

PGMAN SDK

인증 API 가이드

Ver. 6



INDEX

- 01/ 개발환경 구성
- 02/ Setup 정보
- 03/ 인증 클래스 객체 생성
- 04/ 인증 정보
- 05/ Billing 정보
- 06/ Sample App
- 07/ API 사용 함수

1 개발 환경 구성

※ 개발 환경 구성을 위한 필요 항목

- Java
- Android SDK Tool
- softnxySDK (Ver. 6)

버전

현재의 SDK는 ver. 5 입니다

SDK 변경이 있는 경우 사전에 안내해 드립니다.

2 SETUP 정보

■ SDK 설치 준비하기

- ① 다운로드 한 **pgman SDK**를 압축 해제하여 적당한 폴더에 풀어 놓습니다.
- ② **이클립스**(Eclipse in ADT)을 실행합니다.
 - > 이클립스는 안드로이드SDK에 포함 되어 있으며 가이드는 '**이클립스**'를 기준으로 설명 합니다.
- ③ **Window > preferences > General > Workspace > Text file encoding**을 **UTF-8**로 설정합니다.
 - > 그렇지 않을 경우 주석이 깨져 보일 수 있습니다.

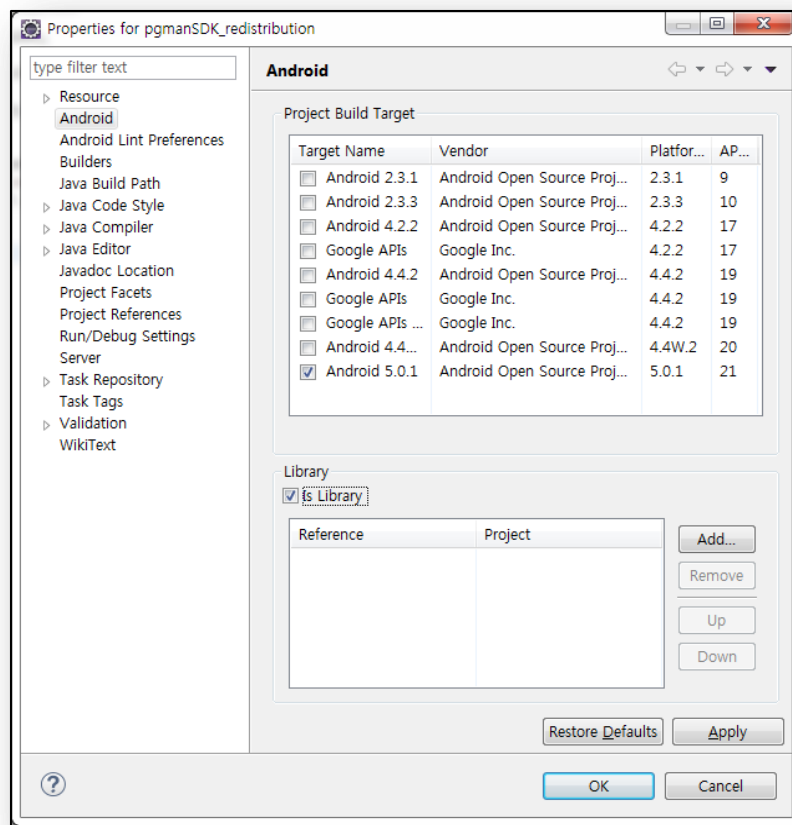
◆ Android SDK 최소 버전

- SDK Minimum API Level 은 10 입니다 (Android 2.3.3 (Gingerbread) 그 이상만 지원합니다.
API Level 9 이하로는 지원하지 않습니다.

2 SETUP 정보

■ SDK 설치

- ① 라이브러리인 **pgmanSDK_redistribution** 프로젝트를 **import** 합니다.
- ② **File -> import -> Android -> Existing Android Code Into Workspace ->** 압축 해제한 '**pgmanSDK_redistribution**' 폴더를 선택하여 불러 옵니다.
- ③ SDK를 다른 프로젝트에서 Library로 사용할 수 있도록 설정합니다. **Project -> Properties -> Android -> Is Library** 체크



2 SETUP 정보

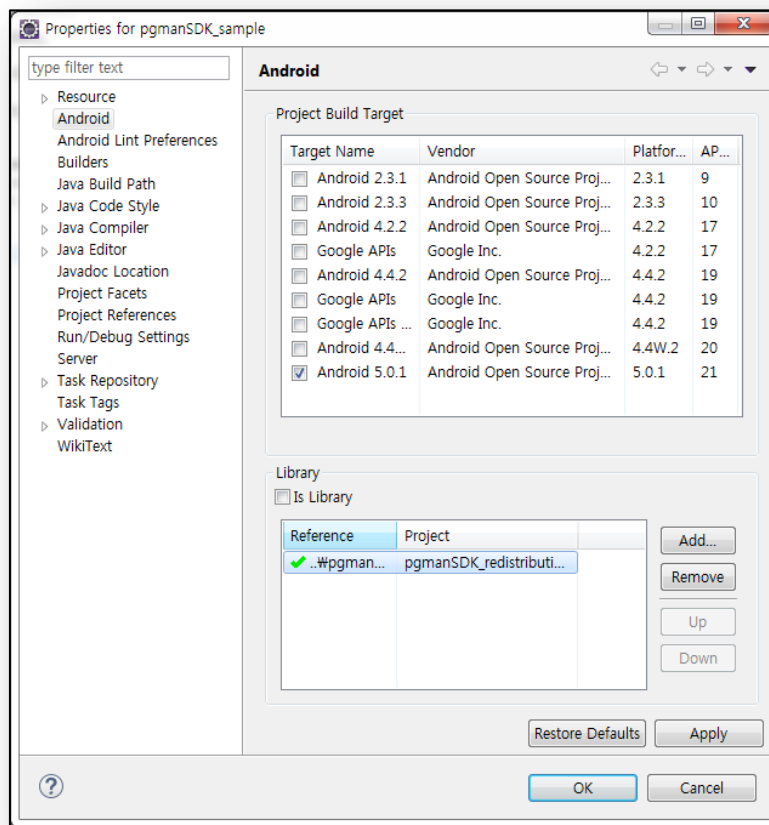
■ 새 SDK 프로젝트 생성

① 새 프로젝트를 생성합니다(API 최소 버전은 API 10 이상 이어야 합니다.)

