



AD@M-ANDROID PUBLISHER SDK GUIDE

이 문서는 DAUM 신디케이션 제휴 당사자에 한해 제공되는 자료로 가이드 라인을 포함한 모든 자료의 지적재산권은 주식회사 다음커뮤니케이션이 보유하고 있습니다.

COPYRIGHT© DAUM COMMUNICATIONS. ALL RIGHTS RESERVED.

목 차

변경이력	4
Ad@m 광고 삽입 방법	8
Ad@m SDK 구성	8
1 단계 : client ID 발급받기	8
2 단계 : 라이브러리 import	8
3 단계 : AndroidManifest.xml 설정	9
4 단계 : 광고 요청을 위한 UI 구성 및 설정	10
4-a. Xml 방식	10
4-b. Java 방식	13
선택 : Interstitial (전면형) 광고 요청을 위한 설정	15
Class reference	16
Package net.daum.adam.publisher	16
Class Summary	16
Interface Summary	16
Enum Summary	16
Class AdInfo	17
Constructor Summary	17
Method Summary	17
Class AdView	18
Constructor Summary	18
Method Summary	18
Enum AdView.AnimationType	21
Constant Summary	21
Interface AdView.OnAdClickedListener	22
Method Summary	22
Interface AdView.OnAdClosedListener	22
Method Summary	22
Interface AdView.OnAdFailedListener	23

Method Summary	23
Interface AdView.OnAdLoadedListener.....	23
Method Summary	23
Interface AdView.OnAdWillLoadListener	23
Method Summary	23
Class AdInterstitial	24
Constructor Summary	24
Method Summary	24
Enum AdError	25
Constant Summary	25
추가 정보(FAQ).....	26
Q1. 광고 수신에 되지 않을 때는 어떻게 하나요?.....	26
Q2. 인터넷(3G 또는 WIFI)이 연결되지 않을 경우에 어떻게 하나요?.....	27
Q3. 광고 영역이 텅 비어보입니다. 아담 버그 아닌가요?	27
Q4. 시스템 앱에서 Expandable 광고가 보이지 않습니다. 왜 그런건가요?.....	27

VER 2.2.3.3

이 가이드는 Android Application 에 모바일 광고를 노출하기 위한 광고 데이터요청과 처리 방법을 설명한다.

사이트/앱 운영정책에 어긋나는 경우 적립금 지급이 거절 될 수 있으니 유의하시기 바랍니다.

- 문의 고객센터 <http://cs.daum.net/mail/form/256.html>
- 사이트/앱 운영 정책 http://mobile.biz.daum.net/guide/guide_siteapp_policy.jsp

변경이력

- **Ver2.2.3.3**
 - ✓ UDID 키 생성 로직 개선
 - ✓ 일부 기기에서 광고를 제대로 받아오지 못하는 버그 수정
- **Ver2.2.3**
 - ✓ 광고 효율 관련 로직 개선
 - ✓ 필수 권한 대상에서 android.permission.WRITE_EXTERNAL_STORAGE 제외
- **Ver2.2.2**
 - ✓ BroadcastReceiver Leak 현상 수정
 - ✓ 일부 기기에서 리치미디어 상품 클릭시 발생하는 NullPointerException 문제 수정
- **Ver2.2.1**
 - ✓ 앱에서 광고 업데이트 시에 간헐적으로 발생하는 NullPointerException 문제 수정
 - ✓ 광고 레이아웃 높이와 관련된 기능 수정 및 관련 API 추가
 - setAdUnitSize
 - ✓ Android SDK 2.1(Level 7) 지원
- **Ver2.2**
 - ✓ 리치미디어 노출 관련 추가 기능 탑재
 - ✓ 필수 권한으로 android.permission.WRITE_EXTERNAL_STORAGE 추가
 - ✓ Android SDK 지원 레벨을 2.1(Level 7)에서 2.2(Level 8)로 변경
 - ✓ AdInterstitial 사용시 가이드 수정
- **Ver2.1**
 - ✓ 앱 사용자의 성별 및 생일 정보 설정을 위한 AdInfo Class 추가
 - ✓ 광고 전환 애니메이션 AnimationType.FADE, AnimationType.SLIDE 추가
 - ✓ 광고 요청시에 광고 영역이 보이지 않으면 명시적으로 adFailed() 호출
 - ✓ 앱 바로가기 사용시에 Interstitial 에서 발생하는 문제점 수정
 - ✓ WebView 에서 Scroll 시에 발생하는 OutOfMemoryError 수정
 - ✓ 추가된 API
 - AdInfo Class
 - void setGender(String gender)
 - String getGender()
 - void setBirth(String date)
 - String getBirth()
 - AdView Class
 - void setAdInfo(AdInfo info)
 - AdInfo getAdInfo()
 - AdInterstitial Class

