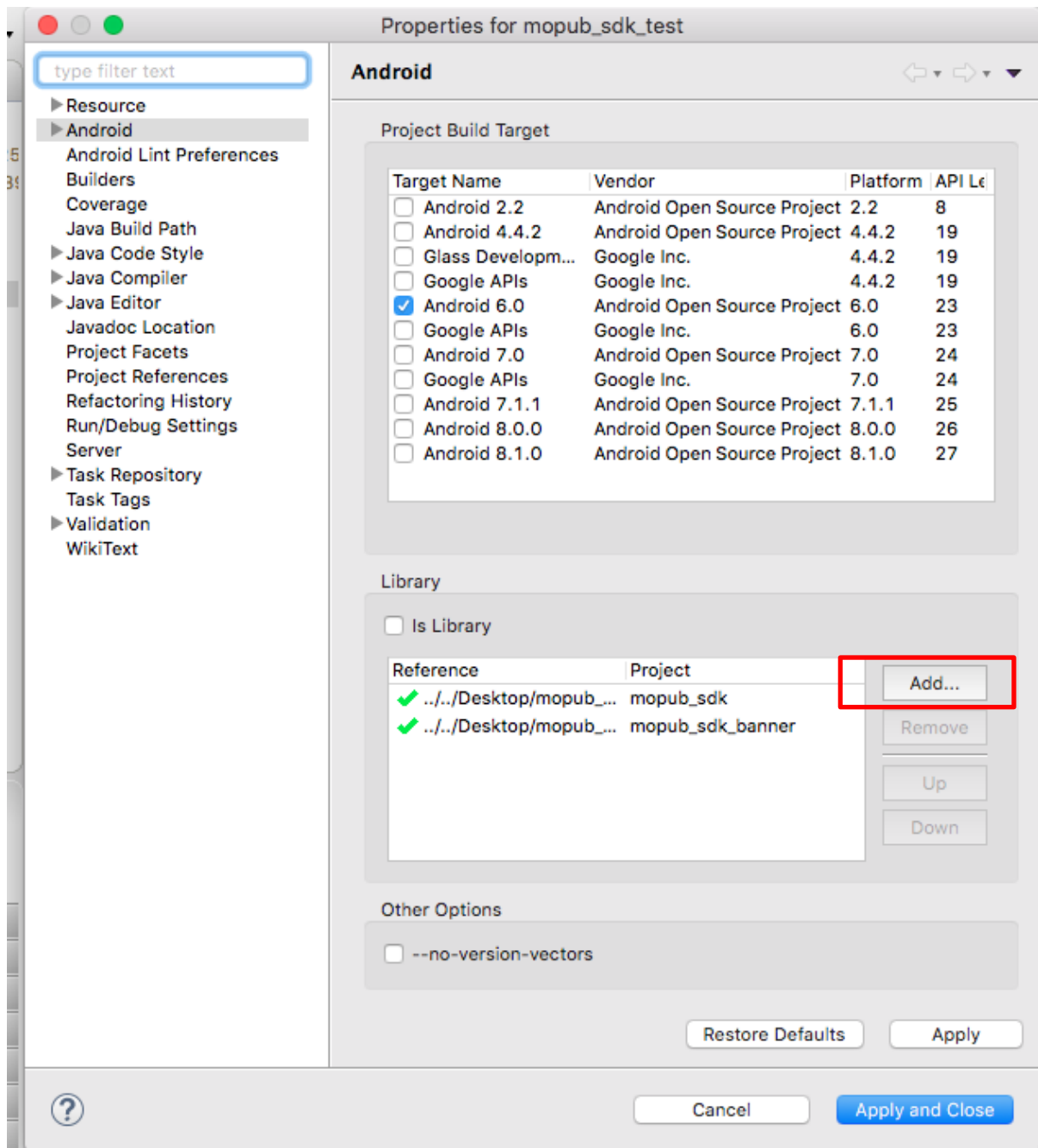
8. 7.에서 import를 통해 만들어진 프로젝트에서 마우스 오른쪽 버튼을 클릭하여 Properties 를 선택합니다.



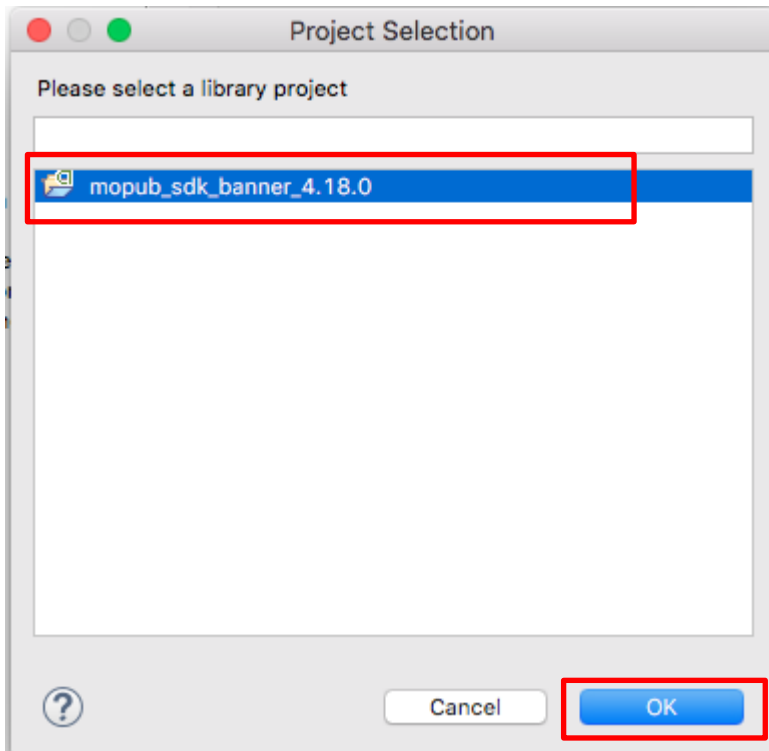
9. Android 탭의 Library의 is Library를 선택하여 해당 프로젝트를 라이브러리로 사용하겠다고 선언합니다.



