

# PmangPlus Android SDK 연동 가이드

SDK Version:	1.3.2
작성일:	2011-10-31

## 목차

1	문서 개요 .....	1
2	연동 준비 사항.....	1
3	SDK 설치 .....	2
3.1	SDK 파일 구성 .....	2
3.2	Eclipse 프로젝트 설정.....	2
3.3	샘플 앱 실행.....	4
4	SDK 연동 .....	7
4.1	Activity/Permission 등록 .....	7
4.2	초기화 로직 호출 .....	8
4.3	SDK API 호출 .....	10
4.3.1	UI 표출 API.....	10
4.3.2	동기 API.....	11
4.3.3	비동기 API.....	11
4.4	SDK 상태 Listener .....	13
4.5	로그 레벨 설정 .....	13
4.6	사용자 게임 정보 Storage .....	13
4.7	Standalone 모드 설정 .....	14
4.8	PPDelegate 구현 .....	14
5	연동 Step-By-Step .....	16
5.1	SDK Import .....	16
5.2	프로젝트 생성.....	17
5.3	PP_SDK를 라이브러리로 지정 .....	18
5.4	연동 코드 작성.....	23
6	Troubleshooting.....	28

# PmangPlus Android SDK 연동 가이드

## 1 문서 개요

본 문서는 안드로이드 앱에 PmangPlus의 기능을 적용하기 위한 개발자 가이드로, 다음의 내용을 제공한다.

- PmangPlus Android SDK(이하 SDK)의 설치 방법
- 샘플 앱의 설치와 실행 방법
- 앱과의 연동 방법

## 2 연동 준비 사항

SDK는 API Level 7 이상에서 동작한다. 연동을 위해 준비되어야 할 사항은 다음과 같다.

- PmangPlus 앱 등록정보: 연동을 하기 위해선 PmangPlus에 앱을 등록할 때 발급된 app id, app key, app secret이 필요하다. (샘플을 실행할 경우라면 불필요함)
- Android 개발 환경: 최소 Android API Level 7(Platform Version 2.1u1)을 빌드 할 수 있는 개발환경.  
Eclipse + ADT Plugin for Eclipse를 권장함.
- PmangPlus SDK: 최신 PmangPlus Android SDK

## 3 SDK 설치

### 3.1 SDK 파일 구성

SDK는 두 개의 Eclipse Project가 압축된 형태로 배포된다. 파일명은 PP\_SDK\_Ver[SDK버전][빌드번호].zip 의 형태이다. SDK 버전이 1.1이고, 빌드 번호가 65라면 파일명은 PP\_SDK\_Ver1\_1b65.zip이 된다.

압축파일을 적당한 위치에 풀면 2개의 프로젝트 디렉터리가 나온다. 각 프로젝트의 내용은 다음과 같다..

- PP\_SDK: 연동에 사용할 SDK 프로젝트. Android Library Project임.
- PP\_SDK\_SAMPLE: 샘플 앱. PP\_SDK 라이브러리를 사용하며, PmangPlus가 제공하는 기능을 시험할 수 있음.

### 3.2 Eclipse 프로젝트 설정

다음은 Eclipse로 프로젝트를 설정하는 절차를 설명한다. Eclipse가 아닌 다른 개발환경을 사용할 경우, 해당 개발환경에 맞도록 프로젝트를 설정해야 한다.

1. 압축을 푼 두 개의 Project를 Eclipse로 Import 한다. Eclipse의 메뉴 File > Import > Existing Projects into Workspace를 이용한다.

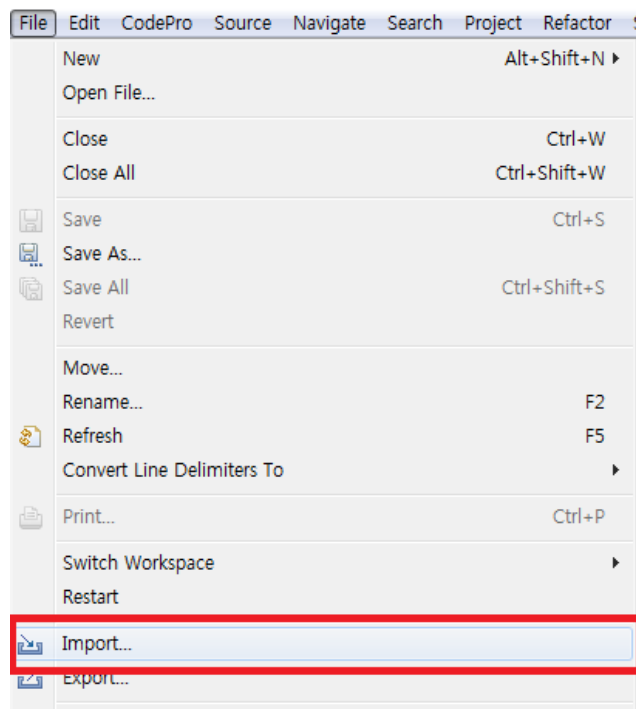


그림 1 Eclipse 의 File > Import 메뉴

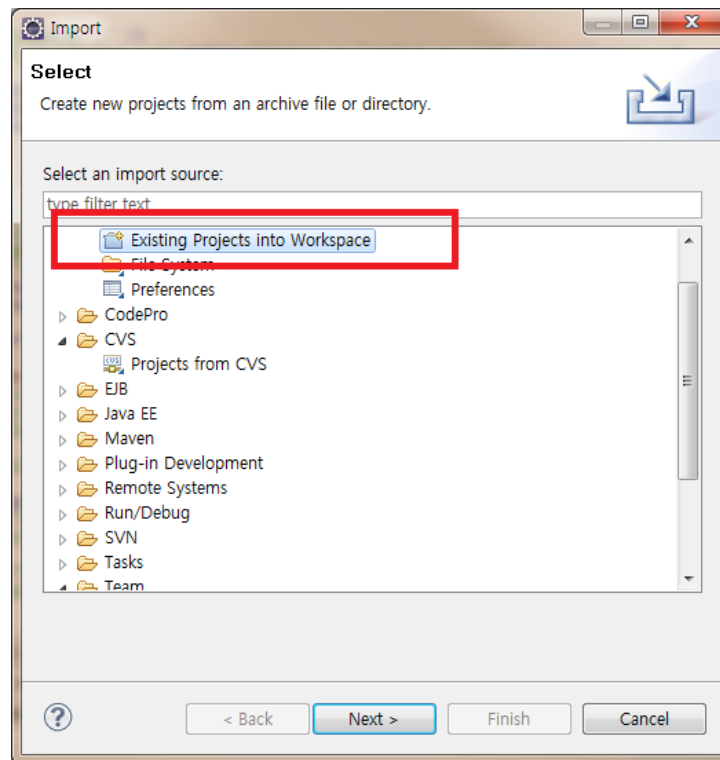


그림 2Eclipse 의 Existing Projects into Workspace 메뉴

2. Import 창이 뜨면, 압축을 풀었던 디렉터리를 지정한다. 2개의 프로젝트가 목록에 보이면 모두 check된 상태로 Finish를 선택한다.

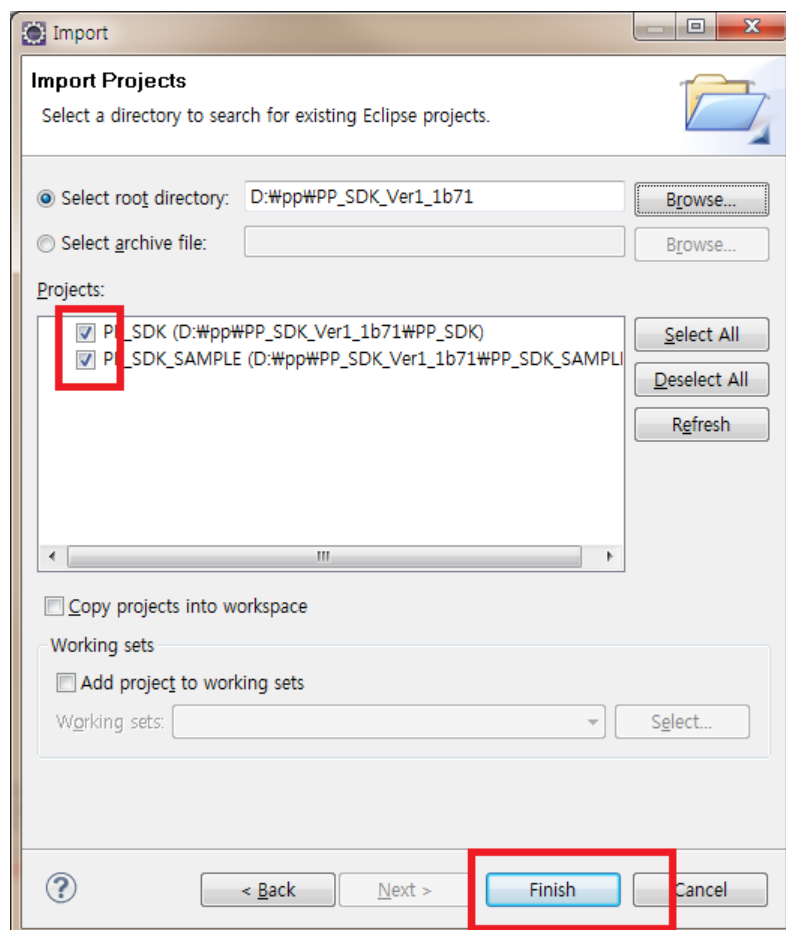


그림 3 프로젝트 Import 화면

3. 여기까지 진행했다면 Project Explorer에 PP\_SDK, PP\_SDK\_SAMPLE 두 개의 프로젝트가 표시된다. 빌드 에러가 발생할 경우 Project> Clean을 통해 Clean을 시도한다.