File -> New-> Other -> Android -> Android Application Project -> 새 프로젝트를 생성

② **pgmanSDK_redistribution** 라이브러리를 연결합니다.

Project -> Properties -> Android -> Add -> pgmanSDK_redistribution 추가



2 SETUP 정보

■ AndroidManifest.xml 설정

- ① SDK의 API를 호출하기 위해서는 기본적으로 **network access**가 가능한 **uses-permission**이 필요하고 로그인 **token** 데이터를 저장하기 위한 SD 카드의 접근 권한이 필요합니다.

• **AndroidManifest.xml**파일을 열고 다음을 추가 합니다.

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

- ② SDK에서 사용되는 **Activity**를 등록해 줍니다.

```
<activity
    android:name="com.pgman.auth.Login_Activity"
    android:label="@string/pgman_app_name"
    android:configChanges="orientation|keyboardHidden|keyboard|screenSize|locale" >
</activity>

<activity
    android:name="com.pgman.inappbilling.pgmanstore_IAB_dlg"
    android:label="@string/pgman_iab_title"
    android:theme="@style/Theme.pgman.Transparent">
</activity>
```

2 SETUP 정보

③ 스토어에서 발급받은 **App APIKey** 를 추가합니다.

- 스토어에서 식별하기 위한 **App APIKey** 를 "**@string/pgman_registered_app_id**" 태그를 사용하여 메타 데이터로 등록 합니다.

```
<application
. . .
<meta-data
    android:name="@string/pgman_registered_app_id"
    android:value="7E01C4B4BF-00e46b247022e7aa6d6293c8cf4b3519_4C86CA"/>
. . .
```

기본정보	APK 등록
<p>등록도중 임시저장 버튼을 누르면 입력정보가 저장 됩니다. 필수 항목은 앱을 등록하기 위해 필수적으로 입력해야 하는 항목으로, 이를 입력하지 않으면 앱을 등록할 수 없습니다. * 표시된 항목은 필수 입력 정보 입니다.</p>	
<p>기본 정보</p>	
<p>* App APIKey</p>	<p>7E01C4B4BF-00e46b247022e7aa6d6293c8cf4b3519_4C86CA</p> <p>* SDK와 API 연동에서 이용됩니다.</p>

3 인증 클래스 객체 생성

■ Auth 인증 클래스 객체 생성

- SDK. 포함되어 있는 인증 클래스(**com.pgman.auth.Auth**) 는 인증 빌더인 **com.pgman.auth.AuthBuilder** 객체를 통해 생성됩니다. 또한 인증 클래스 내의 함수 사용시 호출되는 콜백형식의 **Listener**도 등록해야 합니다.

① Listener 등록

- **Auth** 객체 사용되는 함수의 **Listener**를 설정합니다.
- 생성된 **Listener**는 **AuthBuilder** 객체를 통해 SDK에 등록합니다.

//정보를 가져오기 위해 재정의된 Listener

// 인증 연결 시 결과값을 받는다 AuthResult로 성공 실패를 받는다.

```
AuthConnectionListener ConnectionListener = new AuthConnectionListener() {  
    @Override  
    public void OnConnection( AuthResult result ) {  
    }  
};
```

// 인증 연결 실패 시 결과값을 받는다 AuthResult로 결과값을 받는다.

```
AuthConnectionFailListener failListener = new AuthConnectionFailListener(){  
    @Override  
    public void OnConnectionFail ( AuthResult result) {  
    }  
};
```

//유저정보 가져오기에 대한 결과값을 받는다 AuthResult로 결과값을 받는다.

```
AuthUserInfoListener userListener = new AuthUserInfoListener() {  
    @Override  
    public void OnUserInfo(AuthResult result) {  
    }  
};
```

3 인증 클래스 객체 생성

② AuthBuilder 객체를 사용하여 Auth 인증 객체 생성

- `com.pgman.auth.Auth` 객체는 `com.pgman.auth.AuthBuilder`을 `build`해서 생성합니다.

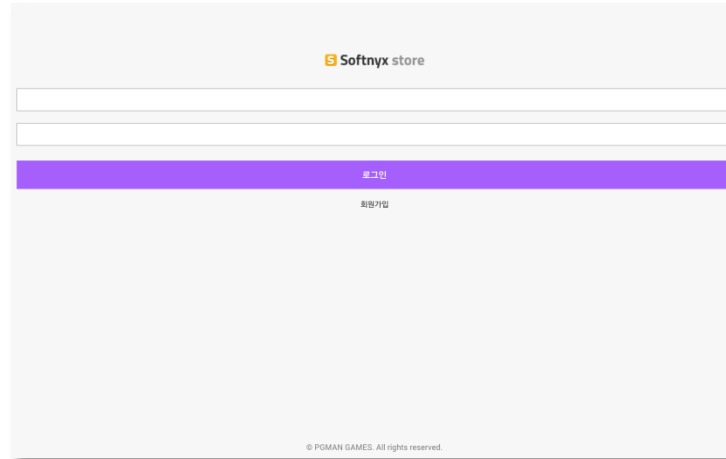
```
// com.pgman.auth.Auth.Builder 를 사용하여  
// com.pgman.auth.Auth 를 생성 해야한다  
  
// 빌더 셋  
AuthBuilder builder = new AuthBuilder( activity )  
    // 메타 데이터로 등록한 AppID를 등록한다  
    .setAppId()  
    //로그인 연결 리스너를 설정한다  
    .setConnectionListener(ConnectionListener)  
    //로그인 실패 리스너를 설정한다  
    .setConnectionFailedListener(faillistener)  
    //유저정보 리스터를 설정한다  
    .setApiForUser(userListener);  
  
Auth mPGManAuth = builder.build();  
  
mPGManAuth.Login();
```