- void setAdInfo(AdInfo info)
 - AdInfo getAdInfo()
- **Ver2.0.4**
 - ✓ 최초 광고 뷰가 INVISIBLE 일 때 광고가 내려오지 않는 부분 수정
 - ✓ 광고를 불러오는 과정에서 발생하는 ANR 문제 수정
 - ✓ 전면형 광고의 닫기 버튼에 대한 리스너 추가
 - AdInterstitial Class
 - void setOnAdClosedListener(AdView.OnAdClosedListener listener)
- **Ver2.0.3**
 - ✓ 화면에서 뷰가 해제될 때, 일부 BroadcastReceiver leak 문제 해결
- **Ver2.0.2**
 - ✓ 포커스 이동 버그 수정
 - ✓ 전면형 광고 노출시 닫기 버튼 위치 수정
 - ✓ 광고 영역 크기가 320DP x 48DP 보다 작을 때 광고 갱신 중단
- **Ver2.0.1**
 - ✓ 광고 무한 갱신 및 깜빡임 버그 수정
 - ✓ WIFI 에서 3G 로 네트워크 전환시 광고 수신 중단되는 버그 수정
 - ✓ WebView 에서 발생하는 Memory Leak 수정
 - ✓ 추가된 API
 - AdView Class
 - void setThreadPriority()
 - int getThreadPriority(int priority)
- **Ver2.0**
 - ✓ **ACCESS_NETWORK_STATE** 권한을 필수 권한 추가
 - ✓ 전면형 광고(AdInterstitial class) 추가
 - ✓ **광고 사용방법(API) 변경**
 - ✓ 일부 클래스 변경 및 제거
 - MobileAdView class 를 AdView class 로 변경하고 API 수정
 - AdHttpListener class 와 AdConfig class 제거
- **Ver1.4.2**
 - ✓ 앱에서 웹뷰 광고 노출시에 간헐적으로 발생하는 에러 수정
- **Ver1.4.1.1**
 - ✓ 앱에서 Proguard 적용시 SDK 와 관련해 발생하는 에러 수정
- **Ver1.4.1**
 - ✓ 기존 AsyncTask 에서 발생하는 NullPointerException 수정
 - ✓ 특정 광고 노출시 AdWebView 에서 발생하는 NullPointerException 수정
 - ✓ 네트워크 상태에 따라 간헐적으로 발생하는 ANR 현상 수정
 - ✓ 광고 영역을 명시적으로 삭제될 경우, 내부적으로 광고 관련 데이터를 제거함

- **Ver1.4.0.1**
 - ✓ SDK 에서 요구하는 필수권한이 있음에도 특정 상황 시 권한 요청 메시지가 출력되는 현상 수정
- **Ver1.4.0**
 - ✓ 다양한 광고 Targeting 기능 추가
 - ✓ INTERNET 과 ACCESS_WIFI_STATE 권한을 필수 권한으로 변경
 - ✓ keyboardView 에 광고 노출시 클릭 작동 안되는 현상 수정
 - ✓ 특정 상황에서 UserAgent 를 조회할 수 없을 때 발생하는 문제점 수정
- **Ver1.3.2**
 - ✓ Android 3.0(honeycomb) 이상 버전에서 Ad@m 클릭스 한글 깨짐 현상 수정
- **Ver1.3.1**
 - ✓ AsyncTask 사용 안함
- **ver1.3.0**
 - ✓ 광고 삽입 방식을 보다 간편한 방식으로 변경
 - Ver1.2 이하 방식과 일부 호환되지 않으니 sample 코드를 참고하여 수정 필요
 - ✓ 다양한 가로 사이즈에서 배너 이미지 변형 없이 광고 노출 지원
 - ✓ 제거된 함수들
 - **MobileAdView class**
 - void setEnabled(Boolean enabled)
 - void refreshFreshAd()
 - void setAdTextColor(String textcolor)
 - String getAdTextColor()
 - void setBackgroundColor(String in_backcolor)
 - String getBackgroundColor()
 - **AdConfig class**
 - void setGender(String in_gender)
 - String getGender()
 - void setBirthday(int year, int month, int day)
 - void setBirthday(GregorianCalendar in_calendar)
 - GregorianCalendar getBirthday()
 - void setAllowUseOfLocation(Boolean in_userLocation)
 - ✓ 추가된 함수들
 - **MobileAdView class**
 - void pause()
 - void resume()
 - void destroy()
 - ✓ 광고 갱신 방식 변경
 - 광고 view 가 화면에 보이지 않는 경우에는 광고 갱신이 자동 중단됨

- 기존에 `setAdListener(null)`이나 `setEnabled(false)` 함수 호출을 통해 광고 갱신을 중단하는 방식은 더 이상 지원되지 않음
 - 수동으로 광고 갱신을 제어하고 싶은 경우에는 `pause`, `resume`, `destroy` 함수를 참조
- ✓ 위치정보 등 수집 항목 제거(location, gender, birthday)



AD@M 광고 삽입 방법

AD@M SDK 구성

- ✓ Ad@MPublisherSDK.jar : Ad@m 광고를 삽입해주는 라이브러리
- ✓ Sample/AdamSample/src/net/daum/adam/publisher/sample/
 - BannerTypeXML1.java : 광고를 xml 로 붙인 샘플
 - BannerTypeXML2.java : 광고 Visible 처리 및 pause, resume 처리 예시 샘플
 - BannerTypeJava.java : 광고를 java 코드로 붙인 샘플
 - InterstitialActivity.java : Interstitial(전면형) 광고를 java 코드로 붙인 샘플