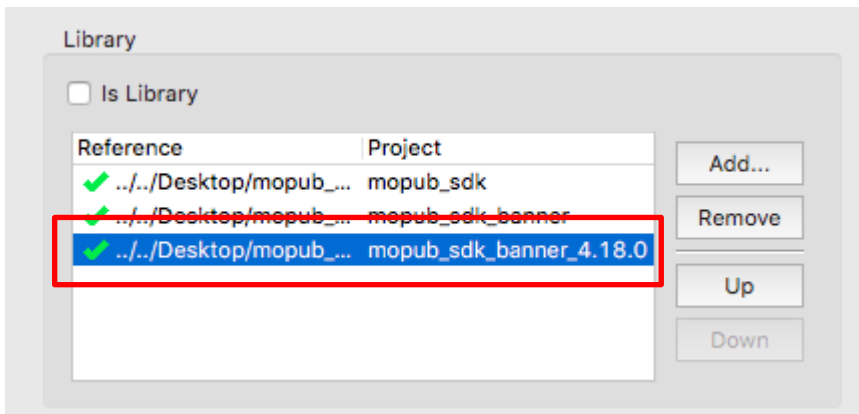
10. MoPub SDK를 적용하고자 하는 프로젝트의 마우스 오른쪽 버튼을 클릭하여 Properties를 선택합니다. 그 후 Android 탭의 Library에서 Add를 클릭합니다.



11. 9.에서 라이브러리로 사용하겠다고 선언한 프로젝트를 선택하고 OK버튼을 눌러줍니다.



다음과 같이 aar에서 추출한 파일이 라이브러리 형태로 추가된 것을 확인 하실 수 있습니다.



위와 같은 과정을 통해 mopub-sdk-base를 필수로 넣어 주시고 사용하시는 상품에 따라 mopub-sdk-banner, mopub-sdk-interstitial 라이브러리를 넣어 주셔야 합니다. 다시 말해서 배너 상품을 이용하신다면 mopub-sdk-base와 mopub-sdk-banner 두 개의 라이브러리 적용 작업을 진행하셔야 하고, 전면광고 상품을 이용하신다면 mopub-sdk-base와 mopub-sdk-interstitial 두 개의 라이브러리 적용 작업을 진행하셔야 하고, 배너 & 전면광고 모두 이용하신다면 mopub-sdk-base, mopub-sdk-banner, mopub-sdk-interstitial 세 개의 라이브러리 적용 작업을 진행하셔야 합니다.

## MoPub SDK 적용 시 주의 사항

### 요구 사항 및 종속성

1. Android 4.1(API버전 16이상) 이상
2. android-support-v4.jar, r22
3. android-support-annotations.jar, r22
4. android-support-v7-recyclerview.jar, r22
5. **mopub-volley-1.1.0.jar**
6. Google Play Services 10.2.0 이상 권장

Eclipse에서 진행 하시는 경우 위와 같은 요구사항과 라이브러리가 포함되어 있어야 합니다. 해당 라이브러리들이 포함되지 않아 빌드가 안되시는 경우 아래의 링크에서 받으실 수 있습니다. **Mopub-volley-1.1.0.jar** 를 제외한 라이브러리의 경우 Android Support Library를 통해 적용 하실 수 있습니다.

Github 링크 : [https://github.com/tpmn/SUEZ-X/tree/master/SUEZ-X-mopub-custom-adapter/Android/for\\_eclipse](https://github.com/tpmn/SUEZ-X/tree/master/SUEZ-X-mopub-custom-adapter/Android/for_eclipse)

## SUEZ-X SDK 적용 방법(필수 사항)

SUEZ-X GitHub : <https://github.com/tpmn/SUEZ-X/tree/master/SUEZ-X-android-sdk/suez-x-sdk>

MoPub SDK 적용 후 TPMN의 광고를 요청하기 위해서는 TPMN의 SUEZ-X SDK가 필요합니다.

SUEZ-X ADSDK jar 파일을 다운받아 프로젝트 내에 라이브러리로 적용합니다.

## Android Manifest 설정 방법

다음과 같은 permission을 선언하십시오.

### Mandatory(필수)

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

### Optional(선택)

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

\* 위의 권한 설정 시 광고 효율이 좋아질 수 있습니다. (위치 타겟팅 광고 가능, MRAID2.0 광고 가능)

<application> 에 다음의 activity 들을 선언하십시오.

```
<activity android:name="com.mopub.mobileads.MoPubActivity"
android:configChanges="keyboardHidden|orientation|screenSize"/>
<activity android:name="com.mopub.mobileads.MraidActivity"
android:configChanges="keyboardHidden|orientation|screenSize"/>
<activity android:name="com.mopub.common.MoPubBrowser"
android:configChanges="keyboardHidden|orientation|screenSize"/>
<activity android:name="com.mopub.mobileads.MraidVideoPlayerActivity"
android:configChanges="keyboardHidden|orientation|screenSize"/>
<activity android:name="com.mopub.mobileads.RewardedMraidActivity"
android:configChanges="keyboardHidden|orientation|screenSize"/>
```

TPMN의 전면 광고를 사용하기 원할 때는 아래의 내용을 추가해야 사용 가능 합니다.

```
<activity android:name="com.suezx.ad.SuezxInterstitialAdActivity"
android:theme="@android:style/Theme.Translucent"><!--SUEZ-X 전면광고 -->
```

Google Play Service 를 사용하기 위해 <application>에 다음과 같은 태그를 추가하십시오.

```
<meta-data android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
<activity android:name="com.google.android.gms.ads.AdActivity"
android:configchanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenSize|smallestScreenSize">
```

## Proguard 설정 방법 (선택사항)

앱에서 Proguard를 사용하고 있다면 아래의 내용을 추가하여 주셔야 합니다.

```
# Keep public MoPub classes and methods.
-keepclassmembers class com.mopub.** { public *; }
-keep public class com.mopub.**
-keep public class android.webkit.JavascriptInterface {}

# Explicitly keep any custom event classes in any package.
-keep class * extends com.mopub.mobileads.CustomEventBanner {}
-keep class * extends com.mopub.mobileads.CustomEventInterstitial {}
-keep class * extends com.mopub.nativeads.CustomEventNative {}
-keep class * extends com.mopub.nativeads.CustomEventRewardedAd {}

# Keep methods that are accessed via reflection
-keepclassmembers class ** { @com.mopub.common.util.ReflectionTarget *; }

# Viewability support
-keepclassmembers class com.integralads.avid.library.mopub.** { public *; }
-keep public class com.integralads.avid.library.mopub.**
-keepclassmembers class com.moat.analytics.mobile.mpub.** { public *; }
-keep public class com.moat.analytics.mobile.mpub.**

# Support for Android Advertiser ID.
-keep class com.google.android.gms.common.GooglePlayServicesUtil {*;}
-keep class com.google.android.gms.ads.identifier.AdvertisingIdClient {*;}
-keep class com.google.android.gms.ads.identifier.AdvertisingIdClient$Info {*;}

# Keep class SUEZ-X
-keep class com.suezx.ad.**{*;}

-keepattributes Signature, *Annotation*, InnerClasses

-keep class * extends java.util.ListResourceBundle {
    protected Object[][] getContents();
}
-keep public class com.google.android.gms.common.internal.safeparcel.SafeParcelable {
    public static final *** NULL;
}
-keepnames @com.google.android.gms.common.annotation.KeepName class *
-keepclassmembernames class * {
    @com.google.android.gms.common.annotation.KeepName *;
}
-keepnames class * implements android.os.Parcelable {
    public static final ** CREATOR;
}
-dontwarn com.google.android.gms.**
-keep class com.google.android.gms.**{*;}
-dontwarn android.support.v4.**

-keep class android.support.v4.**{*;}

