Mobile Service > IAP > Android SDK 사용 가이드

[공지] 구독 결제를 지원하는 신규 IAP SDK가 <u>TOAST SDK</u>로 출시됐습니다. 기존 IAP SDK는 신규 기능을 개발하지 않을 예정입니다.

개발환경

- Android Studio IDE 2.3.3 이상
- Android SDK Version은 2.3.3 (API Level 10) 이상

사용하는 오픈 소스 정보는 다음과 같습니다.

Name	Reference	Version	License
okhttp	http://square.github.io/okhttp/	1.5.4	Apache License 2.0
gson	https://code.google.com/p/google-gson/	2.2.4	Apache License 2.0

Android Studio & Gradle 환경에서 사용하기

Toast IAP SDK는 Gradle을 기반으로한 Android Studio IDE에 대한 개발환경을 제공합니다. jCenter Maven Repository 로부터 Remote로 다운로드 받을수 있습니다. 아래의 같이 프로젝트의 build.gradle 파일에 repository와 dependency에 대한 정의를 하시면 됩니다.

1. Gradle Repository

```
buildscript {
    repositories {
        jcenter()
    }
}
```

Toast IAP SDK에서 공통으로 사용되는 권한은 다음과 같습니다.

Permission	Description
android.permission.INTERNET	응용 프로그램이 네트워크 소켓을 열도록 허용합니다.
com.android.vending.BILLING	애플리케이션이 인앱 결제 권한을 부여합니다.

2. 의존성 추가하기

Google Play Store

```
dependencies {
    implementation 'com.toast.iap:iap:1.5.0'
}
```

SDK v17 (API v5) - 권장

```
dependencies {
    implementation 'com.toast.iap:iap-onestore:1.5.0'
}
```

추가되는 권한은 다음과 같습니다.

Permission	Description
android.permission.ACCESS_NETWORK_STATE	응용 프로그램이 네트워크에 대한 정보에 액세스 할 수있게합 니다.

SDK v16 (API v4)

```
dependencies {
    implementation 'com.toast.iap:iap-tstore:1.5.0'
}
```

```
[참고]
Release History
SDK의 Version의 변경이력은 패키지내의 RELEASE-NOTES.md 를 참조해주세요.
```

One Store 설정 정보

2018년 6월 12일(화)부터 구버전 SDK v16 (API v4) 이하가 적용된 신규 앱의 등록이 불가능합니다. 신규 앱을 작업하실 경우 SDK v17 (API v5)을 사용하시기 바랍니다.

- <u>인앱 SDK v15.xx.xx 버전 미만 적용 상품 지원 종료 안내</u>
- <u>구버전 IAP SDK 적용 신규 앱 등록 불가 안내</u>

1. SDK v17 (API v5)

One Store 업데이트 및 설치 유도하기

만약 SDK의 에러코드 INAPP_ONESTORE_NEED_UPDATE(201) 이 발생한다면 다음의 코드로 설치를 유도할 수 있습니다.

```
Intent intent = new Intent("android.intent.action.VIEW");
intent.setData(Uri.parse("http://m.onestore.co.kr/mobilepoc/etc/downloadGuide.
omp"));
startActivity(intent);
```

One Store 로그인 요청하기

 Toast IAP SDK 에서는 내부적으로 로그인 상태를 확인하기 때문에 별도로 로그인 처리를 하실 필요가 없습니다.

 만약 사용중 로그인이 되어 있지 않다면 OneStore 로그인 팝업(예/아니요)이 나타납니다. 로그인 팝업에서 예를 선택하게 되면 OneStore 로그인 화면으로 연결 되며 아니오를 선택하게 되면

 INAPP_ONESTORE_NEED_LOGIN(202)

```
[참고]
<u>One Store 로그인 요청하기</u>
```

팝업 결제 화면용 사용

팝업 형태의 결제화면을 사용하실 경우 AndroidMenifest.xml 에 아래 설정을 추가해주세요.

```
<application>
  <meta-data
    android:name="iap:view_option"
    android:value="popup | full" />
  </application>
```

[참고] OneStore - 인앱결제 적용을 위한 사전준비 > 7. Android Manifest 파일 설정

2. SDK v16 (API v4)

결제 테스트의 경우 아래의 설정을 AndroidMenifest.xml 에 추가해주세요.

```
<application>
  <meta-data
    android:name="iap:plugin_mode"
    android:value="development" />
  </application>
```

샘플 애플리케이션 제공

IAP Android SDK에서는 Google Play Store, One Store에 대한 샘플 애플리케이션을 제공합니다. 샘플 애플리 케이션을 사용하여 IAP Android SDK가 제공하는 기능을 간편하게 테스트 할 수 있습니다.

```
[참고]
테스트 전 유의 사항
결제 테스트에 앞서 <mark>콘솔 사용 가이드를</mark> 숙지 하시어 콘솔 환경 구성을 먼저 진행 하시기 바랍니다.
```

1. Import Project

배포된 SDK 패키지 내의 /sample 디렉토리를 Android Studio에서 Import Project 를 합니다.

