PartyTrackSDK(Android) 도입자료

2014/04/16

■시작에 앞서

Android 버전의 SDK에 관해서는 jar파일을 제공하여 드리오니, 적용이 필요한 Android 프로젝트에서 라이브러리의 추가를 행하여 주시기 바랍니다.

■도입의 흐름

도입의 흐름은 하기와 같습니다.

1. jar파일을 라이브러리에 추가
2. AmdroidManifest.xml로의 설정 추가
3. 초기설정 메쏘드의 호출
4. 각종 트랙킹 메쏘드의 호출

본 자료에서는 2~4에 관한 해설 및 메쏘드의 사양을 기재하였습니다.

■Facebook mobile app install ad의 효과측정에 관해서

PartyTrackSDK는 Facebook mobile app install Ad의 효과측정에 대응하고 있습니다. 해당 기능을 이용하기 위해서는, 하기의 개인정보취급방침 및 가이드라인을 이해하고, 또한 취득할 수 있는 사용자 데이터를 활용하는 경우에는 사용자로부터 적절한 동의를 얻으신 후 활용하여 주십시요.
(iOS의 경우, IDFA의 취득이 필수입니다.)

https://www.facebook.com/about/privacy
https://www.facebook.com/ad_guidelines.php

측정을 개시하기 위해서는
https://developers.facebook.com/
에서 앱 계정을 등록하시고, App ID를 취득하여 주십시요.

PartyTrackSDK를 도입하고, App ID를 PartyTrack내에 설정한 시점부터, Facebook의 각종규약에 동의하신 것으로 간주 됩니다.

또한, 사용자 레벨의 데이터 수집을 행하는 경우는, Facebook상의 App ID를 발행 후, 하기를 확인하여 주시기 바랍니다.
https://www.facebook.com/ads/manage/advanced_mobile_measurement/tos

■ AndroidManifest.xml로의 설정추가 (필수)

본 항목에서는 하기의 설정에 대한 설명을 기재합니다.

- permission의 설정
- Android광고ID의 설정
- GooglePlay의 인스톨 리퍼러의 설정과 이용방법

모두 반드시 설정이 필요한 부분입니다.

・permission의 설정

하기의 permission을 추가하여 주시기 바랍니다. **(필수)**

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

유저의 중복판단에 이용하는 ID를 취득함에 필요한 permission은 하기와 같습니다.

원칙상 다음항에 안내드리는 Android광고 ID의 이용을 추천드리지만, 일부의 광고 네트워크에서 하기의 ID를 필요로 하는 경우가 있기에, 2014년 7월말까지는 취득하는 것이 좋은 경우도 있습니다.

imei

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

mac address

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

・Android광고ID의 설정

2014년 8월 이후, Android광고ID（Advertising ID）의 이용이 필수화됩니다.

우선, 하기의 내용을 application태그의 하위요소에 추가하여 주십시요.

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

※Android광고ID의 이용에는 Google Play Services SDK의 설정이 필수입니다. 설정내용에 대해서는 하기의 URL을 참조하여 주십시요.
https://developer.android.com/google/play-services/setup.html

※Android광고ID는 Android2.3이상 및 Google Play를 이용하고 있는 단말기에서 취득할 수 있습니다.

※Google이 규정한 Android광고ID의 정책에 대해서는 하기의 URL을 참조하여 주십시요.
http://play.google.com/about/developer-content-policy.html#ADID

※Android광고ID 취득시에, LogCat에「**The Google Play services resources were not found. Check your project configuration to ensure that the resources are included.**」로 출력되는 경우가 있습니다만, ID취득에는 영향을 끼치지 않습니다.