1 단계 : CLIENT ID 발급받기

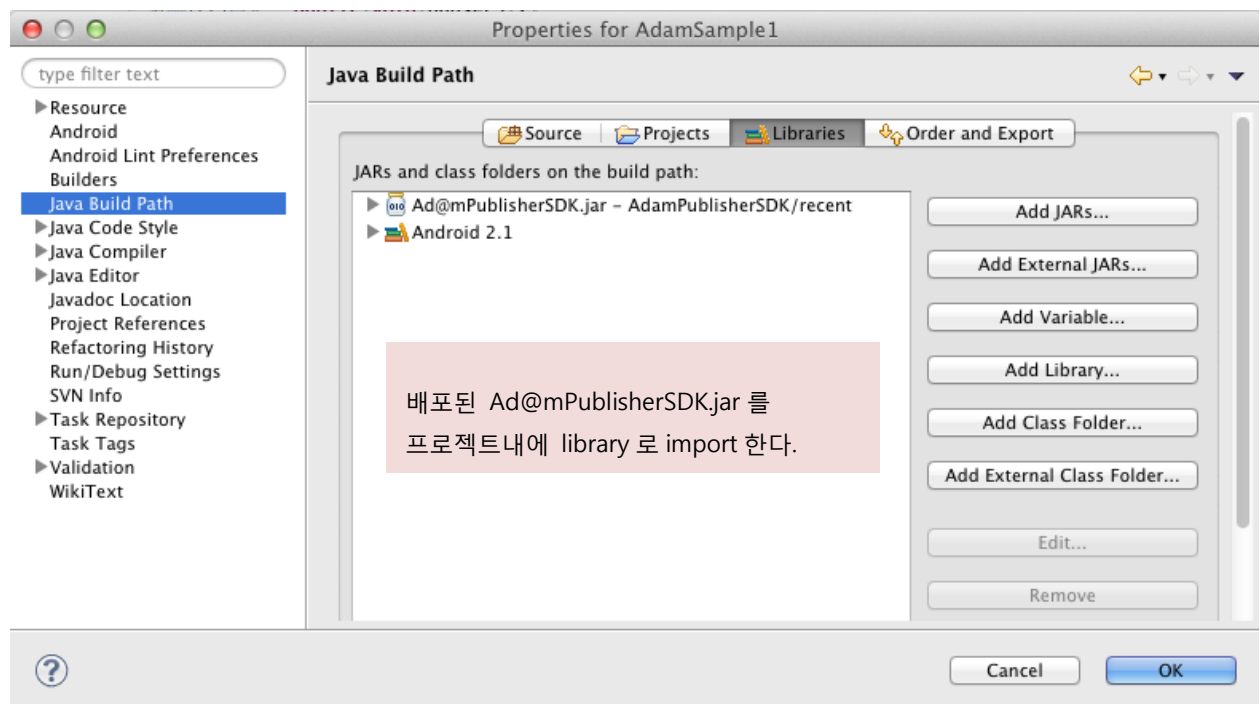
실제 광고를 다운로드 받고, 수익창출을 위해서 mobile.biz.daum.net 에서 사이트/앱 등록 후 client ID 를 발급받아야 한다. 아래 URL 을 통해 애플리케이션을 등록할 수 있다.

http://mobile.biz.daum.net/guide/guide_siteapp1.jsp

2 단계 : 라이브러리 IMPORT

Ad@MPublisherSDK 를 프로젝트 내에 라이브러리로 Import 한다.

(**Android 2.1(API Level 7) 이상**의 환경에서 동작한다.)



3 단계 : ANDROIDMANIFEST.XML 설정

- 아래 세 가지 필수 권한을 AndroidManifest.xml 에 추가한다.

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

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

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

SDK 2.2.0 부터는 최소한의 권한으로 INTERNET, ACCESS_WIFI_STATE, ACCESS_NETWORK_STATE 권한을 설정해야 한다.

(필수 권한 미 설정시 정상적 광고 노출 되지 않음)

- 광고를 넣을 Activity 에 반드시 **android:configChanges="orientation"** 을 설정해준다.
(Application OS Target 이 3.2(Level 13) 이상인 경우에는 추가로 **screenSize** 설정 필요)
- Interstitial(전면형) 광고를 추가하기 위해서는 반드시 아래 명시된 Activity 를 추가해야 한다.

```
<application
    android:icon="@drawable/icon"
    android:label="@string/appName" >
    <activity
        android:name=".TestAppActivity"
        android:configChanges="orientation|keyboardHidden"
        android:label="@string/appName" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
```

Interstitial 광고를 사용하기 위해서는
반드시 이 부분을 추가해야한다.

```
    <activity
        android:name="net.daum.adam.publisher.impl.AdInterstitialActivity"
        android:configChanges="orientation|screenSize"
        android:screenOrientation="portrait" >
    </activity>
```

광고를 노출할 Activity 에
android:configChanges="orientation|screenSize"을 반드시 추가해야 한다.

```
    <activity
        android:name=".BannerActivity"
        android:configChanges="orientation|screenSize" />
</application>
```

아래 네 가지 권한을 반드시 추가해야
한다.