```

# MoPub 배너 광고 적용 방법

MoPub 배너광고: <https://developers.mopub.com/docs/android/banner/>

## (1) Layout XML에 배너 광고를 위한 슬롯을 정의

MoPub SDK는 광고 요청 및 로드를 처리하는 custom 하위 클래스인 MoPubView를 제공합니다. 이 XML 블록을 Activity 또는 Fragment의 레이아웃에 포함시켜 시작하십시오.

```
<com.mopub.mobileads.MoPubView
    android:id="@+id/adview"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
/>
```

## (2) 배너 광고 슬롯에 광고 로딩

Activity 또는 Fragment 코드에서 MoPubView에 대한 인스턴스 변수를 선언하십시오.

```
private MoPubView moPubView;
```

Activity의 onCreate() 또는 Fragment의 onCreateView() 메소드에 TPMN에서 전달 받은 MoPubView의 Ad Unit ID 값을 설정한 다음 loadAd()를 호출하여 광고를 로드합니다.

```
...
moPubView = (MoPubView) findViewById(R.id.adview);
moPubView.setAdUnitId("xxxxxxxxxxx"); // Enter your Ad Unit ID
moPubView.loadAd();
...
```

Activity나 Fragment가 종료되면 다음을 호출하여 MoPubView도 종료해야 합니다.

```
moPubView.destroy();
```

## (3) Listener Interface

MoPubView 는 BannerAdListener 라는 listener interface를 제공합니다.

```
// Sent when the banner has successfully retrieved an ad.
public void onBannerLoaded(MoPubView banner);
```



```
// Sent when the banner has failed to retrieve an ad. You can use the
MoPubErrorCode value to diagnose the cause of failure.
public void onBannerFailed(MoPubView banner, MoPubErrorCode errorCode);

// Sent when the user has tapped on the banner.
public void onBannerClicked(MoPubView banner);

// Sent when the banner has just taken over the screen.
public void onBannerExpanded(MoPubView banner);

// Sent when an expanded banner has collapsed back to its original size.
public void onBannerCollapsed(MoPubView banner);
```

이벤트를 받기 위해 listener interface를 구현해야 합니다.

```
public class ExampleActivity extends Activity implements BannerAdListener {
    @Override
    public void onBannerLoaded(MoPubView banner) {
        Toast.makeText(getApplicationContext(),
            "Banner successfully loaded.", Toast.LENGTH_SHORT).show();
    }

    // ... other BannerAdListener methods ...
}
```

그리고 activity의 MoPubView에 listener를 생성하십시오.

```
moPubView.setBannerAdListener(this);
```

# MoPub 전면 광고 적용 방법

MoPub 전면광고: <https://developers.mopub.com/docs/android/interstitial/>

## (1) 전면 광고 생성

MoPub SDK는 전면 광고 요청 및 로드를 처리하는 custom 하위 클래스인 MoPubInterstitial를 제공합니다.

전면 광고를 표시하려는 activity에 MoPubInterstitial 인스턴스 변수를 선언합니다.

```
private MoPubInterstitial mInterstitial;
```

MoPubInterstitial 은 InterstitialAdListener 라는 이벤트 리스너 interface를 제공합니다.

Activity에 InterstitialAdListener 를 implement 합니다.

Activity의 onCreate() 메소드에 MoPubInterstitial 생성자에 TPMN에서 전달받은 Ad Unit ID를 입력합니다.

```
mInterstitial = new MoPubInterstitial(this, YOUR_INTERSTITIAL_AD_UNIT_ID_HERE);
```

그 다음 activity에 InterstitialAdListener를 등록합니다.

```
// Remember that "this" refers to your current activity.  
mInterstitial.setInterstitialAdListener(this);
```

## (2) 전면 광고 표시

onCreate() 메소드에서 load()를 호출하여 광고 캐싱을 시작하십시오. 삽입 광고 콘텐츠를 표시하려면 먼저 리치 미디어가 포함되어 있고 로드하는 데 시간이 걸릴 수 있으므로 삽입 광고 콘텐츠를 가져 오는 것이 중요합니다. 활동이 처음 생성 될 때 캐싱을 제안하지만 게임 레벨 시작과 같이 앱의 이벤트를 기반으로 수행하도록 선택할 수도 있습니다.

참고 : load() 메소드는 자바 스크립트, 이미지 및 추가 리소스를 비롯한 광고 콘텐츠를 요청하고 캐시하므로 몇 초가 걸릴 수 있습니다. show ()를 호출하기 전에 적절한 InterstitialAdListener 콜백을 기다리면서 충분한 시간을 제공하는 것이 가장 좋습니다.

광고를 게재할 준비가 되면 `InterstitialAdListener # onInststialLoaded()` 콜백과 `MoPubInterstitial.isReady ()`를 사용하여 삽입 광고가 성공적으로 캐시되었는지 확인하십시오.

`isReady()`가 `true`를 반환하면 `show ()` 메소드를 호출하여 삽입 광고를 표시합니다. 다른 삽입 광고 주기 메소드에 로직을 추가하여 사용자가 광고와 상호 작용할 때 어떤 일이 일어나는지 처리 할 수 있습니다

마지막으로 `Activity`의 `onDestroy()` 메소드에서 삽입 광고에 `destroy ()`를 호출하는 것을 잊지 마십시오.