다음으로 GooglePlay의 인스톨 리퍼러를 이용하기 위해 하기의 receiver를 application탭의 하위요소에 추가하여 주십시요.**(필수)**

※다음 항의 MultipleReferrerReceiver를 이용하실 경우에는 본 항을 도입하실 필요가 없습니다.

```
<receiver
   android:name="it.partytrack.sdk.ReferrerReceiver"
   android:exported="true" >
   <intent-filter>
      <action android:name="com.android.vending.INSTALL_REFERRER" />
   </intent-filter>
</receiver>
```

이미 다른 receiver에서 인스톨 리퍼러를 이용하여, PartyTrack SDK과 병행하여 이용하시는 경우는, 기존 사용중이신 Receiver클래스의 onReceive 메쏘드안에서, 하기와 같이 PartyTrack SDK를 호출하여 주시기 바랍니다.

```
package your.app.package;

import it.partytrack.sdk.ReferrerReceiver;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class YourBroadCastReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        ReferrerReceiver partyReceiver = new ReferrerReceiver();
        partyReceiver.onReceive(context, intent);
        ...
        ...
    }
}
```

복수의 receiver를 호출할 필요가 있는 동시에 receive클래스의 변경이 특정한 이유로 인해 불가능한 경우※1 에는、
MultipleReferrerReceiver클래스를 이용하여 주십시요.

※MultipleReferrerReceiver클래스를 이용하실 경우, 전항의 도입을 하실 필요가 없습니다.

※MultipleReferrerReceiver클래스에서는 it.partytrack.sdk.ReferrerReceiver의 onReceive메쏘드는 자동으로 호출됩니다.

도입에는 하기와 같이 receiver를 추가하여 주시고, 추가로 호출하는 receiver클래스의 풀패스를 meta-data태그의 android:name 어토리뷰터에 기재하여 주시기 바랍니다.

```
<receiver
   android:name="it.partytrack.sdk.MultipleReferrerReceiver"
   android:exported="true" >
   <intent-filter>
      <action android:name="com.android.vending.INSTALL_REFERRER" />
   </intent-filter>
   <meta-data android:name="sample.thirdparty.Receiver1" android:value="" />
   <meta-data android:name="sample.thirdparty.Receiver2" android:value="" />
   <meta-data android:name="sample.thirdparty.Receiver3" android:value="" />
</receiver>
```

※meta-data태그에는 android:value어토리뷰터가 필수임으로 해당값은 공란등을 적절히 수정하여 입력하여 주시기 바랍니다.

※1 개발언어가 Java이외의 언어로, BroadcastReceiver를 계승한 클래스의 개발이 되지 않는 등.

다음으로, 난독화 툴을 사용하고 계신 경우는 하기와 같이 설정하여 주시 바랍니다.

Proguard의 경우, 하기와 같이 기재하여 주시기 바랍니다.

```
-keep public class it.partytrack.sdk.ReferrerReceiver{
public protected *;
}
-keep public class it.partytrack.sdk.MultipleReferrerReceiver{
public protected *;
}
-keep public class com.google.android.gms.ads.identifier.** {
public protected *;
}
```

Proguard이외의 경우, 하기를 난독화 대상에서 제외하여 주시기 바랍니다.

```
it.partytrack.sdk.ReferrerReceiver
it.partytrack.sdk.MultipleReferrerReceiver
com.google.android.gms.ads.identifier다음의 서브 클래스 및 메서드
```

※어떠한 이유로 Android광고ID를 이용하지 못 할경우, com.google.android.gms.ads.identifier이하의 서브 클래스 및 메쏘드의 난독화 제외설정은 불필요합니다.

receiver의 동작확인에 관해서는,
다음 항목의 도입상황 확인방법을 보시며 반드시 확인하여 주시기 바랍니다.

또한, 어디까지나 한정적인 확인 방법이기에, 앱 출시 후, GooglePlay를 경유하는 테스트도 필요하오니 주의하여 주시기 바랍니다.