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

AndroidManifest.xml

4 단계 : 광고 요청을 위한 UI 구성 및 설정

4-A. XML 방식

Layout 의 main.xml 에서 광고가 노출되고자 하는 곳에 AdView 객체를 추가한다.

광고를 노출 가능한 최소크기(320x48)보다 작게 광고 뷰가 할당되는 경우에는 광고가 노출되지 않을 수 있다. 그 이외의 속성 값은 어플리케이션의 특성에 따라 자유롭게 변경 가능하다.

```
<RelativeLayout
    xmlns:app="http://schemas.android.com/apk/res/[APP_PACKAGE_NAME]"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <net.daum.adam.publisher.AdView
        android:id="@+id/adview"
        android:visibility="visible"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        clientId="TestClientId"
        requestInterval="60"/>
</RelativeLayout>
```

광고를 사용하기 위해서는 반드시 Client ID 를 발급받아 사용해야 한다.

res/layout/main.xml

위 레이아웃에 설정한 AdView 객체를 Activity 에서 사용하는 방법을 아래 예를 통해 살펴보도록 하자. SDK 2.0 부터는 AdHttpListener 를 반드시 구현할 필요가 없고, 필요한 경우에 해당 Listener 를 구현해서 설정해주면 된다. 현재 5 개의 Listener 를 지원하고 있으며, 자세한 내역은 아래 예제 코드와 Class Reference 를 통해 살펴보도록 하자.

```
public class BannerTypeXML1 extends Activity {
    private static final String LOGTAG = "BannerTypeXML1";

    private AdView adView = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.adam_sample_1);
        initAdam();
    }

    @Override
    public void onDestroy() {
        super.onDestroy();

        if (adView != null) {
            adView.destroy();
            adView = null;
        }
    }

    private void initAdam() {
```

```

// Ad@m sdk 초기화 시작
adView = (AdView) findViewById(R.id.adview);

// 광고 리스너 설정
// 1. 광고 클릭시 실행할 리스너
adView.setOnAdClickedListener(new OnAdClickedListener() {
    @Override
    public void OnAdClicked() {
        Log.i(LOGTAG, "광고를 클릭했습니다.");
    }
});

// 2. 광고 내려받기 실패했을 경우에 실행할 리스너
adView.setOnAdFailedListener(new OnAdFailedListener() {
    @Override
    public void OnAdFailed(AdError error, String message) {
        Log.w(LOGTAG, message);
    }
});

// 3. 광고를 정상적으로 내려받았을 경우에 실행할 리스너
adView.setOnAdLoadedListener(new OnAdLoadedListener() {
    @Override
    public void OnAdLoaded() {
        Log.i(LOGTAG, "광고가 정상적으로 로딩되었습니다.");
    }
});

// 4. 광고를 불러올때 실행할 리스너
adView.setOnAdWillLoadListener(new OnAdWillLoadListener() {
    @Override
    public void OnAdWillLoad(String url) {
        Log.i(LOGTAG, "광고를 불러옵니다. : " + url);
    }
});

// 5. 전면형 광고를 닫았을때 실행할 리스너
adView.setOnAdClosedListener(new OnAdClosedListener() {
    @Override
    public void OnAdClosed() {
        Log.i(LOGTAG, "광고를 닫았습니다.");
    }
});
// 할당 받은 clientId 설정
// adView.setClientId("TestClientId");

// 광고 갱신 주기를 12초로 설정
// adView.setRequestInterval(12);

// 광고 영역에 캐시 사용 여부 : 기본 값은 true
adView.setAdCache(false);

```

Client ID 는 XML 에서 설정할 수도 있고,
직접 코드로 설정할 수도 있다.

```
// Animation 효과 : 기본 값은 AnimationType.NONE
adView.setAnimationType(AnimationType.FLIP_HORIZONTAL);
adView.setVisibility(View.VISIBLE);
    }
}
```

[YourApplication]Activity.java

광고 영역은 웹뷰를 사용하고 있고, 기본적으로 캐시를 사용하고 있다. 만약 캐시를 사용하지 않을 경우에는 위 예제와 같이 `adView.setAdCache(false);` 를 호출해 캐시를 사용하지 않도록 설정할 수 있다. 이 경우에는 기존에 캐시 영역의 데이터를 모두 삭제한다.

AdView 클래스에는 위와 같이 5 개의 리스너를 제공하고 있다.

- `AdView.OnAdClickedListener` : 광고 클릭할 경우 실행할 리스너
- `AdView.OnAdFailedListener` : 광고 내려받기 실패할 경우 실행할 리스너
- `AdView.OnAdLoadedListener` : 광고가 내려받았을 경우 실행할 리스너
- `AdView.OnAdWillLoadListener` : 광고를 불러오기 전에 실행할 리스너
- `AdView.OnAdClosedListener` : 전면형 광고를 닫을 때 실행할 리스너

위 예제에서는 현재 5 개의 리스너를 설정하고 있지만, 리스너가 필요가 없으면 굳이 설정하지 않아도 된다. 리스너와 관련된 자세한 내역은 클래스 레퍼런스를 통해 살펴보도록 하자.