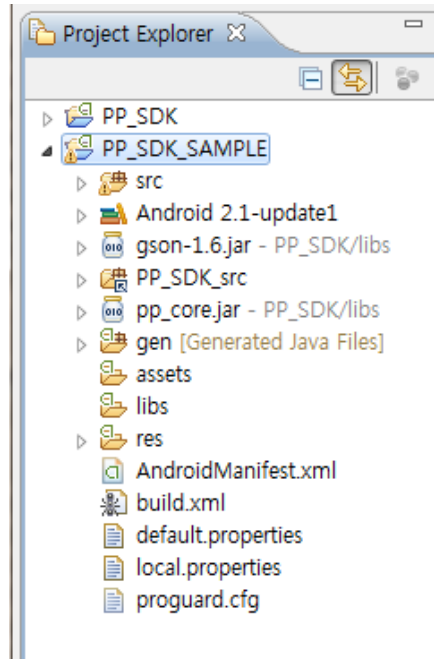


그림 4 프로젝트 import 결과

### 3.3 샘플 앱 실행

Import를 마친 두 개의 프로젝트 중 PP\_SDK\_SAMPLE을 실행해서 SDK의 기능을 확인한다.

1. Project Explorer에서 PP\_SDK\_SAMPLE 프로젝트를 선택한 후, 우클릭> Run As> Android Application을 선택한다.

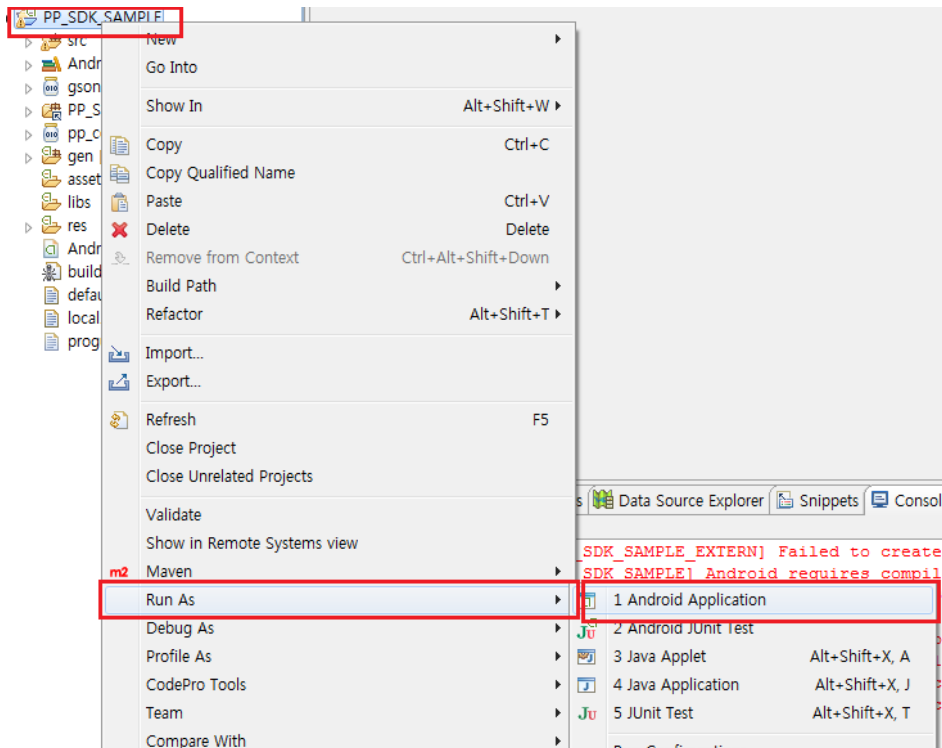


그림 5 샘플 앱 실행

2. 실행에 성공하면 연결된 디바이스나 에뮬레이터로 다음 화면이 표시된다.



그림 6 샘플 앱 초기화면

샘플앱이 제공하는 기능은 다음과 같다.

1. 자동로그인: 이전에 로그인한 적이 있다면, 그 아이디로 PmangPlus 로그인을 한다. 처음 사용하는 것이라면, 새 계정 생성창이 뜬다.
2. 새계정 생성창 띄우기: PmangPlus 새 계정 생성창을 띄운다.
3. SNS 포스팅: PmangPlus의 SNS 글쓰기 창을 띄운다. 로그인 후 사용 가능.
4. 새계정 생성창 배경: 새 계정 생성창의 배경 이미지를 설정한다. 배경 이미지로는 assets 폴더 내의 이미지를 사용한다.
5. 대시보드 열기: PmangPlus 대시보드 화면을 표시한다. 로그인 후 사용 가능.
6. 순위표 열기: PmangPlus 순위표 화면을 표시한다.
7. 도전과제 열기: PmangPlus 도전과제 화면을 표시한다.
8. 도전과제 조작: 로그인한 사용자의 도전과제 달성 정보를 변경한다. 로그인 후 사용 가능.
  - A. 도전과제 조작을 선택하면 앱에 등록된 도전과제 목록이 표시된다.
  - B. 상태를 변경할 도전과제를 선택하면 상태 변경 대화창이 표시되고, 변경 대상 상태를 선택한다.
  - C. 상태 변경에 따른 성공/실패 결과가 표시된다.
9. 순위표 조작: 로그인한 사용자의 순위표 정보를 변경한다. 로그인 후 사용 가능.
  - A. 순위표 조작을 선택하면 앱에 등록된 순위표 목록이 표시된다.

- B. 점수를 변경할 순위표를 선택하면 점수 입력 화면이 표시되고, 새로운 점수를 입력한다. 지역 기반 순위표의 경우 지역명을 입력할 수 있다. 기존 점수보다 낮은 점수를 입력할 경우 변경이 일어나지 않는다.
10. 기타 API 테스트: 그 밖의 API를 테스트한다.
- A. 로그인 확인: 현재 로그인 된 상태인지 확인한다.
- B. 멤버 정보: 로그인 된 사용자의 상세 정보를 조회한다. 로그인 후 사용 가능.
- C. 멤버 앱 정보: 사용자가 현재 앱에서 달성한 정보를 조회한다. 로그인 후 사용 가능.
- D. 친구 닉네임 가져오기: 현재 앱을 사용하는 PmangPlus 친구들의 닉네임을 가져온다. 로그인 후 사용 가능.
- E. 점수 및 내 순위: 기본 순위표의 로그인 사용자의 점수와 등수를 조회한다. 로그인 후 사용 가능.
- F. 데이터 저장하기: PmangPlus가 제공하는 Storage 기능을 사용해 데이터를 저장한다. 데이터는 key-value 형식의 Map으로 구성되며, key와 value는 모두 String 타입이다. Storage는 앱 별로 제공된다.
- G. 데이터 불러오기: 키 값에 해당하는 데이터를 구한다.
- H. 데이터 키 리스트: PmangPlus Storage의 키 리스트를 구한다.
- I. 앱 배너 오픈하기: 프로모션을 위한 전체 화면 배너를 띄운다. "현재 init된 앱 배너 오픈"을 클릭할 경우 앱의 등록된 배너를 표시한다. 사전에 admin 사이트에서 배너를 등록하지 않았을 경우엔 "로딩 창"만 잠시 표시되고 아무 내용도 표시되지 않으며, 등록된 배너가 있을 경우엔 하단의 skip을 선택하면 표출된 배너가 닫힌다.

현재 샘플앱은 PmangPlus의 테스트 앱으로 설정되어 있다. 이 설정을 변경하고자 한다면, com.pmangplus.sample.PPApplicatoin의 PPConfig 부분을 사용하고자 하는 앱의 정보로 변경하면 된다.

```
public class PPApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();

        Handler handler = new Handler();

        // app registration information
        int appId = 1;
        String appKey = "test_app_key";
        String appSecret = "test_app_secret";

        // target pmangplus api server
        ApiServer targetApiServer = ApiServer.QA;

        // target country: have effect on SNS types and payment currency
        Country targetCountry = Country.GLOBAL;

        // other configuration information
        OptionalConfig opConfig = new OptionalConfig();
        opConfig.screenOrientation = ScreenOrientation.LANDSCAPE; // screen orientation. default value is ScreenOrientation.LANDSCAPE
        opConfig.isStandalone = false; // set flag whether player can use app without logging in PmangPlus. default value is false
        opConfig.pg = PG.PAYGATE; // set target payment gateway. ignore if app does not use pmangplus payment.
        opConfig.logLevel = LogLevel.DEBUG; // set log level for PmangPlus. should set WARN or ERROR on release version.

        final PPConfig config = new PPConfig(appId, appKey, appSecret, targetApiServer, targetCountry, opConfig);

        PPDelegate delegate = new PPDelegateImpl();

        // initialize PmangPlus
        PPImpl.init(config, delegate, getApplicationContext(), handler);
    }
}
```

그림 7 샘플 앱의 앱 정보 설정 부분



## 4 SDK 연동

개발 중인 앱(이하 앱)에서 SDK를 연동하기 위한 절차는 다음과 같다.

1. 앱의 AndroidManifest.xml에 SDK의 activity를 등록한다. 만약 SDK가 요구하는 permission이 등록되어 있지 않았다면, permission도 추가로 등록한다.
2. SDK 초기화 로직을 호출하도록 앱의 초기화 과정을 수정한다.
3. 로그인 등 SDK 기능 호출 부분을 앱에 추가한다.

### 4.1 Activity/Permission 등록

SDK는 자체 UI를 제공하기 때문에 Activity를 앱의 AndroidManifest.xml에 등록해야 한다. 등록해야 할 Activity들은 PP\_SDK 프로젝트의 AndroidManifest.xml에서 확인할 수 있다.