- **Manifest**에 메타데이터로 AppID를 넣는 대신 직접 **AuthBuilder**에서 바로 설정할 수 있습니다. (`builder.setAppId("XXXXX")`)

4 인증 정보

■ Login – 스토어 계정 연동

- **Login API**는 **OAuth** 인증을 통해 Store에서 token을 발급 받고 계정 연결을 처리합니다.
- **Login API**를 호출하면 SDK는 인증정보가 있다면 로그인이 진행되고 인증정보가 틀리거나 없다면 SDK에 포함되어 있는 웹뷰를 통해 로그인 인증을 진행합니다.
- 아래는 SDK에 포함되어 Login 웹뷰 샘플 이미지입니다.



- 생성된 **Auth** 객체의 **Login()** 함수를 앱의 **Activity**를 인자로 인증 프로세스를 시작 합니다.

```
mPGManAuth.Login();
```

- **Auth.Login()**는 인증정보가 없거나 새로운 인증 정보의 갱신 등 모든 인증의 경우 사용됩니다.
- 인증 창에서 인증에 성공했을 경우 결과값은 **AuthConnectionListener**로 생성한 콜백 함수가 반환되며 인증에 실패 시 **AuthConnectionFailListener**로 등록한 콜백 함수가 반환합니다.
- 로그인 창이 닫히면 Builder 생성시 등록한 **Activity**의 **onActivityResult()** 메소드로 **Auth,REQUEST_PGMAN_LOGIN_CODE**를 반환 합니다.

4 인증 정보

- **Login()** 함수 외에 이미 인증 정보를 가지고 있다면 **LoginToBackground()**를 사용할 수 도 있습니다

```
mPGManAuth.LoginToBackground();
```

- **Auth.LoginToBackground()**는 인증정보가 없으면 **AuthConnectionFailListener**를 반환하고 인증 요구 창을 띄우지 않습니다
- **Login()** 매소드는 <인증정보가 없을시 반드시 인증요구 창을 띄워 유저에게 인증 정보를 요구> 합니다
- **LoginToBackground()** 매소드는 <인증정보가 없을시 로그아웃 상태로 인식 하고 다음 프로세스로 진행 > 합니다

4 인증 정보

```
AuthConnectionListener connectionListener = new AuthConnectionListener() {
    @Override
    public void OnConnection( AuthResult arg0) {
        // 로그인 연결 결과
        mLogin = arg0.isSuccess();
    }
};

AuthConnectionFailListener connectionFailListener = new AuthConnectionFailListener() {
    @Override
    public void OnConnectionFail ( AuthResult arg0) {
        //인증연결 실패
        mLogin = false;
        int failCode = arg0. getResultCode()
        String ErrorMessage = arg0. getMessage();
    }
};
```

- 인증 실패 시 결과값으로 인증실패 코드가 같이 넘어옵니다.

4 인증 정보

■ LogOut

- **Logout**은 사용자가 현재 사용하고 있는 디바이스에서 스토어 계정과 연결을 해제하고자 할 때 호출하는 기능입니다.

Auth 객체의 **Logout**을 호출해서 처리합니다.

```
mPGManAuth.Logout();
```

- **Logout**이 처리되면 저장 되어있는 인증정보는 모두 삭제되고 재인증시 로그인 창을 띄워 인증 처리를 해야 합니다.
- **Logout**되는 즉시 **Auth.ConnectionListener connectionListener**를 통하여 결과값을 반환합니다.

```
AuthConnectionListener connectionListener = new AuthConnectionListener() {  
    @Override  
    public void OnConnection(Result arg0) {  
        // 결과값은 false  
        mLogin = arg0.isSuccess();  
    }  
};
```