### (3) 전면 광고 예시 코드

```
public class ExampleActivity extends Activity implements InterstitialAdListener {
    MoPubInterstitial mInterstitial;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        mInterstitial = new MoPubInterstitial(this,
YOUR_INTERSTITIAL_AD_UNIT_ID_HERE);
        mInterstitial.setInterstitialAdListener(this);
        mInterstitial.load();
    }

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

    // Defined by your application, indicating that you're ready to show an
    interstitial ad.
    void yourAppsShowInterstitialMethod() {
        if (mInterstitial.isReady()) {
            mInterstitial.show();
        } else {
            // Caching is likely already in progress if `isReady()`
is false.
            // Avoid calling `load()` here and instead rely on the
    callbacks as suggested below.
        }
    }

    // InterstitialAdListener methods
    @Override
    public void onInterstitialLoaded(MoPubInterstitial interstitial) {
        // The interstitial has been cached and is ready to be shown.
    }
}
```

```
@Override
public void onInterstitialFailed(MoPubInterstitial interstitial,
MoPubErrorCode errorCode) {
    // The interstitial has failed to load. Inspect errorCode for additional
information.
    // This is an excellent place to load more ads.
}

@Override
public void onInterstitialShown(MoPubInterstitial interstitial) {
    // The interstitial has been shown. Pause / save state accordingly.
}

@Override
public void onInterstitialClicked(MoPubInterstitial interstitial) {}

@Override
public void onInterstitialDismissed(MoPubInterstitial interstitial) {
    // The interstitial has being dismissed. Resume / load state accordingly.
    // This is an excellent place to load more ads.
}
}
```

## SUEZ-X Adapter 적용 방법(필수 사항)

MoPub을 통해 TPMN의 광고를 요청하기 위해서는 MoPub SDK와 TPMN의 SUEZ-X SDK 가 필요하며 이를 적용하기 위한 SUEZ-X Adapter를 넣어 주셔야 합니다.

GitHub : <https://github.com/tpmn/SUEZ-X/tree/master/SUEZ-X-mopub-custom-adapter/Android>

### 띠배너(320x50)

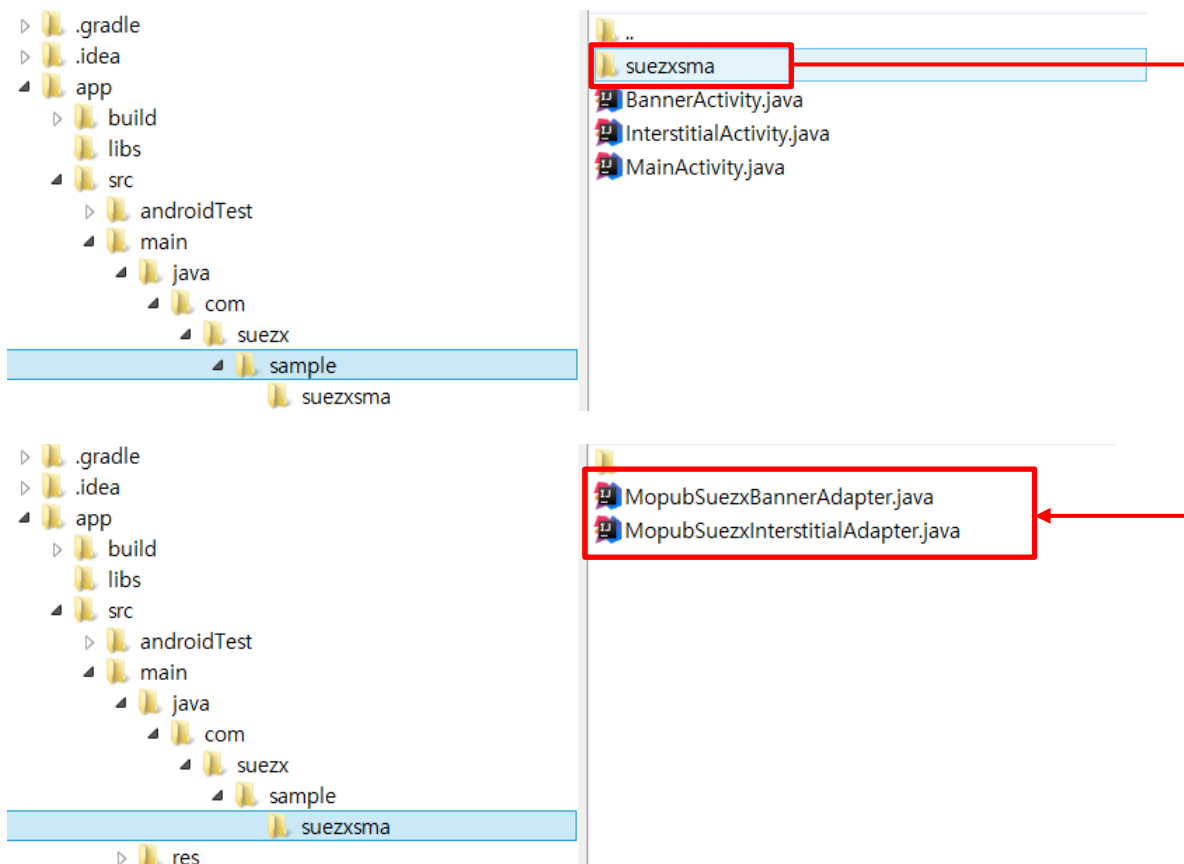
- MopubSuezxBannerAdapter.java: **SUEZ-X 배너 광고**

### 전면광고(320x480)

- MopubSuezxInterstitialAdapter.java: **SUEZ-X 전면 광고**

적용하고자 하는 광고 플랫폼의 Custom Event Class 파일을 [적용앱 패키지이름.suezxsm] 패키지에 복사하여 등록 합니다.

예) `com.example.sampleapp.suezxsm`에 적용할 Custom Event Class 파일 복사하여 등록



# AdMob 추가 방법(필수 사항)

MoPub SDK를 통하여 AdMob(Google) 광고를 집행하는 경우 참고하시기 바랍니다.

## (1)Manifest 설정 추가

AdMob(Google Play Service)를 사용하기 위해 <application>에 다음과 같은 태그를 확인하여 주시고 없다면 추가하십시오.

```
<meta-data android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
<activity android:name="com.google.android.gms.ads.AdActivity"
    android:configchanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenSize|smallestScreenSize">
```

## (2)프로젝트에 CustomEvent 파일 추가

AdMob CustomEvent : <https://github.com/mopub/mopub-android-mediation/tree/master/AdMob/com/mopub/mobileads>

MoPub SDK에서 AdMob 적용을 위한 CustomEvent 파일을 제공합니다.

Google AdMob (through Play Services)

- Banner ads ( `GooglePlayServicesBanner.java` )
- Interstitial ads ( `GooglePlayServicesInterstitial.java` )

프로젝트의 `src/com/mopub/mobileads` 디렉토리에 (해당 디렉토리 없는 경우 생성) 적용하고자 하는 광고 형태에 따라 MoPub에서 제공하는 위와 같은 CustomEvent 파일을 추가해주시기 바랍니다.