PP\_SDK/AndroidManifest.xml의 개략적인 내용은 다음과 같다.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.pmangplus.ui"
    android:versionCode="1" android:versionName="1.0">
    <uses-sdk android:minSdkVersion="7" />

    <!-- required permission for pmangplus -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <!-- required feature for pmangplus -->
    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />

    <!-- put custom content -->
    <!-- pmangplus activity -->
    <activity android:name="com.pmangplus.ui.activity.PmangPlusMain"
        android:label="@string/app_name"
            android:theme="@style/pp_main" android:launchMode="singleTask"
        android:clearTaskOnLaunch="true" android:configChanges="orientation|keyboardHidden" />
        ...
    <activity android:name="com.pmangplus.ui.activity.PPPaymentActivity"
        android:label="@string/app_name"
            android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden" />

    <!-- pmangplus dialog -->

    <activity android:name="com.pmangplus.ui.dialog.PPSignup" android:theme="@style/pp_dialog"
        android:configChanges="orientation|keyboardHidden"
        android:windowSoftInputMode="stateHidden" />
        ...
    <activity android:name="com.pmangplus.ui.dialog.PPNotiDialog"
        android:theme="@style/pp_dialog"
            android:configChanges="orientation|keyboardHidden" />
</manifest>
```

리스트 1 PP\_SDK/AndroidManifest.xml의 형태

위 파일 내용 manifest 선언부를 제외한 내용 전부를 앱의 AndroidManifest.xml로 복사한다. uses-permission과 uses-feature, uses-configuration이 중복될 경우 중복된 항목은 제외할 수 있다.

## 4.2 초기화 로직 호출

SDK의 기능을 사용하려면 사전에 초기화를 해야 한다. 초기화를 하지 않고 SDK의 기능을 호출할 경우엔 exception이 발생할 수 있다. 초기화는 `com.pamngplus.ui.PPImpl.init()` 로 수행한다. 메서드와 각 인자의 의미는 API javadoc을 참고한다.

PmangPlus 는 singleton 구조로 되어 있어 반드시 Application에서 초기화 되어야 한다. 다음은 샘플 앱의 초기화 예시이다.

```
public class PPApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();

        Handler handler = new Handler();

        // app registration information
        int appId = 1;
        String appKey = "test_app_key";
        String appSecret = "test_app_secret";

        // target pmangplus api server
        ApiServer targetApiServer = ApiServer.QA;

        // target country: have effect on SNS types and payment currency
        Country targetCountry = Country.GLOBAL;

        // other configuration information
        OptionalConfig opConfig = new OptionalConfig();
        opConfig.screenOrientation = ScreenOrientation.LANDSCAPE; // screen orientation. default value is ScreenOrientation.LANDSCAPE
        opConfig.isStandalone = false; // set flag whether player can use app without logging in PmangPlus. default value is false
        opConfig.pg = PG.PAYGATE; // set target payment gateway. ignore if app does not use pmangplus payment.
        opConfig.logLevel = LogLevel.DEBUG; // set log level for PmangPlus. should set WARN or ERROR on release version.

        final PPConfig config = new PPConfig(appId, appKey, appSecret, targetApiServer, targetCountry, opConfig);

        PPDelegate delegate = new PPDelegateImpl();

        // initialize PmangPlus
        PPImpl.init(config, delegate, getApplicationContext(), handler);
    }
}
```

그림 8 샘플 앱의 초기화 로직 호출

### 중요

ScreenOrientation은 게이머의 혼선을 줄이기 위해 게임의 특성에 맞도록 설정한다. 즉, 가로 전용 게임일 경우 ScreenOrientation.LANDSCAPE로, 세로 전용 게임일 경우 ScreenOrientation.PORTRAIT로 설정한다.

이렇게 설정할 경우 PmangPlus SDK의 UI의 방향은 기기의 중력센서와 무관하게 설정된 방향으로 고정된다. ScreenOrientation.UNDEFINED의 경우엔 화면을 띄울 시점의 중력센서 값을 이용하므로 가로 전용 게임에서 UI가 세로로 표현되는 문제가 생길 수 있다.

### 중요

ApiServer는 연동 대상 서버를 지정한다. 현재 선택 가능한 서버는 QA, REAL, REAL\_GLOBAL의 3가지이다.

QA는 개발단계 혹은 샘플의 테스트에 사용하는 서버이므로, 앱 연동 시엔 서버를 REAL로 지정해야 한다. 해외 출시 앱이라면 접속 속도를 보장하기 위해 서버를 REAL\_GLOBAL로 지정해야 한다.

### 중요

isStandalone을 true로 설정할 경우, 회원가입 창에 '취소' 버튼이 표시되며, 사용자가 이 버튼을 누를 경우 가입 프로세스가 취소된다.

PmangPlus 로그인을 하지 않은 상태에서도 플레이가 가능해야 하는 앱이라면 이 값을 true로 설정한다. 기본값은 false이다.

단, 성인인증이 필요한 앱의 경우엔 이 값을 반드시 false로 해야 한다.

#### 중요

Country는 연동 대상 SNS의 종류와 결제 통화에 영향을 미친다. 동일한 앱을 여러 국가에 출시할 경우, 각 출시 국가에 맞게 Country를 설정해야 한다.

## 4.3 SDK API 호출

SDK의 API는 다음 3가지 유형으로 나뉜다.

1. UI 호출 API: 대시보드, 순위표 등 SDK가 제공하는 UI를 표시.
2. 동기 API: 동기화된 형태로 값을 반환. 로그인한 사용자 정보 등 네트워크가 필요하지 않은 결과 반환 메시드가 이 유형에 해당한다.
3. 비동기 API: 비동기화된 형태로 값을 반환. 결과를 받기 위한 콜백이 필요하다. 순위표 목록 등 네트워크로 결과를 받아야 하는 대부분의 메시드가 이 유형에 해당한다.

모든 SDK의 API는 인터페이스 `com.pmangplus.ui.PP`에 정의되어 있으며, 기본 구현체인 `PPImpl`을 이용해 호출할 수 있다. 상세한 내용은 API javadoc을 참고한다.

다음은 PP의 메시드의 공통 특성이다.

- return type이 boolean인 method는 실행 환경에 대한 validation을 확인하고 결과를 반환한다. 메시드 실행에 필요한 조건을 충족하지 못한 경우 메시드는 false를 반환하며, validation을 제외한 어떤 기능도 수행하지 않는다. callback의 `onFail`도 호출되지 않으므로 주의한다.
- 동기화 기능은 callback을 인자로 받는다. callback으로 null을 넘길 경우엔 초기화 시 인자로 넘긴 delegate의 해당 기능이 실행된다.

### 4.3.1 UI 호출 API

UI 호출 기능은 SDK가 제공하는 UI를 호출하는 API이다. 접두사 `open`이 붙는 메시드에 해당한다. 다음은 샘플 앱에서 발췌한 코드이다. 참고로 로그인이 되지 않은 상태에서 PmangPlus의 UI 호출 API를 호출한 경우, SDK는 자동로그인을 시도한다.

```
openSignupBtn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // open sign up
        PPImpl.getInstance().openSignUp(Main.this, null);
    }
});

openDashboardBtn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // open dashboard
        PPImpl.getInstance().openDashboard(Main.this);
    }
});

openAchvBtn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // open achievement
        PPImpl.getInstance().openAchievement(Main.this);
    }
});
```

그림 9 샘플 앱에서 발췌한 UI 호출 API 호출 코드

### 4.3.2 동기 API

UI 표출 기능을 제외한 PP의 일부 메서드 중 callback을 인자로 갖지 않는 메서드가 동기 API에 해당한다. SDK 내부의 결과를 반환하므로 즉시 결과를 반환한다. 다음은 샘플 앱에서 발췌한 코드이다.

```
// member information of login user
Member m = PPImpl.getInstance().getMemberInfo();
if( m != null){
    tv.append("email: "+m.getEmail()
        +"\nmy feeling: "+m.getFeeling()
        +"\nmember ID: "+m.getMemberId()
        +"\nnickname: "+m.getNickname()
        +"\nprofile image URL: "+m.getProfileImgUrl()
        +"\nrecent played app ID: "+m.getRecentAppId()
        +"\nrecent played app name: "+m.getRecentAppTitle()
        +"\nadult auth date: "+m.getAdultAuthDt()
        +"\nadult auth flag: "+m.getAdultAuthYn()
        +"\nanonymous user flga: "+m.getAnonymousYn()
    );
}
```

그림 10 샘플 앱에서 발췌한 동기 API 호출 코드

### 4.3.3 비동기 API

대다수의 API는 비동기 API이다. 비동기 API는 APICallback 타입의 callback을 인자로 가지며, 앱은 이 callback을 구현하여 비동기 호출 결과를 받을 수 있다. callback 인자가 null일 경우 PPDelegate의 해당 메서드가 호출된다. (PPDelegate는 SDK 초기화 시 인자로 전달된다.) API 호출 결과가 동일하다면 PPDelegate의 해당 메서드에 미리 구현하고 인자로 null을 넘기도록 구현할 수 있다. 이 내용을 도식화 한 결과가 그림 11에 나와있다.

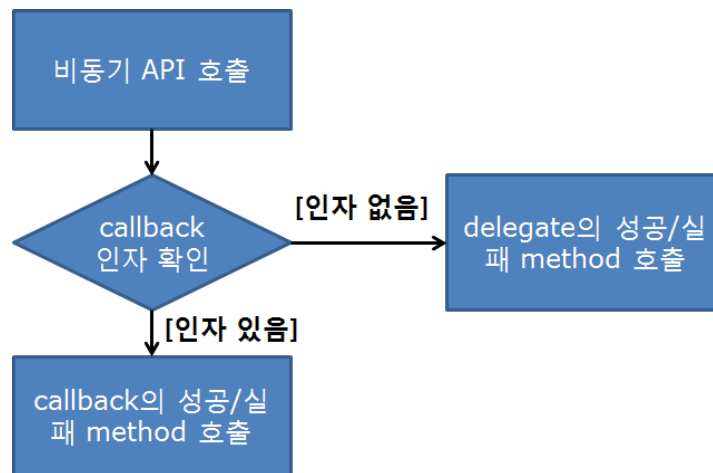


그림 11 비동기 API 호출 시 callback 여부에 따른 분기

다음은 샘플 앱에서 발췌한 코드이다.