4 인증 정보

■ 유저 정보 얻기

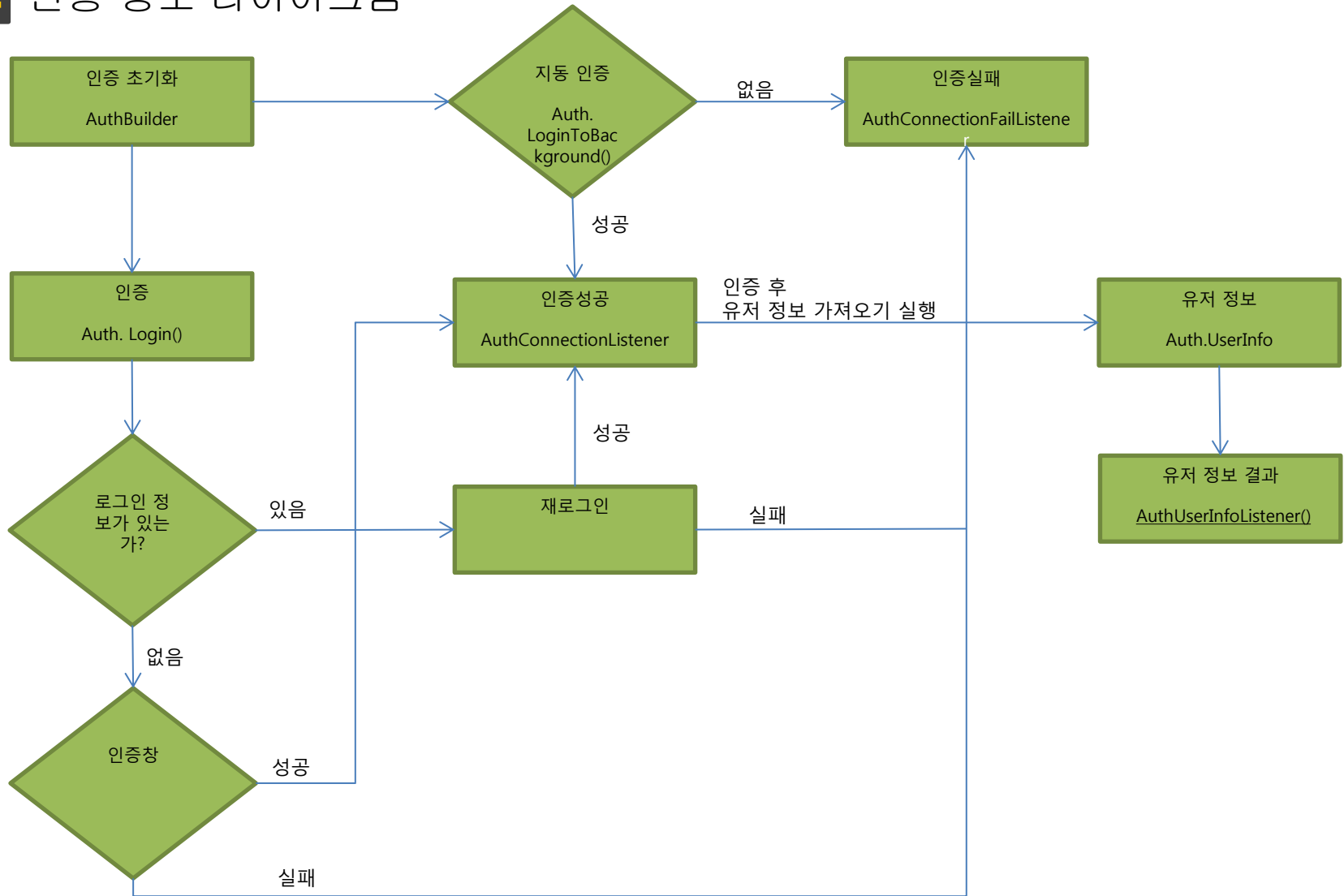
- 인증에 성공한 사용자 본인의 정보를 불러오는 API 입니다.
- 사용자의 **아이디, 닉네임, 사진 url, 캐쉬**의 정보를 **JSONObject** 형태로 얻어옵니다.
또한 원하는 값 일부만 얻어 올 수도 있습니다.
- **Auth** 객체의 **UrerInfo()** 함수를 호출합니다

```
mPGManAuth.UserInfo();
```

- 유저정보는 **Auth.UserInfoListener useInfoListener**를 통하여 결과값을 반환합니다.

```
AuthUserInfoListener userInfoListener = new AuthUserInfoListener() {  
    @Override  
    public void OnUserInfo(Result arg0) {  
        // Json Type String 으로 얻어옴  
        String jsonString = mPGManAuth.getPerson().getJsonPerson();  
  
        // 원하는 데이터 만 얻어옴  
        String ID = mPGManAuth.getApiId();           // ID  
        String Name = mPGManAuth.getApiName();       // Name  
        String photoUrl = mPGManAuth.getApiProfileImage(); // image Url  
        int cash = mPGManAuth.getApiCash();           // cash  
        int cashback = mPGManAuth.getApiCashBack();  // milige  
        boolean IsTesterAccount = mPGManAuth.IsApiDeveloper(); // tester  
    }  
};
```

4 인증 정보 다이어그램



5 Billing 정보

■ In App Billing

- 스토어에서 구매(적립)한 **캐쉬**를 사용해 앱의 '**인앱**' 상품을 구매합니다
- **SDK Version 3** 부터 **com.pgman.inAppBilling.pgman_IAB_Helper** 클래스를 사용하여 인앱결제를 합니다
- **com.pgman.inAppBilling.pgman_IAB_Helper** 클래스의 **com.pgman.inAppBilling.pgman_IAB_Helper.SetupHelper(...)** 를 사용하여 인앱결제 초기화를 합니다
- 초기화이후 결제는 **com.pgman.inAppBilling.pgman_IAB_Helper.launchPurchaseFlow(...)** 매소드를 사용하여 진행합니다
- 결제를 진행하기 전 스토어와 캐쉬 동기화에 대한 결과값을 **IabSetupFinishedListener**로 전달합니다
- 결제 진행 후 결제에 대한 결과값은 **IabPurchaseFinishedListener**로 전달 합니다