4-B. JAVA 방식

광고를 넣고자 하는 view 가 들어 있는 Activity 가 생성될 때 AdView 객체를 생성하고 광고 요청을 위해 광고 View 에 필요한 리스너와 할당 받은 ClientId 를 설정 한다. XML 레이아웃을 이용해 광고 생성할 때와 거의 동일하다.

```
public class BannerTypeJava extends Activity {
    private static final String LOGTAG = "BannerTypeJava";
    private RelativeLayout relativeLayout = null;
    private AdView adView = null;

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

        relativeLayout = new RelativeLayout(this);

        // Ad@M 광고 뷰 생성 및 설정
        adView = new AdView(this);

        // 광고 클릭시 실행할 리스너
        adView.setOnAdClickedListener(new OnAdClickedListener() {
            @Override
            public void OnAdClicked() {
                Log.i(LOGTAG, "광고를 클릭했습니다.");
            }
        });

        // 광고 내려받기 실패했을 경우에 실행할 리스너
        adView.setOnAdFailedListener(new OnAdFailedListener() {
            @Override
            public void OnAdFailed(AdError arg0, String arg1) {
                Log.w(LOGTAG, arg1);
            }
        });

        // 광고를 정상적으로 내려받았을 경우에 실행할 리스너
        adView.setOnAdLoadedListener(new OnAdLoadedListener() {
            @Override
            public void OnAdLoaded() {
                Log.i(LOGTAG, "광고가 정상적으로 로딩되었습니다.");
            }
        });

        // 광고를 불러올때 실행할 리스너
        adView.setOnAdWillLoadListener(new OnAdWillLoadListener() {
            @Override
            public void OnAdWillLoad(String arg1) {
                Log.i(LOGTAG, "광고를 불러옵니다. : " + arg1);
            }
        });
    }
}
```

```

// 광고를 닫았을때 실행할 리스너
adView.setOnAdClosedListener(new OnAdClosedListener() {

    @Override
    public void OnAdClosed() {
        Log.i(LOGTAG, "광고를 닫았습니다.");
    }
});

// 할당 받은 clientId 설정
adView.setClientId("TestClientId");

// 광고 갱신 시간 : 기본 60초
adView.setRequestInterval(12);

// Animation 효과 : 기본 값은 AnimationType.NONE
adView.setAnimationType(AnimationType.FLIP_HORIZONTAL);

adView.setVisibility(View.VISIBLE);

// XML상에 android:layout_alignParentBottom="true" 와 같은 역할을 함
RelativeLayout.LayoutParams params = new
RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.MATCH_PARENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
params.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);

// 위에서 만든 레이아웃을 광고 뷰에 적용함.
adView.setLayoutParams(params);

setContentView(relativeLayout);
}

@Override
public void onDestroy() {
    super.onDestroy();

    if (adView != null) {
        adView.destroy();
        adView = null;
    }
}
}

```

광고를 사용하기 위해서는 반드시 Client ID 를 발급받아 사용해야 한다.

[YourApplication]Activity.java

선택 : INTERSTITIAL (전면형) 광고 요청을 위한 설정

Interstitial(전면형) 광고는 당분간 Ad@m의 네트워크 파트너를 대상으로 노출됩니다. Ad@m의 네트워크 파트너가 아닐 경우에도 Expandable(확장형), Animated Banner (애니메이션형)형의 Rich Media 광고가 노출됩니다.

Interstitial(전면형) 광고를 넣고자 하는 Activity가 생성될 때 AdInterstitial 객체를 생성하고 광고 요청을 위해 필요한 리스너와 할당 받은 ClientId를 설정한다. 이때, 반드시 3 단계에 명시한 XML 코드를 AndroidManifest.xml에 반드시 추가해야 한다.

```
public class InterstitialActivity extends Activity {
    /** 전면형 광고 선언 */
    AdInterstitial mAdInterstitial = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // 1. 전면형 광고 객체 생성
        mAdInterstitial = new AdInterstitial(this);
        // 2. 전면형 광고 클라이언트 ID를 설정한다.
        mAdInterstitial.setClientId("InterstitialTestClientId");
        // 3. (선택)전면형 광고 다운로드시에 실행할 리스너
        mAdInterstitial.setOnAdLoadedListener(new OnAdLoadedListener() {
            @Override
            public void OnAdLoaded() {
                Log.i("InterstitialTab", "광고가 로딩되었습니다.");
            }
        });
        // 4. (선택)전면형 광고 다운로드 실패시에 실행할 리스너
        mAdInterstitial.setOnAdFailedListener(new OnAdFailedListener() {
            @Override
            public void OnAdFailed(AdError error, String errorMessage) {
                Toast.makeText(InterstitialActivity.this,
                    errorMessage, Toast.LENGTH_LONG).show();
            }
        });
        // 5. (선택)전면형 광고를 닫을 시에 실행할 리스너
        mAdInterstitial.setOnAdClosedListener(new OnAdClosedListener() {
            @Override
            public void OnAdClosed() {
                Log.i("InterstitialTab", "광고를 닫았습니다. ");
            }
        });
        // 6. 전면형 광고를 불러온다.
        mAdInterstitial.loadAd();
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
        if (mAdInterstitial != null) {
            mAdInterstitial = null;
        }
    }
}
```

전면형 광고를 사용하기 위해서는 반드시 전면형 광고 Client ID를 발급받아 사용해야 한다.