\* 위의 과정을 모두 마쳤을 경우 매체 계정 담당자(TPMN)에게 AdMob에서 발급받은 Ad Unit ID값을 전달해주시고 네트워크 설정을 요청해 주시기 바랍니다.

# FAN(Facebook Audience Network) 추가 방법

MoPub SDK를 통하여 Facebook 광고를 집행하고자 하는 경우에만 참고하시기 바랍니다.

## (1)프로젝트에 FAN SDK 추가

FAN SDK : <https://developers.facebook.com/docs/audience-network/download>

FAN SDK 다운로드 후 프로젝트 라이브러리에 FAN SDK를 추가하여 주시기 바랍니다.

## (2)프로젝트에 CustomEvent 파일 추가

FAN CustomEvent : <https://github.com/mopub/mopub-android-mediation/tree/master/FacebookAudienceNetwork/com/mopub/mobileads>

MoPub SDK에서 FAN 적용을 위한 CustomEvent 파일을 제공합니다.

### Facebook Audience Network

- Banner ads ( `FacebookBanner.java` )
- Interstitial ads ( `FacebookInterstitial.java` )

프로젝트의 `src/com/mopub/mobileads` 디렉토리에 (해당 디렉토리 없는 경우 생성) 적용하고자 하는 광고 형태에 따라 MoPub에서 제공하는 위와 같은 CustomEvent 파일을 추가해주시기 바랍니다.

\* 위의 과정을 모두 마쳤을 경우 매체 계정 담당자(TPMN)에게 Facebook에서 발급받은 Placement ID(노출 위치 ID)를 전달해주시고 Facebook 네트워크 설정을 요청해 주시기 바랍니다.

## MoPub을 통해 집행 가능한 Network 목록

Network	Network SDK	Network 호환 버전	지원하는 Ad Format
<b>Millennial Media</b>	iOS, Android	iOS: 6.3.1, Android: 6.3.0	Banner, Interstitial, Native
<b>AdMob(Google)</b>	iOS, Android	iOS: 7.21.0, Android: 11.0.2	Banner, Interstitial, Native, Rewarded Video
<b>Facebook Audience Network</b>	iOS, Android	iOS: 4.23, Android: 4.23	Banner, Interstitial, Native Static, Native Video, Rewarded Video
<b>AdColony</b>	iOS, Android	iOS: 3.2.1, Android: 3.2.1	Interstitial, Rewarded Video
<b>Chartboost</b>	iOS, Android	iOS: 6.6.3, Android: 6.6.3	Interstitial, Rewarded Video
<b>Tapjoy</b>	iOS, Android	iOS/Android: 11.10.0	Interstitial, Rewarded Video
<b>Yahoo! Gemini</b>	iOS, Android	iOS: 8.0.0, Android: 7.2.0	Banner, Interstitial, Native Static, Native Video
<b>Unity Ads</b>	iOS, Android	iOS: 2.1.0, Android: 2.1.0	Interstitial, Rewarded Video
<b>Vungle</b>	iOS, Android	iOS: 5.1.0, Android: 4.0.3	Interstitial, Rewarded Video

Unity 앱 적용 가능 (Native Ad 제외)

본 가이드에 기술된 Facebook Audience Network, AdMob을 제외한 네트워크의 광고를 사용하고 싶으신 경우 매체 계정 담당자(TPMN)에게 연락주시기 바랍니다.



## 주의 사항

\* Third-party network의 SDK나 Adapter를 추가하는 경우에 dex 에러가 발생하면 build.gradle 파일에 다음과 같은 설정을 추가하십시오.

```
defaultConfig {  
    multiDexEnable true  
}
```

\* Android API 13 미만의 버전에서 컴파일하는 경우 Manifest 항목에서 screenSize를 제외하여야 합니다.

\* MoPub SDK, SUEZ-X SDK, SUEZ-X Adapter 는 필수 사항으로 적용하여 주셔야 합니다.

\* SUEZ-X Adapter 적용시 Adapter 파일을 명시된 패키지명에 넣어 주셔야 합니다.

\* Facebook 광고를 사용하시는 경우 MoPub에서 제공하는 CustomEvent 파일 (FacebookBanner.java , FacebookInterstitial.java)을 명시된 위치에 넣어 주셔야 합니다.

\* AdMob 광고를 사용하시는 경우 MoPub에서 제공하는 CustomEvent 파일 (GooglePlayServicesBanner.java , GooglePlayServicesInterstitial.java)을 명시된 위치에 넣어 주셔야 합니다.

## Q&A

- MoPub, TPMN(SUEZX) 광고가 정상적으로 나오지 않을 때?
  - Logcat을 활용하여 광고가 정상적으로 나오지 않을 때 원인을 파악하실 수 있습니다.
  - MoPub Logcat 적용방법
    - ◆ **Banner Ad** - MoPub Banner 광고를 적용하는 Activity에 BannerAdListener를 implements 합니다. onBannerFailed(MoPubView banner, MoPubErrorCode errorCode) 를 Override 하여 배너 광고 실패 시 MoPubErrorCode를 Logcat에 출력할 수 있도록 설정해줍니다.
    - ◆ **Interstitial Ad(Fullscreen)** - MoPub의 전면광고를 적용하는 Activity에 InterstitialAdListener를 implements 합니다. onInterstitialFailed(MoPubInterstitial interstitial, MoPubErrorCode errorCode) 를 Override 하여 전면 광고 실패 시 MoPubErrorCode를 Logcat에 출력할 수 있도록 설정해줍니다.
  - TPMN(SUEZX) Logcat 적용방법
    - ◆ MopubSuezxBannerAdapter, MopubSuezxInterstitialAdapter에서 각 광고 이벤트 발생 시 **OnAdLoaded(), OnAdClicked(), OnAdClosed(), OnAdFailed(SuezxAdError adError, String s)** 이벤트 리스너를 통해 기본적으로 "SUEZ-X" 태그로 Log를 출력해줍니다.
  - MoPubErrorCode 정보
    - ◆ NO\_FILL("No ads found.")
      - 해당 로그가 지속적으로 발생할 경우, 매체 계정 담당자(TPMN)에게 문의하여 주시기 바랍니다.
    - ◆ WARMUP("Ad unit is warming up. Try again in a few minutes.")
    - ◆ SERVER\_ERROR("Unable to connect to MoPub adserver.")
    - ◆ INTERNAL\_ERROR("Unable to serve ad due to invalid internal state.")
    - ◆ CANCELLED("Ad request was cancelled.")
    - ◆ NO\_CONNECTION("No internet connection detected.")
    - ◆ ADAPTER\_NOT\_FOUND("Unable to find Native Network or Custom Event adapter.")
      - 해당 로그 발생시 Adapter파일이 올바른 경로에 들어가 있는지 확인하여 주시기 바랍니다.