5 Billing 정보

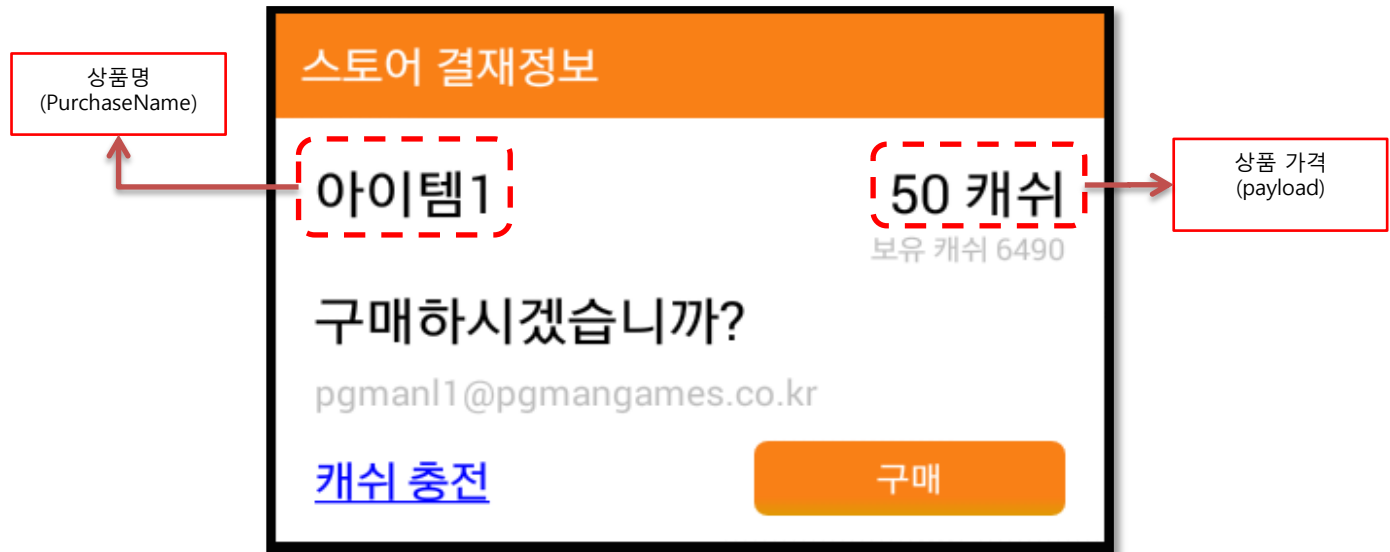
```
// 구매전 서버와 동기화 결과를 받는다
IabSetupFinishedListener setuplistener = new IabSetupFinishedListener() {
    @Override
    public void onIabSetupFinished(IabResult arg0) {
        if (arg0.isSuccess() ) {
        }
        else {
            int ErrorCode = arg0.getErrorCode();
            String ErrorMessage = arg0.getErrorMessage() ;
        }
    }
};

// 구매에대한 결과를 Purchase 로 받는다
IabPurchaseFinishedListener finishlistener = new IabPurchaseFinishedListener() {
    @Override
    public void onIabPurchaseFinished(IabResult arg0, Purchase arg1) {
        if ( arg0.isSuccess() ) {
        }
        else {
            String errorCode = arg0.getErrorCode();
            String errorMessage = arg0.getErrorMessage() ;
        }
    }
};

// IAB helper 생성
boolean bSuccess = com.pgman.inappbilling.pgmanstore_IAB_Helper.SetupHelper( mActivity,
setuplistener, finishlistener);
```

5 Billing 정보

- 스토어 로그인 인증 과정이 완료되면 '인앱' 상품의 구매를 진행합니다.
- `com.pgman.inAppBilling.pgman_IAB_Helper.launchPurchaseFlow(...)`의 인증과 서버 동기화가 끝나면 아래와 같은 구매 확인창이 나타납니다



- 메소드의 인자로 상품명(PurchaseName), 구매 가격인 캐쉬(payload), 반환코드(requestCode) 입니다
- 결제 팝업 창에 **PurchaseName** 은 상품명으로 **Payload**는 구매 가격인 <캐쉬>로 표시 되어 결제 내용을 다시 한번 확인합니다. 또한 스토어에 인증 받은 유저 인증 ID를 명시하여 포인트를 사용할 유저를 확인 합니다.
- 유저 인증 ID가 개발자 사이트에 테스트로 등록된 아이디일 경우 추가로 [This is a test account, No billing charged] 메시지가 포함되며 이는 실제 캐쉬가 소모되지 않습니다(0 이라도 결제 가능)
- 캐쉬충전링크는 스토어의 캐쉬 충전 페이지로 연결되어 구매자의 캐쉬 충전을 돕습니다