CLASS REFERENCE

Package net.daum.adam.publisher

CLASS SUMMARY

AdInfo	광고 요청시 추가로 보낼 정보를 설정하는 클래스이다.
AdInterstitial	Interstitial 광고를 생성하고, 원하는 시점에 화면에 노출한다.
AdView	실제 화면에 보여지는 광고 View 의 Object 에 해당하는 클래스이다.

INTERFACE SUMMARY

AdView.OnAdClickedListener	광고를 클릭할 경우 실행될 Listener
AdView.OnAdClosedListener	광고가 닫힌 경우 실행될 Listener
AdView.OnAdFailedListener	광고 내려받기가 실패할 경우 실행될 Listener
AdView.OnAdLoadedListener	광고가 정상적으로 다운로드가 될 경우에 실행될 Listener
AdView.OnAdWillLoadListener	광고를 불러올 시점에 실행될 Listener

ENUM SUMMARY

AdView.AnimationType	광고 전환시 애니메이션 타입을 지정한다.
--------------------------------------	------------------------

AdError	광고 노출과 관련된 에러 타입을 지정한다.
----------------	-------------------------

CLASS ADINFO

광고 요청시 추가로 보낼 정보를 설정하는 클래스이다.

CONSTRUCTOR SUMMARY

AdInfo ()

광고 추가 정보 객체 생성

METHOD SUMMARY

void	setGender (String gender) 앱 사용자의 성별 정보. 앱 내부에서 사용자의 성별 정보를 가지는 경우, gender 속성에 할당하여 타겟팅 광고를 수신 받을 수 있다. 다음과 같은 포맷의 문자열만 할당 가능하며, 유효하지 않은 포맷으로 할당한 경우 무시된다. - 남성: "M" - 여성: "F"
String	getGender () 설정된 사용자의 성별을 반환한다.
void	setBirth (String date) 앱 사용자의 생년월일 정보. 앱 내부에서 사용자의 생년월일 정보를 가지는 경우, birth 속성에 할당하여 타겟팅 광고를 수신 받을 수 있다.

	<p>다음과 같은 포맷의 문자열만 할당 가능하며, 유효하지 않은 포맷으로 할당한 경우 무시된다.</p> <ul style="list-style-type: none"> - 생년월일 정보를 모두 가진 경우: "yyyyMMdd" - 연도 정보만 가진 경우: "yyyy----" - 생일 정보만 가진 경우: "----MMdd"
String	<p>getBirth ()</p> <p>설정된 사용자의 생일이 있다면 반환한다.</p>

CLASS ADVIEW

실제 화면에 보여지는 광고 view 의 object 에 해당하고, 광고요청을 직접적으로 받는 클래스다.

CONSTRUCTOR SUMMARY

AdView (android.content.Context context)

광고 객체 생성

AdView (android.content.Context context,
android.util.AttributeSet attrs)

XML 기반 광고 객체 생성

AdView (android.content.Context context,
android.util.AttributeSet attrs, int defStyle)

XML 기반 광고 객체 생성

METHOD SUMMARY

void	<p>destroy ()</p> <p>이 함수가 호출된 이후에는 다시 광고 갱신을 재개할 수</p>
------	--

	없다. 할당된 모든 리소스를 해제한다.
<code>java.lang.String</code>	<code>getClientId()</code> 클라이언트 ID 반환
<code>boolean</code>	<code>getNetworkStatus()</code> 네트워크 상태 반환
<code>int</code>	<code>getRequestInterval()</code> 광고 갱신 주기를 반환한다.
<code>java.lang.String</code>	<code>getUserAgent()</code> User-agent 문자열 반환
<code>android.webkit.WebSettings.RenderPriority</code>	<code>getWebViewRenderPriority()</code> WebView Render Priority 반환(Deprecated API)
<code>boolean</code>	<code>isInForeground()</code> 광고가 Foreground 에 있는지 여부
<code>void</code>	<code>pause()</code> 광고 갱신을 중지, 재개 하거나 더 이상 광고 요청을 하지 않는 상태로 만든다.
<code>void</code>	<code>resume()</code> 광고 갱신을 일시적으로 재개한다.
<code>void</code>	<code>setAnimationType(AdView.AnimationType type)</code> 광고 전환시 애니메이션 타입 지정