```
autoLoginBtn.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // auto login
        isAutologin = true;
        PPIImpl.getInstance().autoLogin(Main.this, null);
    }
});
```

그림 12 샘플 앱에서 발췌한 콜백 없는 비동기 API 호출 코드 (자동로그인)

```
PPIImpl.getInstance().listFriend(paging, new ApiCallbackAdapter<PagingList<Friend>>() {
    @Override
    public void onError(Throwable e) {
        Toast.makeText(ListFriend.this, "Failed to retrieve friends list", Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onSuccess(PagingList<Friend> result) {
        List<Friend> list = result.getList();
        StringBuilder sb = new StringBuilder();

        for (int i = 0; i < list.size(); i++) {
            Friend friend = list.get(i);
            sb.append(friend.getNickname().toString() + "\n");
        }

        String str = sb.toString();
        if (str.length() != 0) {
            tv.setText(sb.toString());
        } else {
            tv.setText("None");
        }
    }
});
```

그림 13 샘플 앱에서 발췌한 콜백을 가진 비동기 API 호출 코드 (친구 목록 조회)

위 두 개의 그림 중 그림 12는 콜백을 가진 비동기 API 호출을 보여주고, 그림 13은 콜백이 없는 비동기 API 호출을 보여준다.

그림 12에선 콜백으로 null을 넘겼기 때문에 자동로그인에 성공한 경우엔 PPDelegate.onLogin() 이, 실패한 경우엔 PPDelegate.onLoginFail() 이 호출된다. 이와 달리 그림 13에선 콜백을 넘겼기 때문에 listFriend() 가 성공한 경우엔 콜백의 onSuccess() 가, 실패한 경우엔 콜백의 onError() 가 호출된다. 만약 그림 13의 listFriend에서 콜백으로 null을 넘겼다면 PPDelegate.onListFriend() 와 PPDelegate.onListFriendFail() 이 성공/실패 시 호출되었을 것이다.

## 4.4 SDK 상태 Listener

앱이 SDK의 상태 변화를 알아야 하는 경우를 위해 SDK 상태 Listener 기능이 제공됩니다.

com.pmangplus.core.PPStatusListener 인터페이스가 이러한 기능을 제공한다. com.pmangplus.ui.PPDelegate가 이 인터페이스를 확장하므로, 앱은 PPDelegate의 메서드에 관련 코드를 구현하면 된다. 앱은 이 인터페이스를 통해 멤버의 정보 변경, 로그아웃 또는 토큰 만료, SNS 연결 상태 변경 시 변경된 내용을 통보 받을 수 있다. SDK 초기화 시 PPDelegate를 SDK에 전달하므로 별도의 listener 등록 작업은 필요하지 않다.

## 4.5 로그 레벨 설정

PmangPlus SDK가 남기는 로그의 레벨은 PPConfig의 인자로 설정할 수 있다. 특별히 레벨을 지정하지 않을 경우의 기본 레벨은 WARN이다. 즉, WARN, ERROR 레벨의 로그만 남는다.

로그 레벨 설정은 eum 클래스인 com.pmangplus.core.LogLevel을 이용하며, DEBUG, INFO, WARN, ERROR이 제공된다. 각 레벨은 안드로이드의 Log 클래스가 제공하는 레벨과 동일한 의미를 갖는다.

다음은 DEBUG로 LogLevel을 설정한 예시이다. **출시 버전에선 반드시 LogLevel을 WARN 이상으로 설정할 것을 권장한다.**

```
public class PPApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();

        Handler handler = new Handler();

        // app registration information
        int appId = 1;
        String appKey = "test_app_key";
        String appSecret = "test_app_secret";

        // target pmangplus api server
        ApiServer targetApiServer = ApiServer.QA;

        // target country: have effect on SNS types and payment currency
        Country targetCountry = Country.GLOBAL;

        // other configuration information
        OptionalConfig opConfig = new OptionalConfig();
        opConfig.screenOrientation = ScreenOrientation.LANDSCAPE; // screen orientation. default value is ScreenOrientation.LANDSCAPE
        opConfig.isStandalone = false; // set flag whether player can use app without logging in PmangPlus. default value is false
        opConfig.pg = PG.PAYGATE; // set target payment gateway. ignore if app does not use pmangplus payment.
        opConfig.logLevel = LogLevel.DEBUG; // set log level for PmangPlus. should set WARN or ERROR on release version.

        final PPConfig config = new PPConfig(appId, appKey, appSecret, targetApiServer, targetCountry, opConfig);

        PPDelegate delegate = new PPDelegateImpl();

        // initialize PmangPlus
        PPImpl.init(config, delegate, getApplicationContext(), handler);
    }
}
```

그림 14 로그 레벨 설정 예시

## 4.6 사용자 게임 정보 Storage

SDK는 게임의 정보를 저장하는 Storage 기능을 제공한다. 단순한 Map의 형태로 저장되며, Map의 key와 value는 모두 String 타입이다. Storage는 앱 별로 제공된다.

관련 메서드는 PP.saveStorageData(), PP.listStorageKey(), PP.loadStorageData() 이다.



## 4.7 Standalone 모드 설정

반드시 로그인에 필요한 앱이 있는 반면(ex. 성인인증이 필요한 앱), 로그인에 실패한 경우에도 사용할 수 있어야 하는 앱도 있다. PPConfig의 OptionalConfig의 isStandalone을 true로 할 경우엔 PmangPlus의 로그인 창에 로그인 취소 버튼이 표시되며, 사용자는 취소 버튼을 눌러 로그인 하지 않은 상태로 앱을 사용할 수 있다. 반대로 isStandalone이 false인 경우엔 로그인 창에 취소 버튼이 표시되지 않으며, 사용자는 이 상태에서 반드시 로그인을 해야만 진행이 가능하다.

로그인 하지 않은 상태에서 기록한 점수, 도전과제 달성 기록은 내부 DB에 저장되어 다음 번 로그인 시 자동으로 서버에 전송된다.

## 4.8 PPDelegate 구현

PmangPlus SDK를 초기화하는 PPIimpl.init() 메서드를 호출할 때 인자로 넘기는 PPDelegate의 인스턴스는 다음의 용도로 활용된다.

- 4.3.3 비동기 API를 콜백 인자 없이 호출했을 경우 수행할 성공/실패 로직 제공.
- 4.4 SDK 상태 Listener의 로직 제공.

인터페이스 com.pmangplus.ui.PPDelegate는 모든 비동기 API의 성공/실패에 해당하는 메서드와 com.pmangplus.core.PPStatusListener의 메서드를 갖고 있어 구현하려면 매우 많은 작업이 필요하므로 작업의 편의를 위해 기본 구현체인 com.pmangplus.ui.PPDelegateImpl이 제공된다.

PPDelegateImpl의 기본 동작은 다음과 같다. 상세한 내용은 SDK에 포함된 소스코드를 참고한다.

- 로그인 성공, SNS 포스팅 성공, 점수 등록 성공, 도전과제 상태 변경 성공 시 PmangPlus 토스트 창을 표시한다.
- 자동로그인 실패 시 회원가입 창을 띄운다.

PPDelegateImpl은 기본적인 구현만을 제공하기 때문에 **앱에 맞게 PPDelegateImpl을 적절하게 수정해야 한다.** PPDelegate 인스턴스는 com.pmangplus.ui.PP.setDelegate() 메서드를 이용해 초기화 이후에도 새로 설정할 수 있다.

다음은 샘플앱에서 발췌한 PPDelegateImpl 구현 예시이다.



```

PPImpl.getInstance().setDelegate(new PPDelegateImpl() {

    @Override
    public void onLogin(Member member) {
        super.onLogin(member);
        setBtnEnabler(true);
        writeLog(member.getNickname() + " successfully logged in!");
    }

    @Override
    public void onLoginFail(boolean isAutoLogin, Activity act, Throwable e) {
        //check standalone
        if(isAutologin){
            if (PPImpl.getInstance().isStandalone()) {
                writeLog("login fail, Standalone set 0, can proceed game play" + e.getMessage());
            }else{
                writeLog("login fail, Standalone set X, cannot proceed game play!");
                writeLog("trying login...");
            }
        }
        super.onLoginFail(isAutoLogin, act, e);
    }
});

```

그림 15 샘플 앱에서 발췌한 PPDelegateImpl 설정 코드

**중요**

자동으로 익명계정을 생성하는 앱의 경우, 자동로그인에 실패하더라도 계정 생성 창이 뜨면 안 된다. 이 경우엔 PPDelegateImpl.onLoginFail() 을 override해서 아무런 동작도 수행하지 않도록 하면 된다. 다음은 override 예시이다.