▪인스톨 리퍼러 도입상황 확인방법

도입 확인에서는 도입된 Receiver 클래스를 shell상으로 호출하여 동작확인을 합니다.
여기서 호출하는 Receiver 클래스는 반드시 AndroidManifest.xml에 기재된 것을 지정하여 주시기 바랍니다.

도입상황 확인은 이하의 순서로 커맨드에서 진행합니다.

1. 인스톨 리퍼러의 도입이 끝난 앱을 가지고 계신 단말기에 인스톨 합니다.
- adb install ○○○.apk

2. 앱을 인스톨 단말기에서 Android의 shell을 실행합니다.
- adb shell

3. 실행한 shell에서 인스톨 리퍼러의 BroadcastIntent를 발행합니다.
- am broadcast -a com.android.vending.INSTALL_REFERRER -n your.app.package/.ReceiverClassName --es "referrer" "adzcore123adzcore"
※"adzcore123adzcore" 는 변경하지 말아 주시기 바랍니다.

4. PartyTrack 용의 리퍼러 파일이 작성된것을 확인합니다.
- cd /data/data/your.app.package/files
- ls partytrack.referrer
- ls partytrack.full_referrer

ls커멘드에서「No such file of directory」등의 결과가 되돌아 오는 경우에는 PartyTrack의 Receiver가 정상적으로 작동되지 않는 것을 의미합니다.

또한, Rooted가 아닌 단말기에서는「Permission denied」값이 되돌아 오지만, 이 경우는 특별히 문제가 없습니다.

다음 항목에서는, 인스톨 리퍼러의 적용 내용에 따른 구체적인 확인 순서와 커멘드 내용을 기재합니다.

기본적으로 PartyTrack용의 리퍼러 파일이 작성되어 진다면 정상적인 상태라고 판단하실 수 있습니다.

[Case1] PartyTrack의 Receiver클래스만을 사용하는 경우

예.
- 앱의 package는「com.test.app1」
- 작성한 apk 파일명은「TestApp1.apk」

1. 인스톨 리퍼러의 도입이 끝난 앱을 단말기에 인스톨합니다.
- adb install TestApp1.apk

2. 앱을 인스톨한 단말기에서 Android의 shell을 실행합니다.
- adb shell

3. 실행한 shell에서 인스톨 리퍼러의 BroadcastIntent를 발행합니다.
- am broadcast -a com.android.vending.INSTALL_REFERRER -n com.test.app1/it.partytrack.sdk.ReferrerReceiver --es "referrer" "adzcore123adzcore"

4. PartyTrack 용의 리퍼러 파일이 생성된것을 확인합니다.
- cd /data/data/com.test.app1/files
- ls partytrack.referrer
- ls partytrack.full_referrer

※각각의 apk 파일명, package명, Receiver클래스명을 적절한 것으로 변경하신 후, 확인작업을 행하여 주시기 바랍니다

[Case2] PartyTrack의 MultipleReferrerReceiver클래스만을 이용하는 경우

예.
- 앱의 package는「com.test.app2」
- 작성한 apk파일명은「TestApp2.apk」

1. 인스톨 리퍼러의 도입이 끝난 앱을 단말기에 인스톨합니다.
- adb install TestApp2.apk

2. 앱을 인스톨한 단말기에서 Android의 shell을 실행합니다.
- adb shell

3. 실행한 shell에서 인스톨 리퍼러의 BroadcastIntent를 발행합니다.
- am broadcast -a com.android.vending.INSTALL_REFERRER -n com.test.app2/it.partytrack.sdk.MultipleReferrerReceiver --es "referrer" "adzcore123adzcore"

4. PartyTrack 용의 리퍼러 파일이 생성된것을 확인합니다.
- cd /data/data/com.test.app2/files
- ls partytrack.referrer
- ls partytrack.full_referrer

※각각의 apk 파일명, package명, Receiver클래스명을 적절한 것으로 변경하신 후, 확인작업을 행하여 주시기 바랍니다

[Case3] 독자적인 Receiver클래스를 사용하는 경우

예.
- 앱의 package는「com.test.app3」
- 작성한 apk파일명은「TestApp3.apk」
- 독자적인 Receiver의 클래스명은「com.test.app3.MyReceiver」