	자세한 사항은 <code>AdView.AnimationType</code> 을 참고할 것.
void	setClientId (java.lang.String clientId) 클라이언트 ID 설정
void	setOnAdClickedListener (AdView.OnAdClickedListener listener) 광고 클릭시 실행될 Listener
void	setOnAdClosedListener (AdView.OnAdClosedListener listener) 광고를 닫았을 경우에 실행될 Listener
void	setOnAdFailedListener (AdView.OnAdFailedListener listener) 광고 요청 실패시 실행될 Listener
void	setOnAdLoadedListener (AdView.OnAdLoadedListener listener) 광고 요청 완료시 실행될 Listener 설정
void	setOnAdWillLoadListener (AdView.OnAdWillLoadListener listener) 광고 요청 전에 실행될 Listener 설정
void	setRequestInterval (int interval) 광고 갱신 시간 간격을 지정할 수 있다. 최저 12 초 ~ 최고 120 초 범위에 속하는 값으로 지정할 수 있다. <u>SDK 기본값은 광고 노출 효과를 고려하여 60sec 로 지정되어 있다.</u>
void	setWebViewRenderPriority (android.webkit.WebSettings.RenderPriority priority)

	<p>광고 View 의 Render Priority 설정(Deprecated API)</p> <p>설정할 수 있는 RenderPriority 값은 아래와 같다.</p> <p>RenderPriority.NORMAL RenderPriority.HIGH RenderPriority.LOW</p>
void	<p>setAdCache(boolean useCache)</p> <p>광고 View 의 캐시 사용 여부 설정</p>
void	<p>setThreadPriority(int priority)</p> <p>광고 View 갱신 클래스의 Priority 설정(Deprecated API)</p>
int	<p>getThreadPriority(boolean useCache)</p> <p>광고 View 의 캐시 사용 여부 설정(Deprecated API)</p>
void	<p>setAdInfo(AdInfo info)</p> <p>광고 요청시 추가로 보낼 정보 설정</p>
AdInfo	<p>getAdInfo()</p> <p>광고 요청시 추가로 보낼 정보 반환</p>

ENUM ADVIEW.ANIMATIONTYPE

광고 전환시 사용할 애니메이션 타입을 지정한다.

CONSTANT SUMMARY

FLIP_HORIZONTAL

좌우 FLIP 효과

FLIP_VERTICAL

상하 FLIP 효과
SLIDE 좌에서 우로 Slide 효과
FADE FADE IN/OUT 효과
NONE 효과 없음(기본값)

INTERFACE ADVIEW.ONADCLICKEDLISTENER

광고를 클릭할 경우 실행될 Listener

METHOD SUMMARY

void	OnAdClicked() 광고를 클릭하면 실행한다.
------	--

INTERFACE ADVIEW.ONADCLOSEDLISTENER

광고가 닫힌 경우 실행될 Listener

METHOD SUMMARY

void	OnAdClosed() 광고가 닫힌 경우 실행된다.
------	--

INTERFACE ADVIEW.ONADFAILEDLISTENER

광고 내려받기가 실패할 경우 실행될 Listener

METHOD SUMMARY

void	OnAdFailed (AdError error, String errorMessage) 광고 내려받기가 실패할 경우 실행된다.
------	---

INTERFACE ADVIEW.ONADLOADEDLISTENER

광고가 정상적으로 다운로드가 될 경우에 실행될 Listener

METHOD SUMMARY

void	OnAdLoaded () 광고가 정상적으로 다운로드 되면 실행된다.
------	---

INTERFACE ADVIEW.ONADWILLLOADLISTENER

광고를 불러올 시점에 실행될 Listener

METHOD SUMMARY

void	OnAdWillLoad (java.lang.String url) 광고를 불러올 시점에 실행된다.
------	---

CLASS ADINTERSTITIAL

Interstitial 광고를 생성하고, 원하는 시점에 화면에 노출한다.

CONSTRUCTOR SUMMARY

AdInterstitial (android.app.Activity activity)

Interstitial 광고 생성자

AdInterstitial (android.app.Activity activity, java.lang.String clientId)

Interstitial 광고 생성자

METHOD SUMMARY

boolean	<p><u>isReady</u> ()</p> <p>Interstitial 광고가 READY 상태 여부를 반환한다.</p>
void	<p>loadAd ()</p> <p>Interstitial 광고를 요청하고, 즉시 화면에 노출한다..</p> <p>광고 요청이 성공하면</p> <p><code>setOnAdLoadedListener (AdView.OnAdLoadedListener)</code> 에서 지정한 Listener 를 실행하고,</p> <p>광고 요청이 실패하면</p> <p><code>setOnAdFailedListener (AdView.OnAdFailedListener)</code> 에서 지정한 Listener 를 실행한다.</p>
void	<p>setClientId (java.lang.String clientId)</p> <p>발급받은 ClientId 를 설정한다.</p>

void	setOnAdFailedListener (AdView.OnAdFailedListener listener) 광고 로딩 실패시 사용할 Listener 를 설정한다.
void	setOnAdLoadedListener (AdView.OnAdLoadedListener listener) 광고 로딩 성공시 사용할 Listener 를 설정한다.
void	setOnAdClosedListener (AdView.OnAdClosedListener listener) 불러온 광고를 닫을시에 사용할 Listener 를 설정한다.
void	setAdCache (boolean useCache) 광고 View 의 캐시 사용 여부 설정
void	setAdInfo (AdInfo info) 광고 요청시 추가로 보낼 정보 설정
AdInfo	getAdInfo () 광고 요청시 추가로 보낼 정보 반환

ENUM ADERROR

광고 요청 실패시, [AdView.OnAdFailedListener.OnAdFailed\(\)](#) 메서드 첫 번째 인자로 전달 된다. 에러에 대한 대응을 하고 싶다면, [AdView.OnAdFailedListener](#) 를 구현해 [AdView](#) 객체의 [setOnAdFailedListener\(\)](#) 메서드로 설정하면 된다.