5 Billing 정보

- 스토어로부터 받은 인앱 상품 구매에 대한 결과는 **IabPurchaseFinishedListener** 로 전달되며 두 번째 인자인 **Purchase** 객체에 구매정보가 담겨있습니다.

```
IabPurchaseFinishedListener finishlistener = new IabPurchaseFinishedListener() {
    @Override
    public void onIabPurchaseFinished(IabResult arg0, Purchase arg1) {
        if ( !arg0.isSuccess() ) {
            Log.i( "PGMAN IAB Error", "code : " + arg0.getErrorCode() + "msg : "
                + arg0.getErrorMessage() );
        }
        else {
            setTextView( arg1.toString() );
            String orderId = arg1.getOrderId();
            String paymentToken = arg1.getToken();
            int Cash = arg1.getCash();
        }
    }
};

// 반환코드
int REQUEST_CODE = 1;

// 구매할 상품명
String PurchaseName = "아이템1";

// 구매할 금액
int Payment = 50;

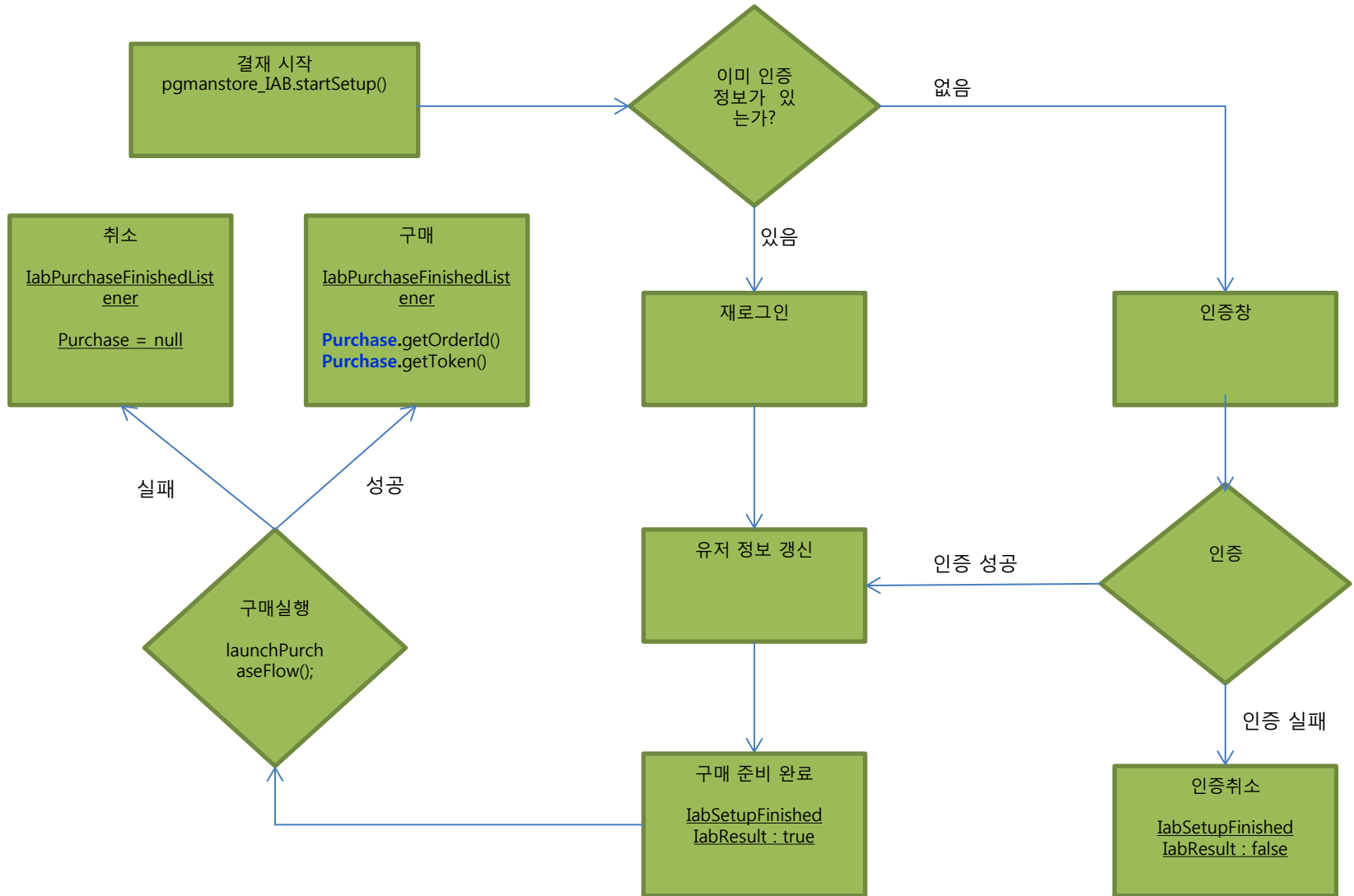
// 구매 요청 시작
pgmanstore_IAB_Helper.launchPurchaseFlow( PurchaseName, Payment , REQUEST_CODE );
```

5 Billing 정보

- **Purchase class**는 상품에 대한 **orderId** 와 **paymentToken** 그리고 사용 후 남은 캐쉬값을 반환 합니다
- 스토어로 부터 받은 <orderId>와 <paymentToken>을 사용하여 해당 구매에 대한 검증을 스토어에 질의할 수 있습니다
- 결제 확인은 최초 한번에 한해 성공으로 결과값을 리턴하며 같은 건에 두번째 요청 부터는 -2값으로 리턴합니다
- 주문번호
`String orderID = Purchase.getOrderID()`
- 결제확인 토큰
`String pamentToken = Purchase.getToken()`

결제값 확인		
호출 Url	https://storeres.softnyx.com:446/user/purchaseverify	
전송방식	Post	
요청 파라미터	order_id	주문번호(클라이언트를 통해 받는값)
	payment_token	결제토큰(클라이언트를 통해 받는값)
응답 Json	<pre>{ "returnCode": "문자열 내용" , "purchasePrice": "문자열 내용" }</pre>	
응답값 설명	returnCode	0 : 성공 -1 : 실패 -2 : 주문번호가 없거나 같은 건에 두 번 이상 요청 -3 : 결제확인 유효기간 지남 -99 : 응용프로그램에러
	purchasePrice	결제금액