```

PPDelegateImpl delegateForAnonymousApp = new PPDelegateImpl() {

    @Override

    public void onLoginFail(boolean isAutoLogin, Activity act, Throwable e) {

        //do nothing

    }

}

```

**중요**

SDK에 포함된 PPDelegateImpl의 소스코드를 직접 수정할 경우, 새로 배포된 SDK를 적용할 경우 수정된 코드가 삭제될 수 있으므로 **SDK 자체 코드는 수정하지 말고**, 익명 클래스 등을 이용해 동작을 변경한다.

## 5 연동 Step-By-Step

프로젝트의 생성에서 PmangPlus SDK 연동까지의 전체 과정을 개략적으로 설명하면 다음과 같다.

### 5.1 SDK Import

배포된 SDK를 workspace로 import한다. 이 과정은 본 문서의 3.2 절에 기술되어 있다. import를 마친 workspace의 모습은 다음과 같다.

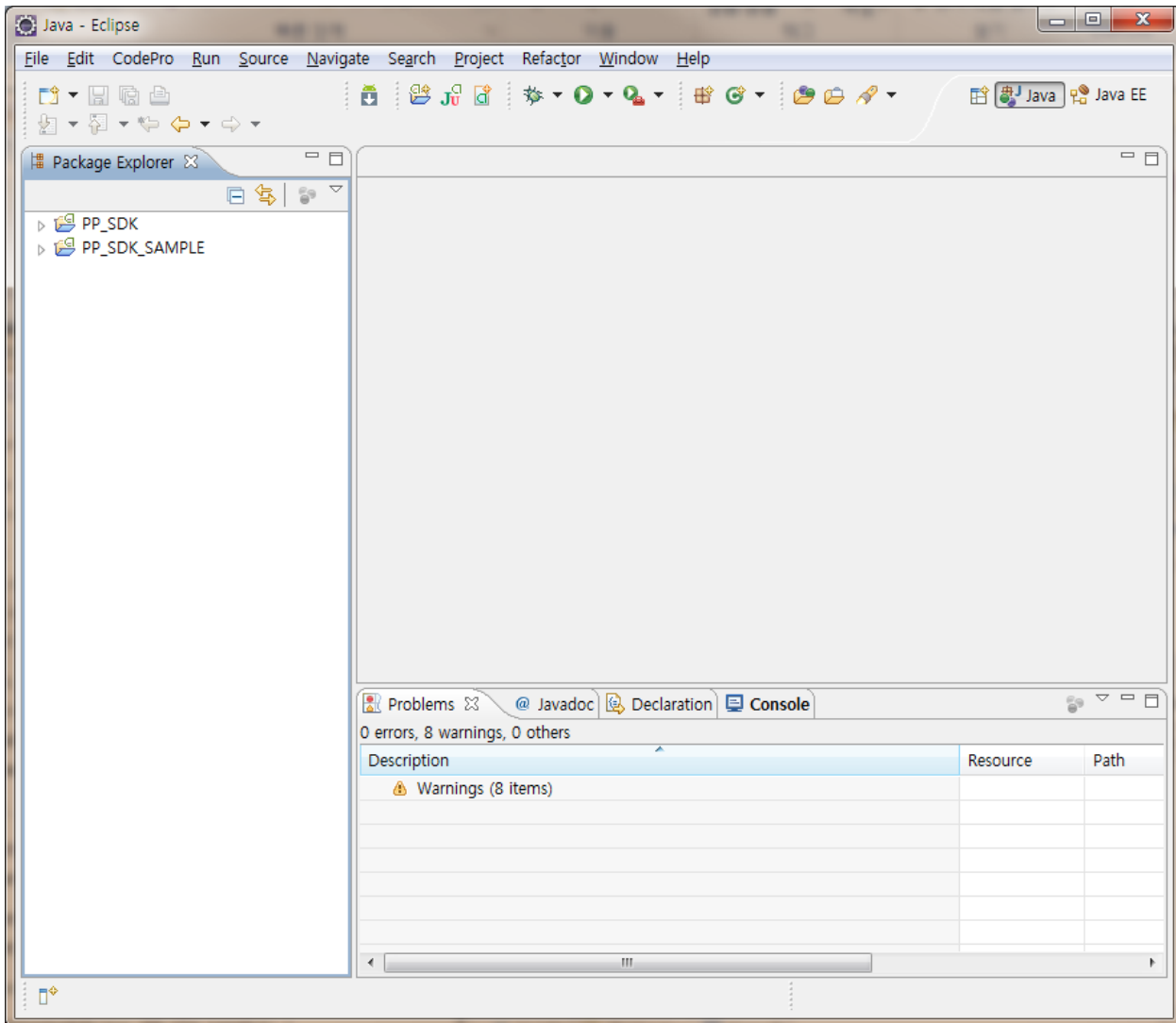


그림 16 SDK Import 를 마친 Workspace

## 5.2 프로젝트 생성

PP와 연동할 프로젝트를 생성한다. 기존에 개발중인 프로젝트에 SDK를 연동한다면 본 단계는 생략한다.

1. 메뉴에서 File > New > Android Project를 선택한다.

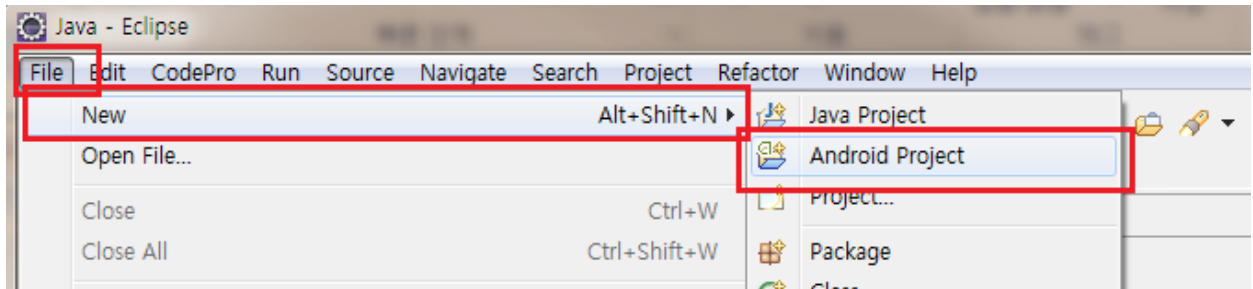


그림 17 안드로이드 프로젝트 생성

2. 프로젝트 생성 대화창이 표시되면, Project name과 Properties에 적절한 값을 입력한다. 참고로 SDK의 최소 API 레벨이 7이기 때문에 앱도 API 레벨 7 이상이어야 한다.

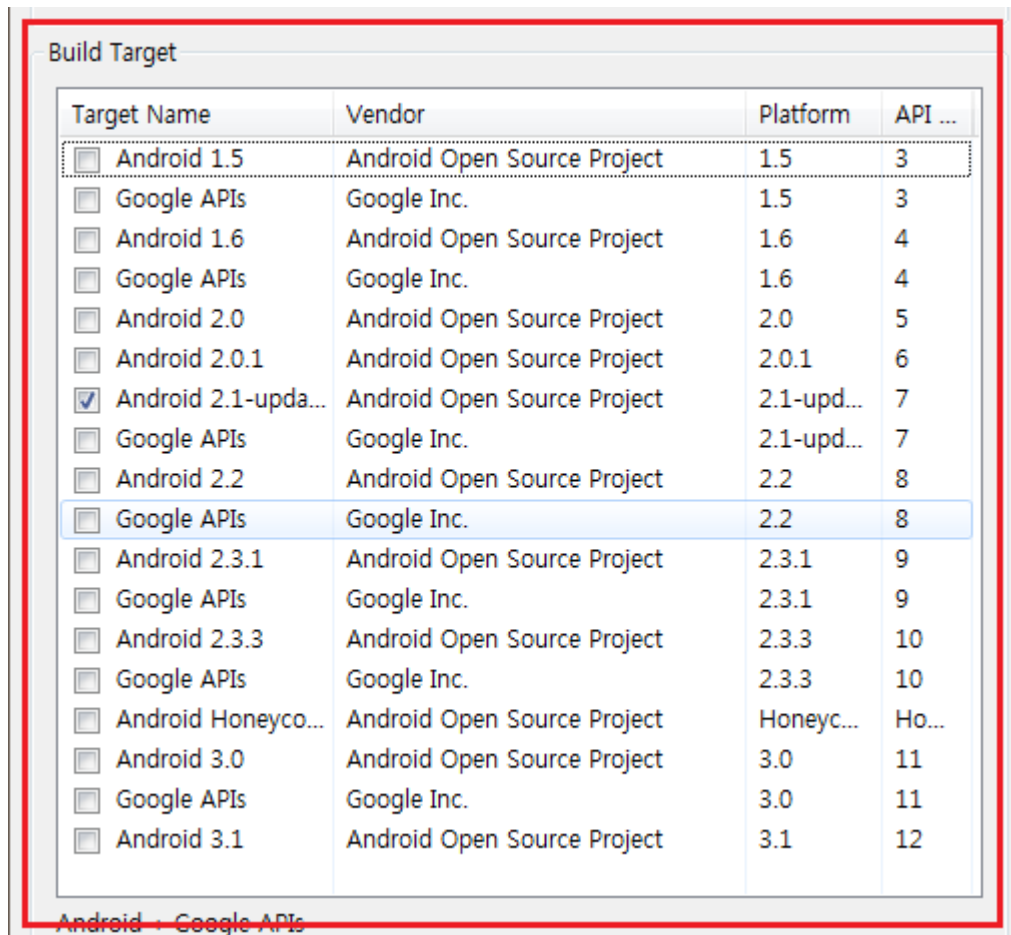


그림 18 새 프로젝트의 Build Target 설정 - API level 7 이상이어야 함