2. AndroidManifest.xml 정보 설정

IAP Web Console에 등록한 Store APP ID 를 샘플 애플리케이션의 applicationId와 동일하게 설정 합니다.

```
android {
    defaultConfig {
        applicationId "your app id"
     }
}
```

[참고] applicationId 반드시 실제 스토어(Google Play Store, One Store)의 정보와 일치해야 합니다.

API Reference

1. 로그정보 활성화

디버그를 위한 로그 정보에 대한 노출을 활성화 합니다.

[Method]

public void setDebugMode(boolean isDebuggable);

[Parameter]

Туре	Name	Description
Boolean	isDebuggable	디버깅 로그 노출 여부

[Example Code]

InAppPurchases.InAppPurchase.setDebugMode(true);

2. 스토어(마켓) 설정

SDK에서 초기화 시 사용할 스토어(마켓)를 설정합니다.

[스토어 별 마켓 아이디]

MarketId	Store
GG	Google Play Store
TS	One Store SDK V16 (API V4) - 구 TStore
ONESTORE	One Store SDK V17 (API V5)

[Method]

public boolean registerMarketId(String marketId);

[Parameter]

Туре	Name	Description
String	marketId	마켓 아이디

[Example Code]

AndroidMenifest.xml 파일에서 설정 시 :

```
<meta-data
android:name="com.toast.iap.config.market"
android:value="GG" />
```

Java 코드 에서 설정 시 :

```
InAppPurchases.InAppPurchase.registerMarketId(marketId); // marketId : String
value
```

3. App ID 등록

IAP Android SDK를 사용하기 위한 서비스 ID 입니다. App ID는 TOAST Console > Mobile Service > IAP 에서 확인 가능합니다.

[Method]

```
public boolean registerAppId(long appId);
```

[Parameter]

Туре	Name	Description
Long	appld	TOAST 콘솔의 IAP Service에서 발급한 App ID

[Example Code]

AndroidMenifest.xml 파일에서 설정시:

```
<meta-data
```

```
android:name="com.toast.iap.config.appId"
android:value="1234567" />
```

Java 코드 에서 설정 시 :

```
InAppPurchases.InAppPurchase.registerAppId(1234567);// appId : long integer
```

4. 유저 등록

인증을 완료한 사용자 ID를 등록합니다. 개발사에서 정의한 사용자 식별키이며, 아이템이 지급되는 대상입니다.

[Method]

public boolean registerUserId(String userId);

[Parameter]

Туре	Name	Description
String	userld	사용자 아이디 식별자

[Example Code]

InAppPurchases.InAppPurchase.registerUserId(userId); // userId : String value

5. 결제 요청

클라이언트에서 아이템 구매를 요청합니다. 결제 요청에 대한 응답은 PurchaseCallback 을 통해 전달 받게 됩니다. 결제가 성공적으로 완료되면 결과값을 서버에 전달하여 <u>9.결제 소비</u>를 진행해야 합니다.

[참고] 인앱 결제는 결제요청과 결제소비 2단계로 진행됩니다. IAP 결제 흐름도

[Method]

public void requestPurchase(Activity activity, long itemId, PurchaseCallback callback);

[Parameter]

Туре	Name	Description
Activity	activty	어플리케이션의 현재 액티비티
Long	itemId	Web Console에서 발급된 아이템 아이디
PurchaseCallback	callback	API 요청 결과를 전달 하는 콜백

[Example Code]

```
InAppPurchases.InAppPurchase.requestPurchase(this, 1000001, new
PurchaseCallback() {
    @Override
    public void onCallback(JSONObject result, InAppPurchaseException exception)
{
        if (!result.isSuccess()) {
            // An error occurred, we need to handle the error
            return;
            }
            // Success! Include your code to handle the results here
        }
});
```

[Response Example]

```
{
    "paymentSeq": "2014082210002092",
    "purchaseToken":
"5PYSHgisiCU8BditHnDbPhmlS/0DSt4JDs2UMyg1/EY8oC6Q8qkuw5VBo7GNrBYLNUy656GCAh7h9e1
BtXeoBA==",
    "itemSeq": 1000001,
    "currency": "KRW",
    "price": 1000.0
}
```

6. 미소비 결제 내역 조회

유저의 소비(Consume) 되지 않은 결제내역을 조회합니다.