5 Billing 다이어그램



6 에러 코드

◆ 인증 실패 코드

인터넷 연결 끊어짐	ERROR_CODE_NETWORK_DISCONNECT(-999)
유저 로그 아웃	LOGIN_USER_LOGOUT(10)
유저 로그인 중지	LOGIN_USER_CANCEL (11)
캐쉬 충전 사이트 접속	LOGIN_IAB_CASH_CHARAGE (12)
로그인 에러	ERROR_CODE_LOGIN_ERROR (-10)
인증토큰 틀림	ERROR_CODE_LOGIN_INVALID_TOKEN (-11)
인증토큰 유효 만료	ERROR_CODE_LOGIN_EXPIRED_TOKEN (-12)
알 수 없는 에러	ERROR_CODE_LOGIN_CONNECTED (-1)
알 수 없는 에러	ERROR_CODE_LOGIN_CONNECTED_UNKNOWN (-2)
서버오류	ERROR_CODE_LOGIN_SERVER_ERROR (-100)
클라이언트 키값 없음	ERROR_CODE_LOGIN_NO_CLIENT_KEY (-101)
App ID 오류	ERROR_CODE_LOGIN_REDIRECTION_URL (-102)
오류	ERROR_CODE_LOGIN_RESPONSE_TYPE (-103)
사용자 권한부여 취소	ERROR_CODE_LOGIN_USER_RIGHTS_CANCELED (-203)
데이터 없음(디비 에러)	ERROR_CODE_LOGIN_DB_ERRIR (-501)
사용자 정보 없음	ERROR_CODE_LOGIN_USR_INFO_NULL (-503)
Api 에러	ERROR_CODE_LOGIN_API_ERROR (-504)
인증 세션 에러	ERROR_CODE_LOGIN_SESSION_ERROR (-13)

6 에러 코드

인증 클래스 빌드 에러	ERROR_CODE_AUTH_BUILD (-90)
캐쉬정보 에러	ERROR_CODE_CASH_INFO_ERROR (-97)
인증 없음	ERROR_CODE_IAB_NO_AUTH (-91)
상품 ID 틀림	ERROR_CODE_IAB_PURCHASE_ID_WORNG (-92)
상품 지급금액 틀림	ERROR_CODE_IAB_PURCHASE_PAYLOAD_WORNG (-93)
인앱결제 초기화 안함	ERROR_CODE_IAB_INITIALIZION (-94);
상품 에러	ERROR_CODE_IAB_PURCHASE_ERROR (-14)
상품 취소	ERROR_CODE_IAB_PURCHASE_CANCEL (-15)
유저 정보 에러	ERROR_CODE_USER_INFO_ERROR (-10)
유저 디비 에러	ERROR_CODE_USER_INFO_DB_ERROR (-1)
Api 에러	ERROR_CODE_USER_INFO_API_ERROR (-99)
유저 정보 토큰 에러	ERROR_CODE_USER_INFO_TOKEN_ERROR (-504)
인앱 결제 디비 에러	ERROR_CODE_PURCHASE_DB_ERROR (-1)
인앱 결제 api 에러	ERROR_CODE_PURCHASE_API_ERROR (-99)
인앱 결제 App 없음	ERROR_CODE_PURCHASE_TOKEN_ERROR (-501)
인앱 결제 토큰 에러	ERROR_CODE_PURCHAES_TOKEN_ERROR (-502)
인앱 결제 App없음	ERROR_CODE_PURCHASE_APP_NONE (-503)
인앱 결제 타입입력 오류	ERROR_CODE_PURCHASE_TYPE_ERROR (-599)
인앱 결제 회원정보 없음	ERROR_CODE_PURCHASE_USER_INFO_NONE (-601)

6 에러 코드

인앱 결제 마이너스금액을 보유

인앱 결제 해당 개발자가 없는 경우

차감금액 부족

인앱 결제 차감캐쉬백 부족

인앱 결제 캐시 충전 및 차감 DB 오류

이미 구매한 어플

ERROR_CODE_PURCHASE_CASH_NEGATIVE (-602)

ERROR_CODE_PURCHASE_NO_DEVELOPER (-603)

ERROR_CODE_PURCHASE_CASH_NOT_ENOUGHT (-701)

ERROR_CODE_PURCHASE_MAILIGE_NOT_ENOUGHT (-702)

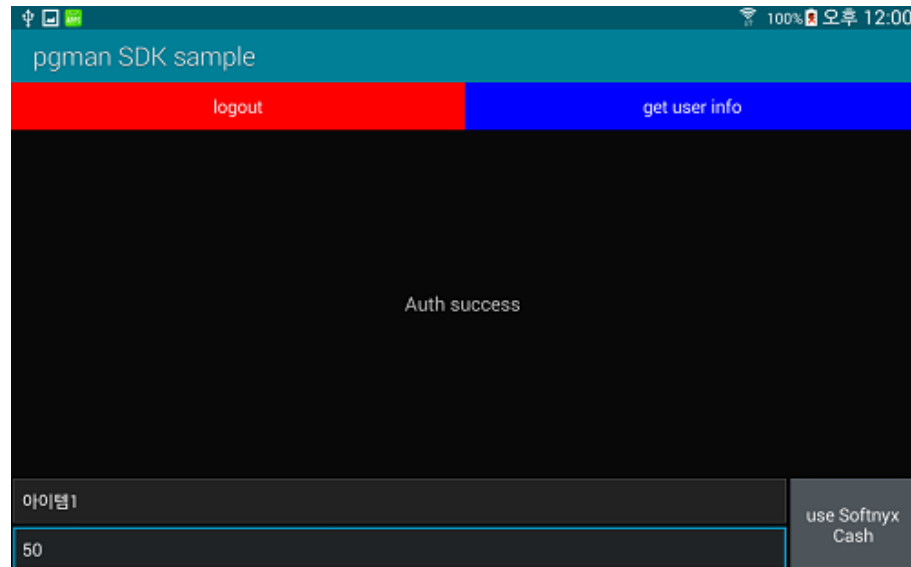
ERROR_CODE_PURCHASE_CASH_CHARAGE_DB_ERROR (-703)

ERROR_CODE_PURCHASE_BUY_ALREADY (-801)