3. 생성한 프로젝트의 Java 컴파일 설정을 1.6 Compliance 하도록 설정한다. (Project > Properties > Java Compiler > JDK Compliance )

### 5.3 PP\_SDK 를 라이브러리로 지정

1. 연동 프로젝트에 PP\_SDK 프로젝트를 의존 라이브러리로 지정한다. 의존 라이브러리를 등록하기 위해 Project > Properties > Android > Library > Add... > PP\_SDK 를 선택한다. 미리 SDK를 import해야 Add... 를 했을 때 PP\_SDK가 표시된다.

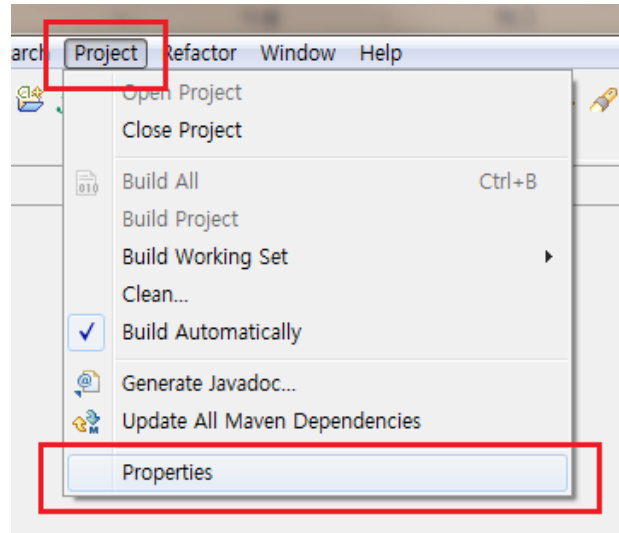


그림 19 Project > Properties 메뉴

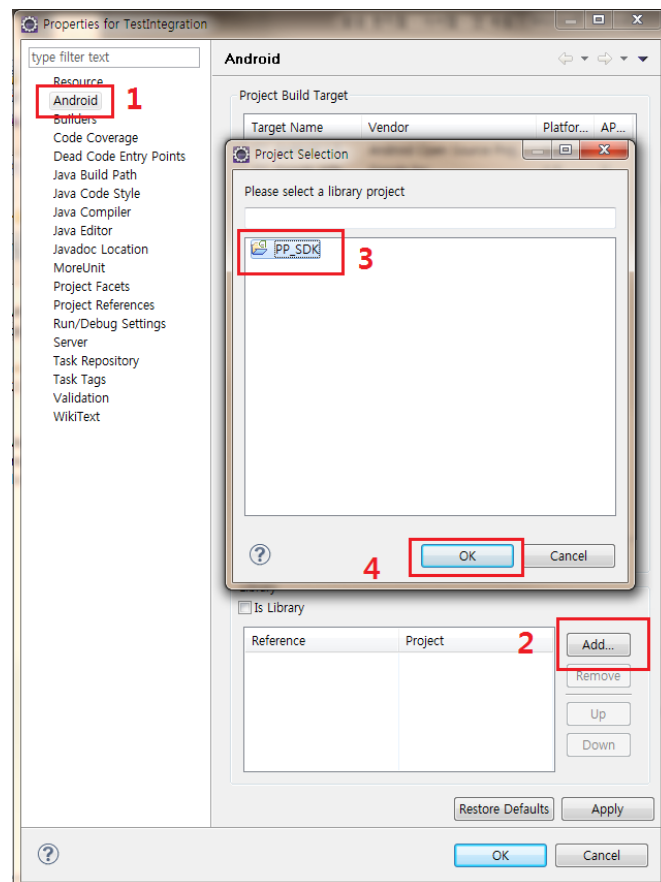


그림 20 Android Library Project 추가 메뉴 - 명시한 숫자는 메뉴 선택 순서임

2. Library 목록에 PP\_SDK가 표시되고, 녹색 마크가 붙는지 확인한다.

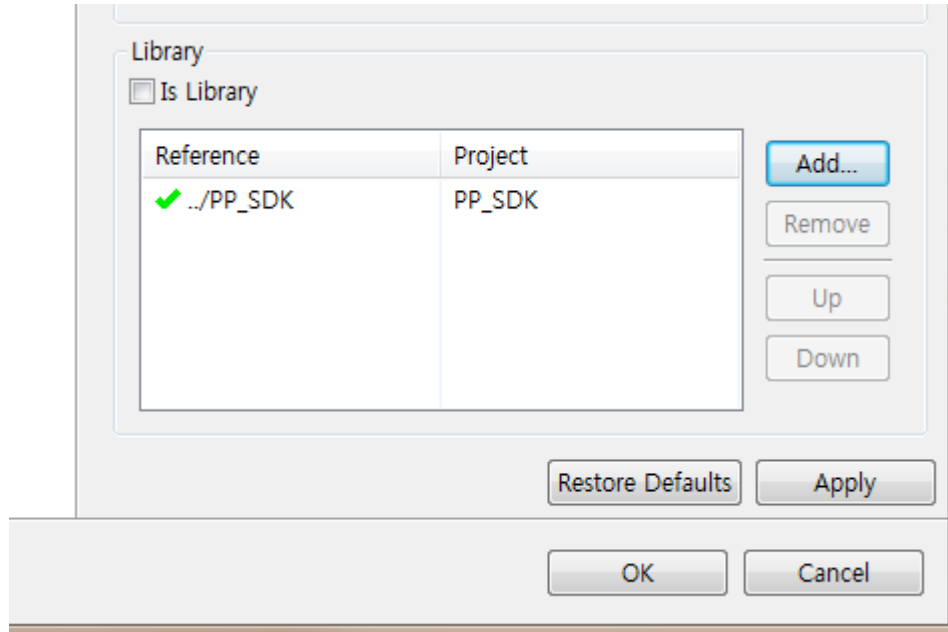


그림 21 정상적으로 PP\_SDK 라이브러리 프로젝트를 참조한 경우

3. 정상적으로 PP\_SDK를 라이브러리로 등록한 경우, 프로젝트의 Project Explorer 창에 PP\_SDK\_src 라는 항목이 추가된다.

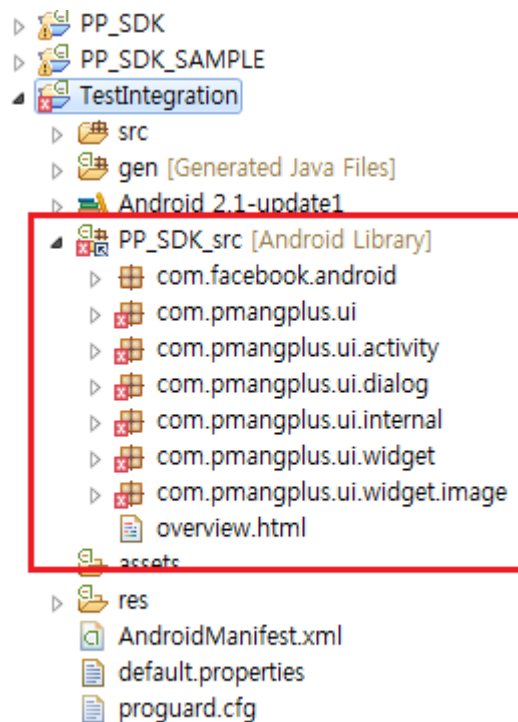


그림 22 추가된 PP\_SDK\_src 항목. 컴파일 에러가 난다.

4. PP\_SDK\_src는 아직 build path 문제로 컴파일이 되지 않는다. 컴파일 문제를 해결하기 위해 PP\_SDK/libs에 포함된 jar를 java build path에 추가한다. java build path에 추가하기 위해 Project > Properties > Java Build Path > Libraries > Add JARs... 를 선택한다.

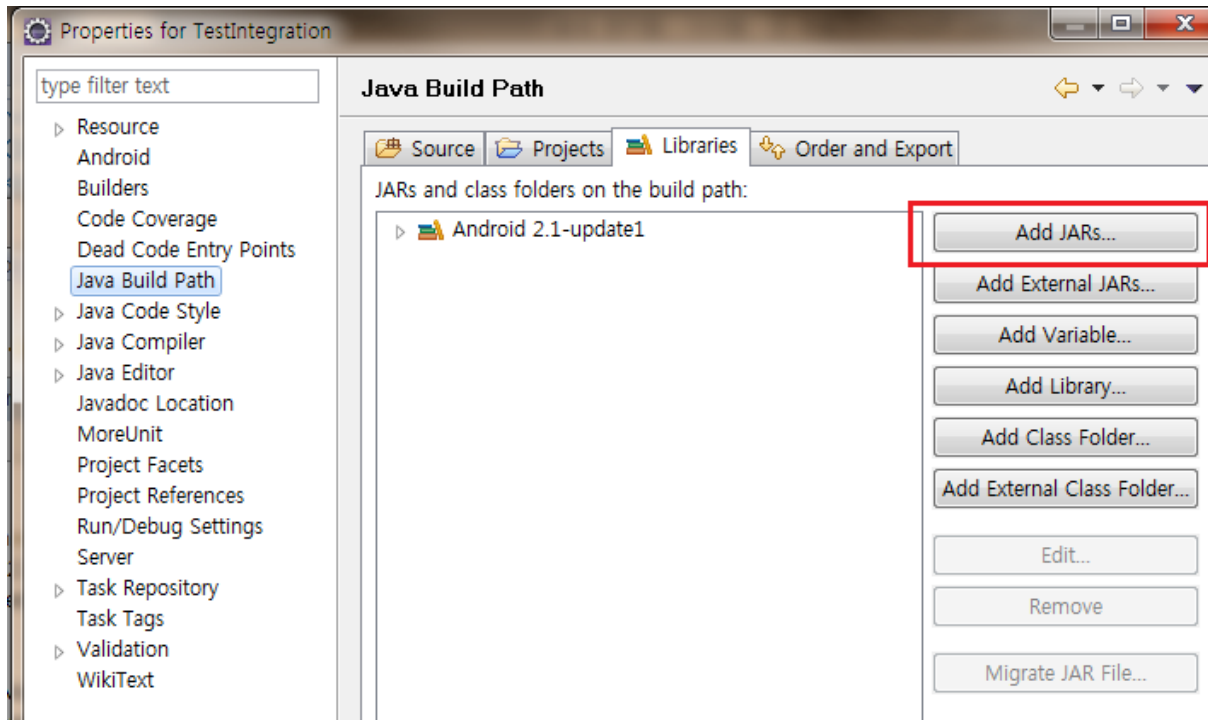


그림 23 Java Build Path 메뉴에서 Add JARs... 선택