니다.

- ◆ `ADAPTER_CONFIGURATION_ERROR`("Native Network or Custom Event adapter was configured incorrectly.")
  - 해당 로그 발생시 매체 계정 담당자(TPMN)측에 문의하여 주시기 바랍니다.
- ◆ `NETWORK_TIMEOUT`("Third-party network failed to respond in a timely manner.")
- ◆ `NETWORK_NO_FILL`("Third-party network failed to provide an ad.")
- ◆ `NETWORK_INVALID_STATE`("Third-party network failed due to invalid internal state.")
- ◆ `MRAID_LOAD_ERROR`("Error loading MRAID ad.")
- ◆ `VIDEO_CACHE_ERROR`("Error creating a cache to store downloaded videos.")
- ◆ `VIDEO_DOWNLOAD_ERROR`("Error downloading video.")
- ◆ `VIDEO_NOT_AVAILABLE`("No video loaded for ad unit.")
- ◆ `VIDEO_PLAYBACK_ERROR`("Error playing a video.")
- ◆ `UNSPECIFIED`("Unspecified error.")

#### ■ SuezxAdError 에러 코드 정보

- ◆ `AD_DOWNLOAD_ERROR_HTTPFAILED`
  - 해당 로그 발생시 Internet Connection 상태를 확인하여 주시기 바랍니다.
- ◆ `AD_DOWNLOAD_ERROR_INVALIDIDAD`("Check Your InventoryId or PublisherID!")
  - 해당 로그 발생시 매체 계정 담당자(TPMN) 측에 문의하여 주시기 바랍니다.
- ◆ `AD_DOWNLOAD_ERROR_PERMISSION_DENIED`("Ensure that you add the INTERNET, ACCESS\_NETWORK\_STATE and ACCESS\_WIFI\_STATE permissions in your AndroidManifest.xml")
  - 해당 로그 발생시 Android Manifest의 권한설정 부분을 확인하여 주시기 바랍니다.

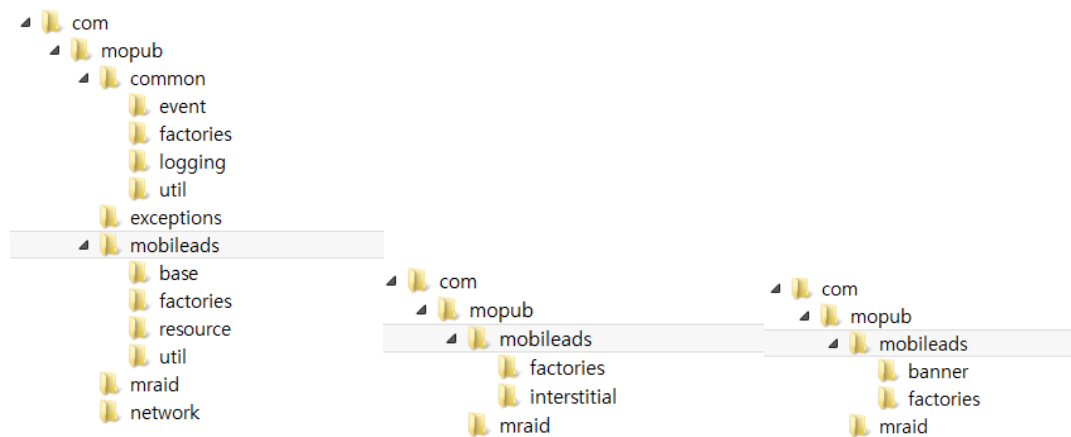
## Issue Case

MoPub SDK 적용 과정에서 다음과 같은 오류가 있을 시 아래의 적용 과정을 진행해 주시기 바랍니다.

### Case 1:

MoPub SDK 라이브러리의 패키지경로 명에 맞게 import 선언을 했는데 `ClassNotFoundException` 에러가 뜨는 경우

→ MoPub SDK의 경우 패키지 내부 디렉토리명이 각 라이브러리 모듈별로 동일하게 되어 있습니다.



MoPub base-sdk

MoPub banner-sdk

MoPub interstitial-sdk

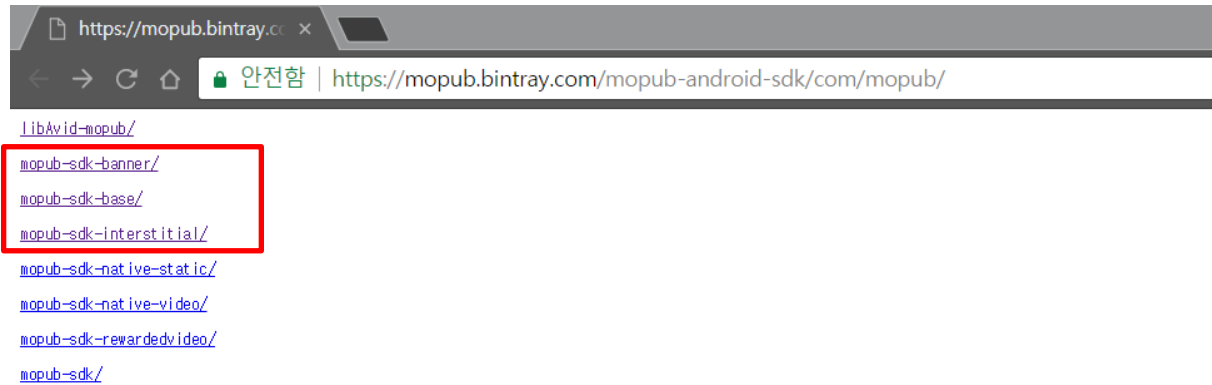
→ 이러한 이유로 특정 클래스를 추가하려는 상황에서 패키지 명이 겹쳐서 엉뚱한 sdk에서 클래스를 찾게되어 `ClassNotFoundException` 에러가 발생하게 됩니다.

→ (대부분의 개발환경에서는 4~12페이지에 가이드 되어있는 방법으로 정상적인 적용이 가능하지만 특정 이클립스 버전에서 이와 같은 문제가 발생하는 것으로 추측됩니다.)