1. 인스톨 리퍼러의 도입이 끝난 앱을 단말기에 인스톨 합니다.
- adb install TestApp3apk

2. 앱을 인스톨한 단말기에서 Android의 shell을 실행합니다.
- adb shell

3. 실행한 shell에서 인스톨 리퍼러의 BroadcastIntent를 발행합니다.
- am broadcast -a com.android.vending.INSTALL_REFERRER -n com.test.app3/.MyReceiver --es "referrer" "adzcore123adzcore"

4. PartyTrack용의 리퍼러 파일이 생성된것을 확인합니다.
- cd /data/data/com.test.app3/files
- ls partytrack.referrer
- ls partytrack.full_referrer

※각각의 apk 파일명, package명, Receiver클래스명을 적절한 것으로 변경하신 후, 확인작업을 행하여 주시기 바랍니다

[Case4] 다른 제3의 Receiver 클래스를 사용하는 경우

예.
- 앱의 package는「com.test.app4」
- 작성한 apk파일명은「TestApp4.apk」
- 독자적인 Receiver의 클래스명은「com.thirdparty.Receiver」

1. 인스톨 리퍼러의 도입이 끝난 앱을 단말기에 인스톨 합니다.
- adb install TestApp4apk

2. 앱을 인스톨한 단말기에서 Android의 shell을 실행합니다.
- adb shell

3. 실행한 shell에서 인스톨 리퍼러의 BroadcastIntent를 발행합니다.
- am broadcast -a com.android.vending.INSTALL_REFERRER -n com.test.app4/com.thirdparty.Receiver --es "referrer" "adzcore123adzcore"

4. PartyTrack용의 리퍼러 파일이 생성된것을 확인합니다.
- cd /data/data/com.test.app4/files
- ls partytrack.referrer
- ls partytrack.full_referrer

※각각의 apk 파일명, package명, Receiver클래스명을 적절한 것으로 변경하신 후, 확인작업을 행하여 주시기 바랍니다

■초기설정메쏘드의 호출 (필수)

우선, Track클래스를 import하여 주십시요.

```
import it.partytrack.sdk.Track;
```

※이후에 Track클래스를 이용하는 경우는 적절히 import를 하여 주시기 바랍니다.

다음으로, Track클래스의 start메쏘드를 호출하여 주십시요.
앱 실행시에 반드시 호출하여 주시기 바랍니다.

```
Track.start(context, appId, appKey);
```

context : Context형.
appId : int형. PartyTrack 관리화면에서 발행된 App ID를 입력하여 주십시요.
appKey : String형. PartyTrack 관리화면에서 발행된 App Key를 입력하여 주십시요.

Track.start메쏘드는 PartyTrackSDK의 초기화를 행하여, install의 Event를 송신합니다.
또한, 미송신의 Event가 존재하는 경우는 해당 Event 송신 역시 행해집니다.

호출의 예)
Track.start(this.getApplicationContext(), 100, “abcdefghijk”);

주의
WebView에 독자적인 UserAgent를 설정할 경우, 일부 트랙킹이 작동되지 않습니다.
독자적인 UserAgent는 사용하시지 않도록 부탁 드립니다.

■ 각종 트랙킹 메쏘드의 호출

PartyTrack에서 install, update, launch, payment의 4가지 이벤트는 자동적으로 생성되며, update, launch는 초기설정 메쏘드의 호출이 성공하면 자동적으로 측정됩니다.

install, update, launch, payment이외, 예를들어 회원등록 완료지점, 튜토리얼 완료지점 등의 이벤트를 측정하는 경우에는, 관리화면에서 이벤트를 등록하여 생성된 이벤트 ID를 가지고, event 메쏘드를 호출할 필요가 있습니다.