5. Add JARs... 선택 후 나오는 대화창에서 PP\_SDK > libs 폴더 밑에 있는 모든 jar를 선택한 후, OK를 클릭한다.

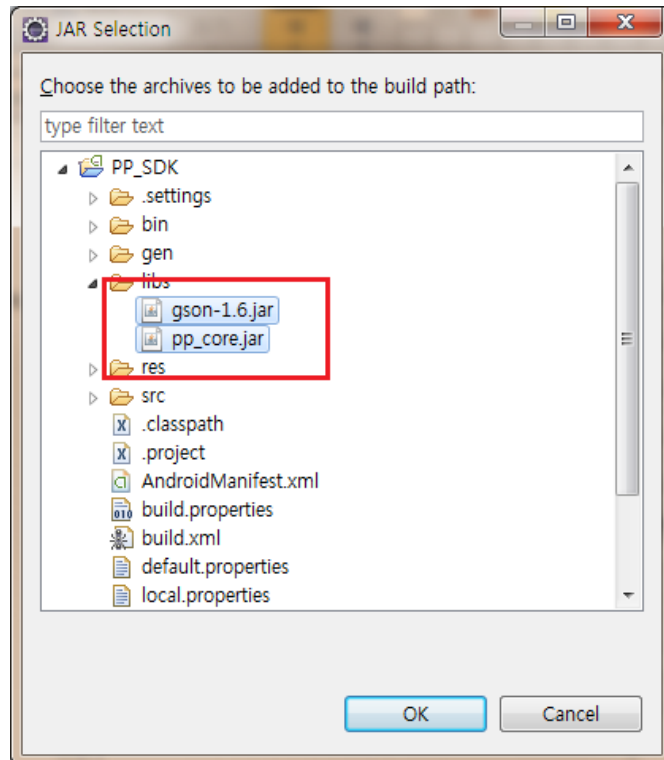


그림 24 PP\_SDK &gt; libs 하위의 jar 선택

6. 이제 Libraries 목록에 두 개의 jar가 표시된 것을 확인하고, OK를 클릭해서 설정창을 빠져나온다.

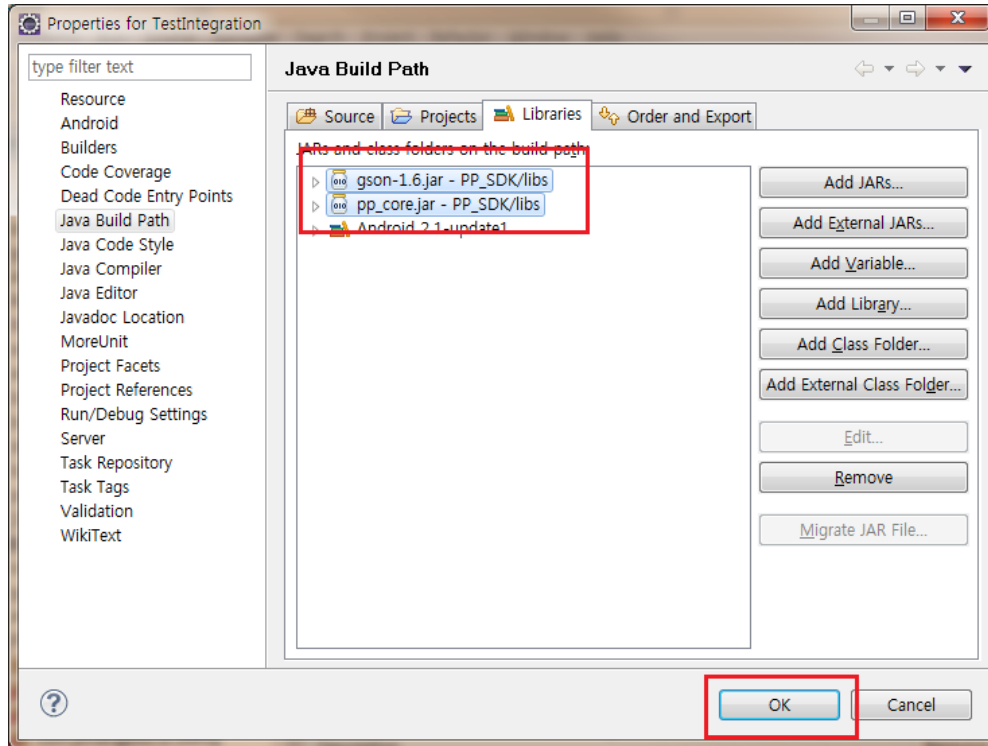


그림 25 두 개의 jar를 추가한 결과

7. jar를 추가했는데도 compile error가 발생한다면 encoding 차이를 의심해 봐야 한다. PP\_SDK는 utf-8로 작성되었기 때문에 앱이 utf-8 이외의 encoding을 사용한다면 다음과 같이 소스코드가 깨져 보이며, 컴파일 에러가 발생할 수 있다. 이럴 땐 PP\_SDK\_src 항목의 encoding을 명시적으로 UTF-8로 맞추면 된다.

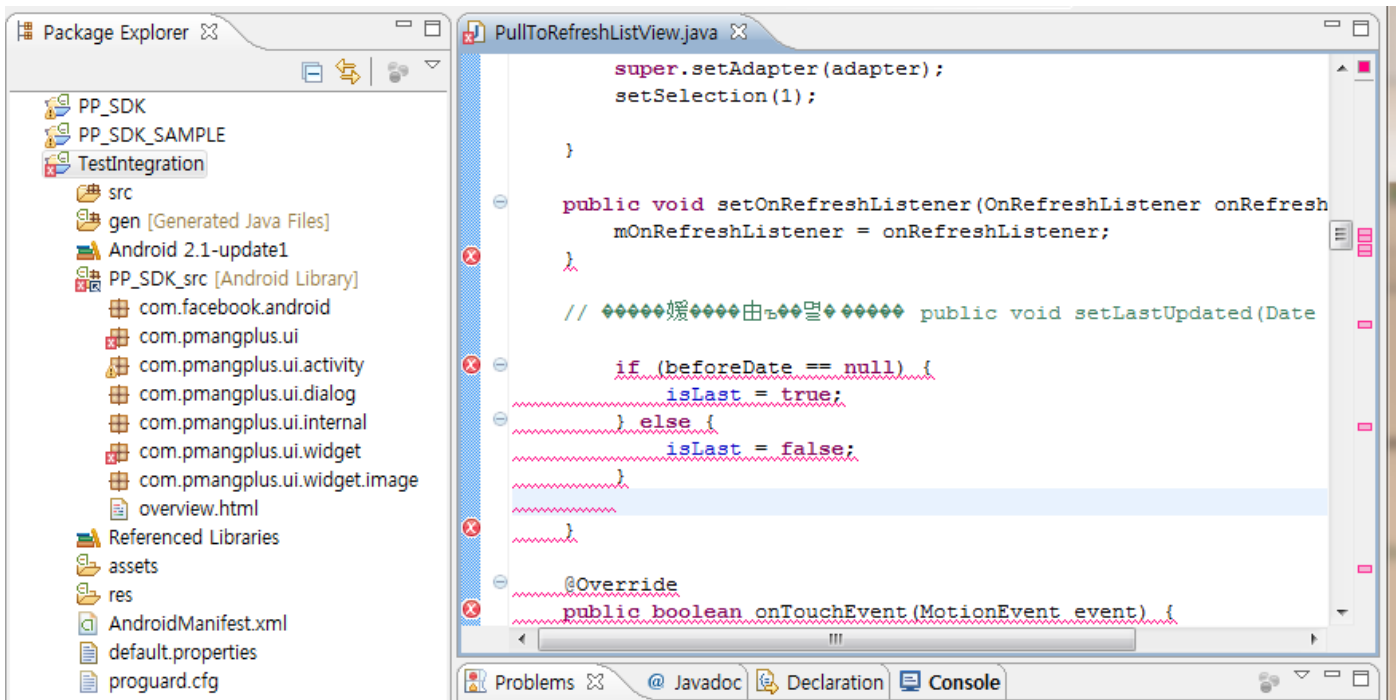


그림 26 인코딩이 맞지 않아 코드가 깨져보이는 상황 (PP\_SDK는 UTF-8, TestIntegration은 EUC\_KR)

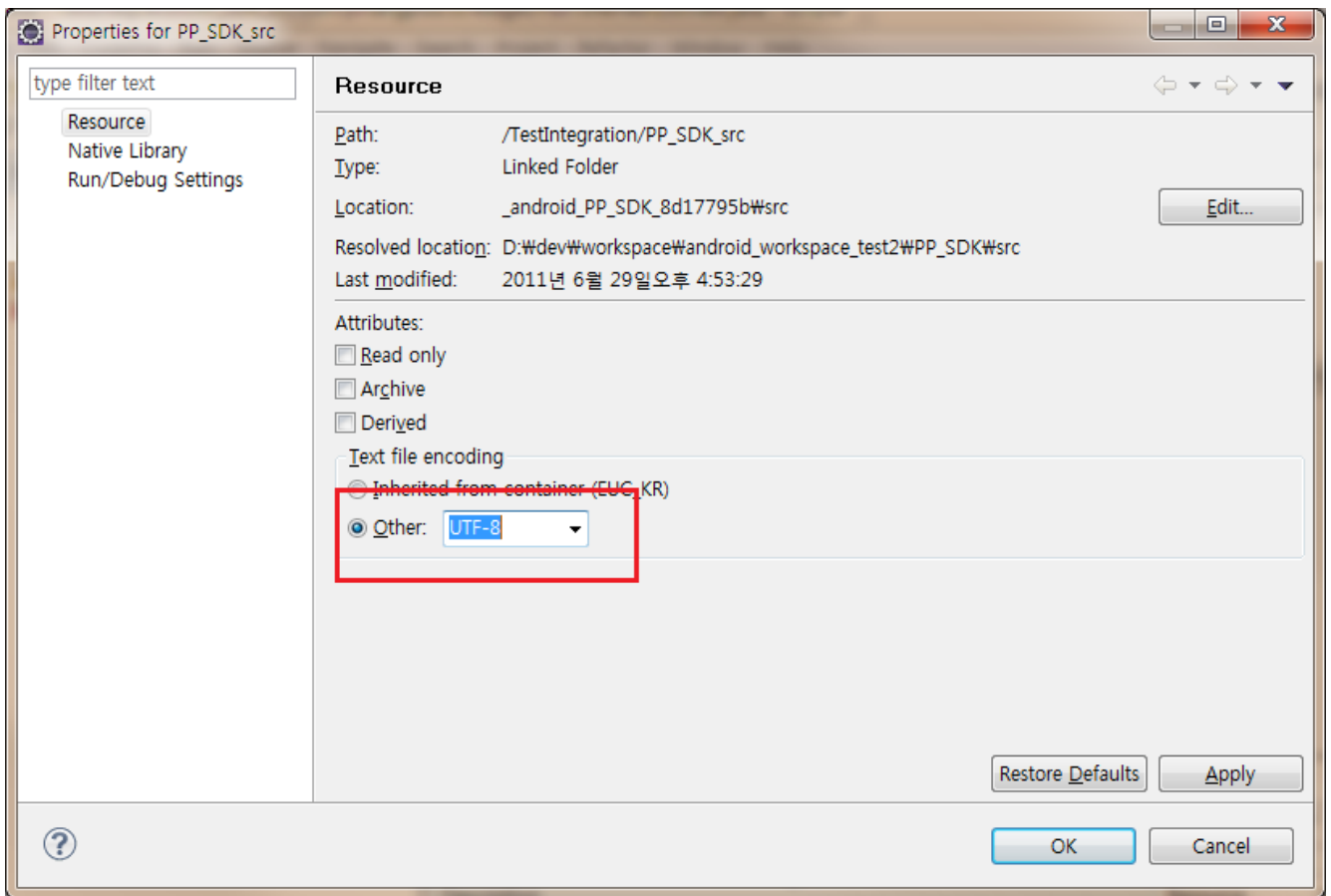


그림 27 PP\_SDK\_src의 인코딩을 UTF-8로 지정

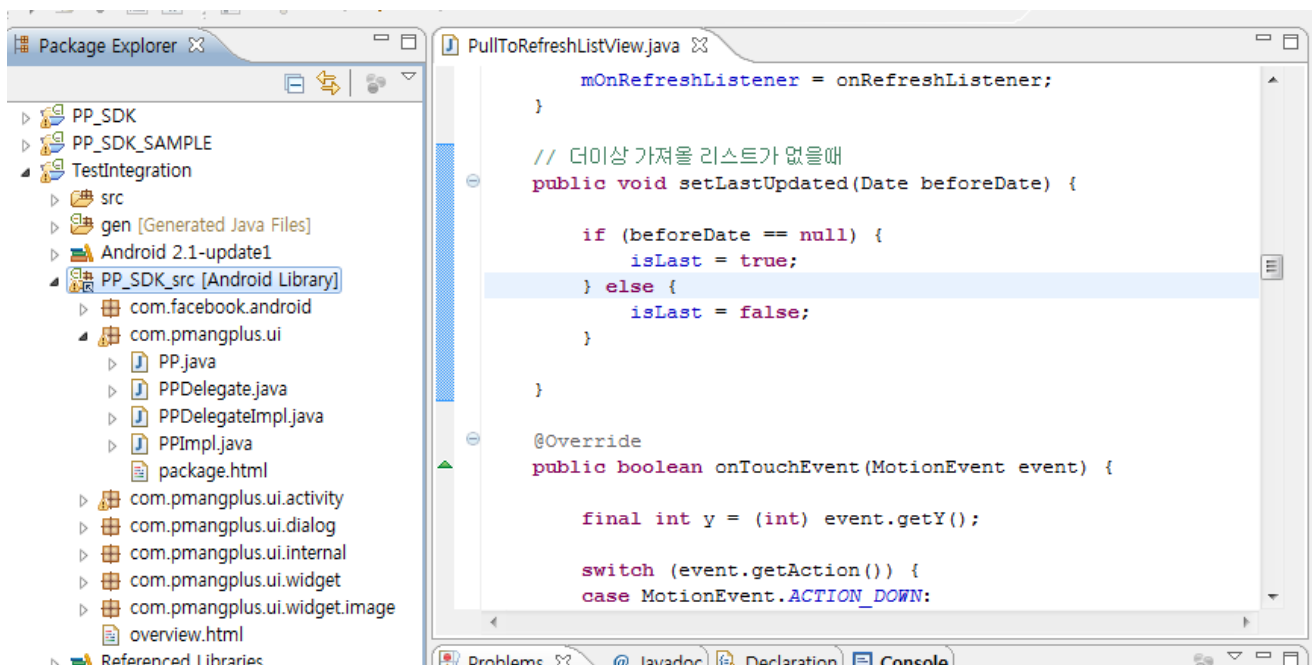


그림 28 인코딩 문제가 해결된 결과



## 5.4 연동 코드 작성

1. TestIntegration 프로젝트의 AndroidManifest.xml에 PP\_SDK/AndroidManifest.xml의 내용을 추가한다. 이 내용은 본 문서의 4.1을 참고한다.
2. 본 예제에선 자동로그인과 대시보드 표출만을 확인한다. 애플리케이션의 최초 실행 Activity를 Main으로 설정한 경우, 결과 코드는 다음과 같다. (**참고:** 아래 코드는 단순 참고용이다. 실제 앱에선 반드시 Application에서 초기화를 해야 한다.)