자세한 에러 코드에 대한 설명은 아래와 같다.

CONSTANT SUMMARY

AD_DOWNLOAD_ERROR_NOAD	광고는 정상적으로 요청했으나 광고서버에서 보내줄 광고가 없는 경우
--	--------------------------------------

AD_DOWNLOAD_ERROR_NONE	에러가 없는 경우로 사실 이 함수에는 이 값은 전달되지 않는다.
AD_DOWNLOAD_ERROR_PERMISSION_DENIED	필수 권한을 추가하지 않은 경우 전달되는 에러다. 이 경우, AndroidManifest.xml 에 필수 권한을 설정해주면 된다.
AD_DOWNLOAD_ERROR_HTTPFAILED	광고요청도중 전달되는 에러이다.
AD_DOWNLOAD_ERROR_UNKNOWN_CLIENTID	Daum으로부터 발급받은 유효한 client id가 없거나 틀린 경우 전달되는 에러이다. 발급된 ID가 없다면 발급을 받아서, 해당 client ID를 입력해야 한다.
AD_DOWNLOAD_ERROR_INVALID_DEVICE	유효하지 않은 단말일 경우 전달되는 에러이다. 안드로이드 버전이나 유효한 단말인지 확인이 필요하다.
AD_DOWNLOAD_ERROR_INVALIDAD	해당 애플리케이션에 유효하지 않은 광고가 수신된 경우 전달되는 에러이다. 서버측 콘텐츠 오류에 해당한다.
AD_DOWNLOAD_ERROR_SDKEXCEPTION	SDK 내부에서 발생한 에러이다. SDK배포자에게 알려주어 처리되도록 해야한다.
AD_DOWNLOAD_ERROR_FAILTODRAW	광고를 어떠한 사유로 인해 그리지 못했을 경우에 발생하는 에러다. 광고를 보여줄 수 없는 경우나, 광고 실패시에 전달된다.

추가 정보(FAQ)

Q1. 광고 수신이 되지 않을 때는 어떻게 하나요?

Ad@m 은 유효 광고의 100% 노출을 보장하지 않습니다. 유효 광고 노출율은 송출 가능한 광고의 총 수량과 광고 호출수에 따라 달라지게 됩니다. 광고의 총 수량은 한정되어 있으나, 이에 비해 광고의 호출수가 많기 때문에 유효 광고의 수신에 실패하는 경우가 자주 발생할 수 있습니다. 또한 시간대나 앱의 종류, 날짜에 따라서도 노출 가능한 광고의 수가 달라질 수 있습니다.

배너 광고의 경우 <http://mobile.biz.daum.net> 에서 '사이트/앱 등록' 메뉴에 접속하면 '하우스애드'를 등록할 수 있습니다. 하우스애드란 자신의 애플리케이션에 자체 광고를 노출할 수 있는 기능으로, 광고 서버에서 유효 광고를 보내줄 수 없는 경우 자신이 등록한 하우스애드가 수신됩니다.

Interstitial 광고의 경우에는 하우스 애드가 지원되지 않으므로, 일정 시간 이후 다시 호출해야 합니다.

Q2. 인터넷(3G 또는 WIFI)이 연결되지 않을 경우에 어떻게 하나요?

내부적으로 인터넷 연결이 끊기면 광고 송출이 자동으로 중지되고, 연결되면 자동으로 광고 송출을 시작하게 됩니다.

Q3. 광고 영역이 텅 비어보입니다. 아담 버그 아닌가요?

최초 광고를 내려받기 전까지는 광고 요청에 시간이 걸리기 때문에 잠시 비어있을 수 있습니다. SDK 2.0 부터는 기본적으로 광고를 감싸고 있는 영역이 View.GONE 상태였다가, 광고가 완전히 내려받은 후에 View.VISIBLE 로 바꾸고 있습니다. 한번 View.VISIBLE 로 바뀐 영역은 광고 내려받기가 실패할 지라도 다시 가리지 않습니다.

어떠한 사유로 광고 내려받기가 실패할 경우에는 AdView 객체의 OnAdFailedListener 리스너를 설정해 필요한 기능을 앱에 맞게 설정하시면 됩니다.

Q4. 시스템 앱에서 EXPANDABLE 광고가 보이지 않습니다. 왜 그런건가요?

키보드 앱과 같은 시스템 앱에서는 기존 광고 영역을 자유롭게 변경하기 어려운 관계로 Expandable(확장형) 광고를 보여주기 어렵습니다. 따라서 시스템 앱에서는 단순 클릭형 배너만 노출됩니다.