구체적으로는, 측정을 원하시는 임의의 타이밍에, Track클래스의 event메쏘드를 호출하여 주시기 바랍니다.

```
Track.event(eventId);
```

eventId : int형 PartyTrack 관리화면에서 발행한 이벤트의 ID를 입력하여 주시기 바랍니다.

Track.event 메쏘드는 지정된 이벤트 데이타를 송신합니다.
단, 어떠한 이유로 인해 인스톨 이벤트의 송신이 실패한 경우, 인스톨 이벤트의 송신이 완료될 때 까지 이벤트 데이터는 송신되지 않습니다.

또한 결제정보를 측정하실 경우에는, payment메쏘드를 사용합니다.

```
Track.payment(itemName, itemPrice, itemPriceCurrency, itemNum);
```

itemName: String형. 상품명
itemPrice: float형. 상품단가
itemPriceCurrency: String형. 통화코드
itemNum: int형. 상품수 ※기본적으로 1입니다.

Track.payment메쏘드는 지정되어진 과금 데이터를 송신합니다.
이 경우에도 Track.event와 동일하게, 인스톨 이벤트의 송신에 실패한 경우, 인스톨 이벤트의 송신이 완료하기까지 과금 데이터는 송신되지 않습니다.
itemPriceCurrency는 ISO 4217에 정해져 있는 알파펫 3문자의 통화코드를 지정하여 주십시요. 이용 가능한 코드 일람에 대해서는http://en.wikipedia.org/wiki/ISO_4217를 참조하여 주시기 바랍니다.

■옵션 설정(필수 항목은 아닙니다.)

PartyTrackSDK에서는 하기의 옵션이 설정가능합니다.
하기의 어느 옵션도 필수 항목은 아닙니다.

・Debug모드 ON/OFF(Default는 OFF)

debug모드를 ON으로 하기 위해서는 Track 클래스의 setDebugMode메쏘드를 true를 인수로 지정하여 호출해 주시기 바랍니다.

```
Track.setDebugMode(true);
```

debug모드가 ON의 상태에서는 PartyTrackSDK의 각 처리중에 로그가 출력되도록 되어있습니다.

※앱을 출시할 때에는 debug설정을 OFF로 하여 주시기 바랍니다.

・서버의 송신에 임의의 파라메터를 추가

서버측에 송신할 파라메터를 추가할 경우에는 Track.setOptinalParam메쏘드를 호출하여 주십시요.

※Track.setOptionalparam메쏘드는 Track.setOptionalParam메쏘드의 에일리어스 입니다.

```
Track.setOptionalParam(paramKey, paramValue);
```

paramKey: String형
paramValue: String형

※대문자의 알파벳은 사용하지 말아 주시기 바랍나다.

지정되어진 인수의 값을 paramKey=paramValue의 형식으로 서버에 송신합니다.
서버측에 송신 되어진 파라메터의 값을 포스트백에 추가할 경우는, 당사 관리화면에서 설정을 행할 필요가 있습니다.
현재는 보내주신 임의의 파라미터를 표시하는 인터페이스는 지원되고 있지 않습니다.

귀사 서비스의 회원 ID등을 파라미터로서 송신하실 경우는 하기를 사용하여 주시기 바랍니다.

Key	Value	Notes
Track.CLIENT_ID	client_id	귀사 서비스의 회원 ID등을 지정할 경우 사용하여 주십시요.

・유저 데이터의 facebook광고 최적화 이용　옵트아웃

Facebook의 효과측정을 행할 시에, 앱을 인스톨한 유저에 대하여, 해당 유저의 데이터를 Facebook광고의최적화에 이용하는 것에 대한 허가를 받을 수 있습니다.
만일 본 기능을 이용하는 경우, 하기의 메쏘드를 호출하여 주십시요.

```
Track.disableAdvertisementOptimize();
```