[Method]

public void queryPurchases(Activity activity, PurchaseListCallback callback);

[Parameter]

Туре	Name	Description
Activity	activty	어플리케이션의 현재 액티비티
PurchaseCallback	callback	API 요청 결과를 전달 하는 콜백

[Example Code]

```
InAppPurchases.InAppPurchase.queryPurchases(this, new PurchaseListCallback() {
    @Override
    public void onCallback(JSONArray result, InAppPurchaseException exception) {
        if (!result.isSuccess()) {
            // An error occurred, we need to handle the error
            return;
        }
        // Success! Include your code to handle the results here }
});
```

[Response Example]

```
[{
    "paymentSeq": "2014082210002092",
    "purchaseToken":
"5PYSHgisiCU8BditHnDbPhmlS/0DSt4JDs2UMyg1/EY8oC6Q8qkuw5VBo7GNrBYLNUy656GCAh7h9e1
BtXeoBA==",
    "itemSeq": 1000208,
    "currency": "KRW",
    "price": 1000.0
}, {
    "paymentSeq": "2014082210002093",
    "purchaseToken":
"Q+os4dDsYaGiEEqkLeXQfhmlS/0Dst4JDs2UMyg1/EY8oC6Q8qkuw5VBo7GNrBYLNUy656GCAh7h9e1
BtXeoBA==",
    "itemSeq": 1000208,
   "currency": "KRW",
   "price": 1000.0
}, {
    "paymentSeq": "2014082210002094",
    "purchaseToken":
"GMBcODtMnX306wVlFGIcDRmlS/0Dst4JDs2UMyg1/EY8oC6Q8qkuw5VBo7GNrBYLNUy656GCAh7h9e1
BtXeoBA==",
    "itemSeq": 1000208,
   "currency": "KRW",
    "price": 1000.0
}]
```

7. 구매 가능한 아이템 내역 조회

구매 가능한 모든 아이템 내역을 조회합니다.

[Method]

```
public void queryItems(Activity activity, PurchaseListCallback callback);
```

[Parameter]

Туре	Name	Description
Activity	activty	어플리케이션의 현재 액티비티
PurchaseCallback	callback	API 요청 결과를 전달 하는 콜백

[Example Code]

```
InAppPurchases.InAppPurchase.queryItems(activity, new
InAppPurchase.ItemListCallback() {
    @Override
    public void onCallback(JSONArray result, InAppPurchaseException exception) {
        if (exception != null) {
            // An error occurred, we need to handle the error
            return;
        }
        // Success! Include your code to handle the results here
    }
});
```

[Response Example]

```
[
    {
        "itemSeq" : 1000208,
        "itemName" : "Test item 01",
        "marketItemId": "item01",
        "price": 1000,
        "currency": "KRW",
        "localizedPrice":"#1,000"
   },
    {
        "itemSeq" : 1000209,
        "itemName" : "Test item 02",
        "marketItemId": "item02",
        "price": 7.99,
        "currency": "USD",
        "localizedPrice":"$7.99"
```

8. 미처리 결제건 일괄 재처리

미처리된 결제건(IAP 서버 검증 실패)들에 대해 일괄로 재처리 작업을 진행합니다.

[Method]

public void processesIncompletePurchases(Activity activity, IncompletePurchasesCallback callback);

[Parameter]

Туре	Name	Description
Activity	activty	어플리케이션의 현재 액티비티
IncompletePurchasesCallback	callback	API 요청 결과를 전달 하는 콜백

[Example Code]

```
InAppPurchases.InAppPurchase.processesIncompletePurchases(activity, new
InAppPurchase.IncompletePurchasesCallback() {
    @Override
    public void onCallback(JSONObject result, InAppPurchaseException exception)
{
        if (exception != null) {
            // An error occurred, we need to handle the error
            return;
        }
        // Success! Include your code to handle the results here }
});
```

[Response Example]

```
"price" : 1000.0
      },
      {
        "paymentSeq" : "2014082510002164",
        "purchaseToken" :
"8nkx3SzATK1174vmgQLzHExmlS/0DSt4JDs2UMyg1/EY8oC6Q8qkuw5VBo7GNrBYLNUy656GCAh7h
9e1BtXeoBaAC",
        "itemSeq" : 1000209,
          "marketItemId" : "item02",
        "currency" : "KRW",
        "price" : 1000.0
      }
    ],
    "failList": [
      {
        "paymentSeq" : "2014082510002165",
        "purchaseToken" : null,
        "itemSeq" : 1000210,
        "marketItemId" : "item03",
        "currency" : "KRW",
        "price" : 1000.0
      }
    ]
}
```

9. 결제 소비

사용자 애플리케이션 서버는 아이템을 지급하기 전 IAP 서버에게 결제를 소비할 것을 알려야 합니다. 결제 소비를 위 한 API는 아래를 참고 해주세요.

[참고] <u>Payment Consume API</u>

에러 처리

1. InAppPurchaseException

API 호출에 대한 에러 정보를 전달 합니다. 만약 InAppPurchaseException이 null이 아니라면 실패 상황으로 처리합니다.

Method Name	Return type	Description
getErrorCode	Integer	에러 코드를 반환 합니다.
getMessage	String	에러의 상세정보를 반환 합니다.

[Example Code]

```
InAppPurchases.InAppPurchase.queryItems(activity, new
InAppPurchase.ItemListCallback() {
    @Override
    public void onCallback(JSONArray result, InAppPurchaseException exception)
{
        if (exception != null) {
            int errorCode = exception.getErrorCode();
            String errorMessage = exception.getMessage();
            // TODO : 에러 발생시에 대한 처리를 정의 합니다.
            ....
            return;
        }
        // Success! Include your code to handle the results here
    }
});
```