6 Sample App

■ SDK Sample App 설명

- **Auth** 객체 생성과 인앱 상품 구매 사용법 등 전반적인 기능이 **Sample**에 포함되어 있습니다.
- **Login/Logout** 버튼을 누르면 등록된 ID와 Password를 묻는 창이 호출 됩니다.
- 성공 시 유저정보를 반환 합니다.
- 이미 인증이 되어 있다면 로그인 창을 띄우지 않고 바로 유저 정보를 반환 합니다.
- **UserInfo** 버튼을 누르면 '유저정보'를 얻어 옵니다.
- use Softnyx Cash 버튼을 누르면 '결제확인' 을 위한 창을 띄웁니다.
- 상품명(PurchaseName) 이름을 넣어 결제 테스트를 진행합니다
- 상품 가격(payload) 을 넣어 결제 테스트를 진행 합니다



7 API 사용 함수

■ com.pgman.auth.AuthConnectionListener Class Method

사용 함수	설명
public static abstract class AuthConnectionListener()	인증연결 결과에 대한 콜백을 받는다
public abstract void OnConnection(Result result);	인증 성공 시 true, 인증 로그아웃 시 false를 결과값으로 받는다

■ com.pgman.auth.AuthConnectionFailListener Class Method

사용 함수	설명
public static abstract class AuthConnectionFailListener ()	인증연결 실패 시 결과에 대한 콜백을 받는다
public abstract void OnConnectionFail (Result result);	결과값으로 false를 받는다

■ com.pgman.auth.AuthUserInfoListener Class Method

사용 함수	설명
public static abstract class AuthUserInfoListener ()	유저정보에 대한 콜백을 받는다
public abstract void OnUserInfo (Result result);	성공시 true 실패시 false를 결과값으로 받는다

7 API 사용 함수

■ com.pgman.auth.Auth Class Method

사용 함수	설명
public void UserInfo() 결과를 PointInfoListener 로 반환	유저정보를 받아온다.
public String getApiId(){	유저의 ID를 얻어온다.
public String getApiName(){	유저의 이름을 얻어온다.
public String getApiProfileImage(){	유저의 이미지 주소를 얻어온다.
public int getApiCash(){	유저의 캐쉬 정보를 가져온다.
public int getApiCashBack(){	유저의 포인트 정보를 가져온다.
public boolean IsLogin()	유저의 로그인 여부를 알린다.

7 API 사용 함수

■ com.pgman.auth.Auth Class Method

사용 함수	설명
public int SessionState() 반환값 Session.SESSION_IN Session.SESSION_OUT Session.SESSION_EXPIRY	현재 유저의 인증된 상태를 가져온다. 인증코드의 기한이 유효함 - 로그인 상태 인증코드가 없음 - 로그아웃 상태 인증코드의 기한이 만료 - 재인증 해야함
public void Login() 성공시 AuthConnectionListener 반환 실패시 AuthConnectionFailListener 반환	현재 실행된 앱의 Activity를 인자로 로그인 창을 띄운다.
public void LoginToBackground() 성공시 AuthConnectionListener 반환 실패시 AuthConnectionFailListener 반환	로그인 아이디를 묻지 않고 로그인을 시도한다
public void Logout()	로그아웃
public void addConnectionListener(AuthConnectionListener Listener) public void addConnectionFailedListener(AuthConnectionFailListener Listener) public void addApiForUser(AuthUseInfoListener Listener)	인증정보를 위한 Listener를 등록한다.
public Person getPerson()	유저정보를 저장한 Person class를 가져온다.

7 API 사용 함수

■ com.pgman.auth.Person Class Method

사용 함수	설명
public String getUnityPerson (){	유저정보를 JSON형식의 string으로 읽어온다.
Public String unityCashString ()	캐쉬정보를 JSON형식의 string으로 읽어온다.
public String getName(){	유저이름을 가져온다
public String getId()	유저 ID를 가져온다
public Integer getCash()	캐쉬 정보를 가져온다
public Integer getCashBack(){	유저 캐쉬 포인트를 가져온다
public String getImagePath(){	유저 이미지 정보를 가져온다

■ com.pgman.InAppBilling Class Method

사용 함수	설명
public abstract class IabSetupFinishedListener	결제 프로세스 초기화(스토어 인증)에 대한 결과를 돌려 받는다
public abstract class IabPurchaseFinishedListener	결제 처리 후 서버에서 처리된 결과를 돌려 받는다
public class IabResult	결과에 대한 에러 코드를 반환한다

7 API 사용 함수

■ com.pgman.InAppBilling.pgman_IAB_Helper Class Method

사용 함수	설명
public static boolean SetupHelper (Activity activity, IabSetupFinishedListener setuplistener, IabPurchaseFinishedListener finishlistener)	인앱 결제에 대한 초기화를 수행한다
public static boolean launchPurchaseFlow (String purchaseId, int payload, int requestCode)	서버와 인앱 결제를 시작한다

■ com.pgman.InAppBilling.Purchase Class Method

사용 함수	설명
public String toString ()	결제 정보를 JSON형식으로 받는다
public String getOrderId ()	서버에서 발급한 주문ID를 가져온다
public int getCash ()	결제 처리후 남은 캐쉬를 가져온다
public int getReturnCode ()	결제 처리에 대한 에러코드를 가져온다
반환값 ERROR_CODE_PURCHASE_DB_ERROR ERROR_CODE_PURCHASE_TOKEN_ERROR	DB 에러 Token 오류



End

문의 사항이 있으신 경우 help@pgmangames.co.kr로 메일 주시기 바랍니다.

감사합니다.