→ 위와 같은 이슈로 적용에 문제가 있는 경우 다음과 같이 적용하여 주시기 바랍니다.

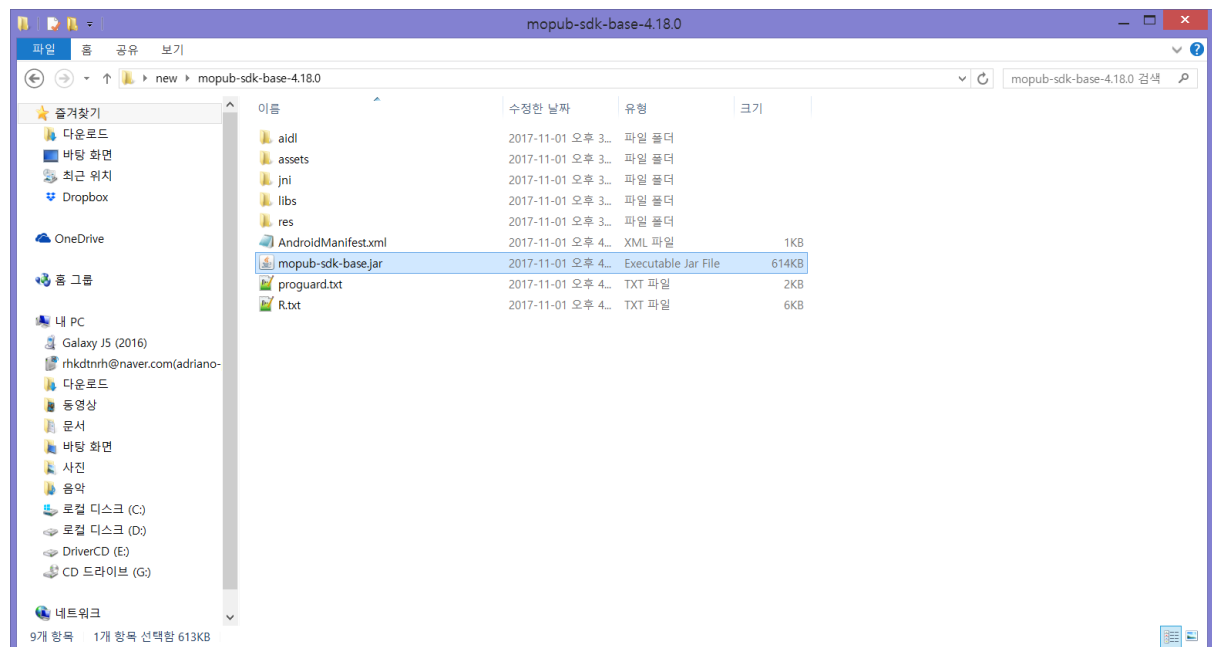
MoPub SDK(AAR) : <https://mopub.bintray.com/mopub-android-sdk/com/mopub/>

1. 위의 링크에서 **mopub-sdk-base**와 사용하시는 광고 상품(Banner, Interstitial)에 알맞은 디렉토리를 선택하고 최신 버전(가이드 작성일 기준 4.18 버전)의 MoPub SDK AAR 파일들을 다운로드 합니다.



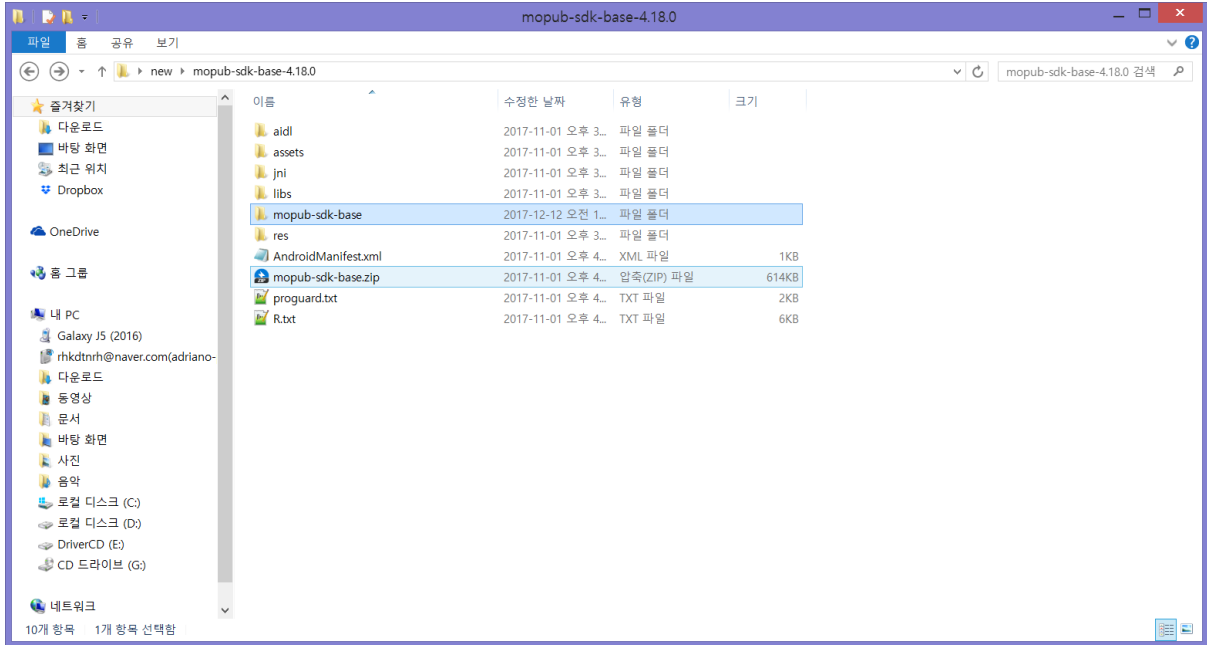
2. 다운로드한 AAR 파일의 확장자를 **.zip**으로 바꾸어 압축을 해제한 후 **classes.jar** 파일을 알맞은 모듈 명으로 바꾸어 줍니다.

예) mopub-sdk-base AAR파일의 classes.jar 파일명을 mopub-sdk-base.jar 으로 바꾸어 줌

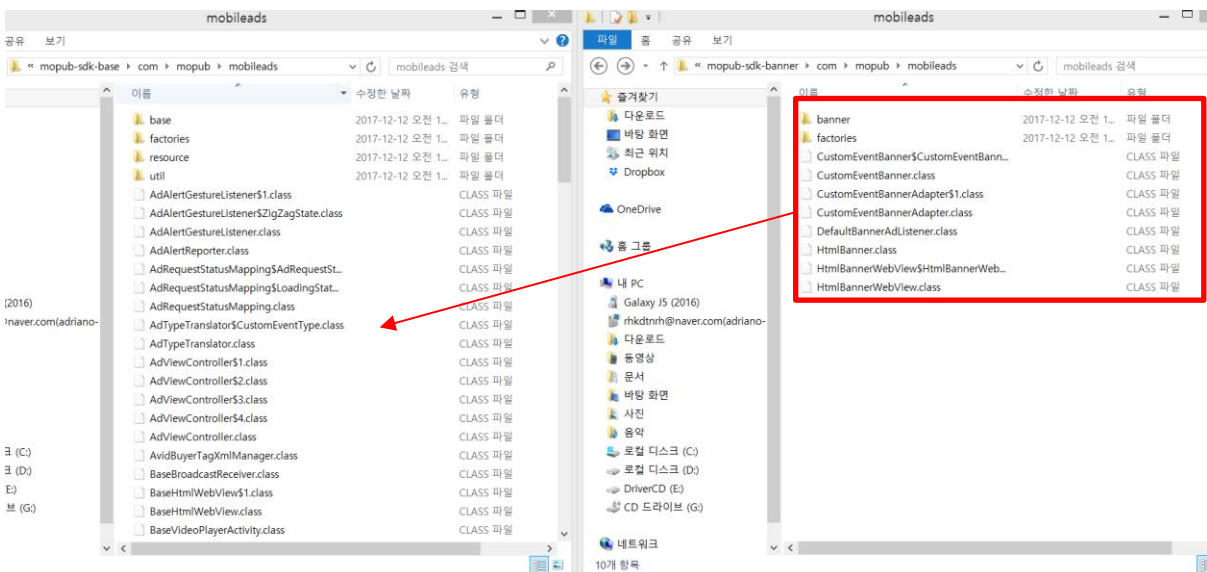


위와 같은 방법으로 mopub-sdk-base와 사용하실 상품에 따라 mopub-sdk-banner, mopub-sdk-interstitial의 jar 파일을 만들어줍니다.

3. 2.에서 만든 mopub-sdk-base.jar 파일을 .zip으로 확장자를 바꾸어 준 후 압축을 해제하여 줍니다. 마찬가지로 mopub-sdk-banner.jar, mopub-sdk-interstitial.jar 의 확장자를 .zip으로 바꾸어 준 후 압축을 해제합니다.



4. 3.에서 만들어진 mopub-sdk-base 폴더안에 압축 해제 후 생성된 mopub-sdk-banner, mopub-sdk-interstitial 폴더내부의 클래스 파일을 경로에 맞게 옮겨줍니다. (모든 일치하는 경로에 있는 class 파일을 mopub-sdk-base로 클래스 경로에 맞게 옮겨주어야 합니다. com.mopub.mobileads, com.mopub.mobileads.factories, com.mopub.mraid 등)



5. 사용하려는 모든 MoPub 라이브러리 모듈의 클래스파일을 mopub-sdk-base로 옮긴 후 CMD를 이용하여 mopub-sdk-base 폴더로 이동합니다. 그 후 jar 명령을 사용해서 폴더내의 클래스파일들을 jar파일로 압축하여 줍니다.

jar cvf mopub-sdk.jar . 주의 : mopub-sdk.jar 뒤의 . 을 붙여야합니다.

```

관리자: 명령 프롬프트
2017-11-01 오후 04:19 5,705 R.txt
2017-11-01 오후 03:30 <DIR> res
4개 파일 636,273 바이트
8개 디렉터리 150,218,051,584 바이트 남음

C:\Users\Admin\Desktop\new\mopub-sdk-base-4.18.0>cd mopub-sdk-base
C:\Users\Admin\Desktop\new\mopub-sdk-base-4.18.0\mopub-sdk-base>jar cvf mopub-sdk.jar .
  
```

```

관리자: 명령 프롬프트
추가하는 중: com/mopub/network/RequestQueueHttpStack.class(입력 = 2345) (출력 = 1007)(57% 감소함)
추가하는 중: com/mopub/network/ScribeBackoffPolicy.class(입력 = 1664) (출력 = 962)(42% 감소함)
추가하는 중: com/mopub/network/ScribeRequest$Listener.class(입력 = 342) (출력 = 215)(37% 감소함)
추가하는 중: com/mopub/network/ScribeRequest$ScribeRequestFactory.class(입력 = 524) (출력 = 244)(53% 감소함)
추가하는 중: com/mopub/network/ScribeRequest.class(입력 = 3799) (출력 = 1465)(61% 감소함)
추가하는 중: com/mopub/network/ScribeRequestManager$1.class(입력 = 702) (출력 = 385)(45% 감소함)
추가하는 중: com/mopub/network/ScribeRequestManager$2.class(입력 = 1468) (출력 = 739)(49% 감소함)
추가하는 중: com/mopub/network/ScribeRequestManager.class(입력 = 2026) (출력 = 843)(58% 감소함)
추가하는 중: com/mopub/network/TrackingRequest$1.class(입력 = 1752) (출력 = 844)(51% 감소함)
추가하는 중: com/mopub/network/TrackingRequest$Listener.class(입력 = 457) (출력 = 287)(37% 감소함)
추가하는 중: com/mopub/network/TrackingRequest.class(입력 = 8041) (출력 = 2986)(62% 감소함)

C:\Users\Admin\Desktop\new\mopub-sdk-base-4.18.0\mopub-sdk-base>
  
```

6. mopub-sdk-base 폴더 내부에 생성된 mopub-sdk.jar 파일을 프로젝트에 라이브러리로 적용하여 줍니다.

## Case 2:

Caused by: **java.lang.NoSuchMethodError: No static method `checkSelfPermission(Landroid/content/Context;Ljava/lang/String;)I` in class `Landroid/support/v4/content/ContextCompat`; or its super classes (declaration of `'android.support.v4.content.ContextCompat'` appears in `/data/app/-----.apk`)**

위와 같은 에러가 발생하는 경우 적용중인 android.support.v4 라이브러리의 교체가 필요합니다.

Github 링크 : [https://github.com/tpmn/SUEZ-X/tree/master/SUEZ-X-mopub-custom-adapter/Android/for\\_eclipse](https://github.com/tpmn/SUEZ-X/tree/master/SUEZ-X-mopub-custom-adapter/Android/for_eclipse)

위의 링크에서 android.support.v4 라이브러리를 다운받아 교체해주시기 바랍니다.