```
package com.pmangplus.test;

import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Toast;

import com.pmangplus.core.PPConfig;
import com.pmangplus.core.model.ScreenOrientation;
import com.pmangplus.ui.PPDelegateImpl;
import com.pmangplus.ui.PPImpl;

public class Main extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // 초기화

        // 앱 등록 시 발급받은 정보
        int appId = 1;
        String appKey = "test_app_key";
        String appSecret = "test_app_secret";

        // 연동 대상 서버
        ApiServer targetApiServer = ApiServer.QA;

        // 출시 대상 국가 : SNS 설정과 결제 통화에 영향을 미침
        Country targetCountry = Country.GLOBAL;

        // 기타 설정
        OptionalConfig opConfig = new OptionalConfig();
        opConfig.screenOrientation = ScreenOrientation.LANDSCAPE; // 화면 방향. 기본값은
ScreenOrientation.LANDSCAPE
        opConfig.isStandalone = false; // PmangPlus에 로그인하지 않은 상태에서 플레이 가능 여부 설정. 기본
값은 false
        opConfig.pg = PG.PAYGATE; // 결제 대상 PG 설정. 결제를 사용하지 않을 경우 무시해도 무방함
        opConfig.logLevel = LogLevel.DEBUG; // PmangPlus의 로그 레벨. release 버전에선 WARN 이상으로
설정을 권장함
    }
}
```

```

final PPConfig config = new PPConfig(appId, appKey, appSecret, targetApiServer,
targetCountry, opConfig);
PPImp1.init(testAppConfig, new PPDelegateImpl(),
    getApplicationContext(), new Handler());
PPImp1.getInstance().autoLogin(Main.this, null);

// 대시보드 표시 버튼 listener
findViewById(R.id.btn_dashboard).setOnClickListener(
    new OnClickListener() {

        @Override
        public void onClick(View v) {
            if (PPImp1.getInstance().isLogin()) {
                PPImp1.getInstance().openDashboard(Main.this);
            } else {
                Toast.makeText(getApplicationContext(), "로그인하세요",
                    Toast.LENGTH_SHORT).show();
            }
        }
    });
}
}

```

리스트 2. 자동로그인과 대시보드 호출하는 Main.java

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button android:id="@+id/btn_dashboard" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="대시보드" />
</LinearLayout>

```

리스트 3. 대시보드 호출 버튼을 가진 res/layout/main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.pmangplus.test"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="7" />
    <!-- required permission for pmangplus -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <!-- required feature for pmangplus -->
    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Main"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
    </application>

```

```

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<!-- put custom content -->

<!-- pmangplus activity -->
<activity android:name="com.pmangplus.ui.activity.PmangPlusMain"
android:label="@string/app_name"
    android:theme="@style/pp_main" android:launchMode="singleTask"
    android:clearTaskOnLaunch="true" android:configChanges="orientation|keyboardHidden" />
    <activity android:name="com.pmangplus.ui.activity.PPMainGame"
    android:label="@string/app_name" android:theme="@style/pp_main"
        android:configChanges="orientation|keyboardHidden" />
    <activity android:name="com.pmangplus.ui.activity.PPGameInfo"
    android:label="@string/app_name" android:theme="@style/pp_main"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPFriendsFriends"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPMainFriend"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPFriendInfo"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPMainMe"
    android:label="@string/app_name" android:theme="@style/pp_main"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPMainPP"
    android:label="@string/app_name" android:theme="@style/pp_main"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPLeaderBoardList"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden" />
    <activity android:name="com.pmangplus.ui.activity.PPLeaderBoard"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden" />
    <activity android:name="com.pmangplus.ui.activity.PPSelectDistrict"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPSelectCounty"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPAchievementMain"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden" />
    <activity android:name="com.pmangplus.ui.activity.PPCompareFriend"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPMessageMain"
    android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.activity.PPMessageList"
    android:label="@string/app_name"

```

```

        android:theme="@style/pp_main"
        android:windowSoftInputMode="stateVisible|adjustResize"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPFriendRequested"
        android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPFindPPFriend"
        android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPFindSnsFriend"
        android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPChooseEmoticon"
        android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPMemberSetting"
        android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPPostOnSns"
        android:label="@string/app_name" android:theme="@style/pp_main"
        android:windowSoftInputMode="adjustPan|stateAlwaysVisible"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPMemberSettingInput"
        android:label="@string/app_name"
        android:theme="@style/pp_main"
        android:windowSoftInputMode="adjustResize|stateAlwaysVisible"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPError" android:label="@string/app_name"
        android:theme="@style/pp_main.translucent"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPEula" android:label="@string/app_name"
        android:theme="@style/pp_main"
        android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPNoticeList"
        android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.activity.PPPaymentActivity"
        android:label="@string/app_name"
        android:theme="@style/pp_main" android:configChanges="orientation|keyboardHidden" />

<!-- pmangplus dialog -->
        <activity android:name="com.pmangplus.ui.dialog.PPSignup" android:theme="@style/pp_dialog"
        android:configChanges="orientation|keyboardHidden"
        android:windowSoftInputMode="stateHidden" />
        <activity android:name="com.pmangplus.ui.dialog.PPLogin" android:theme="@style/pp_dialog"
        android:configChanges="orientation|keyboardHidden"
        android:windowSoftInputMode="stateAlwaysVisible"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.dialog.PPResetPasswd"
        android:theme="@style/pp_dialog"
        android:configChanges="orientation|keyboardHidden"
        android:windowSoftInputMode="stateAlwaysVisible"
        android:screenOrientation="behind" />
        <activity android:name="com.pmangplus.ui.dialog.PPAdultApprove"

```

```

android:theme="@style/pp_dialog"
    android:configChanges="orientation|keyboardHidden"
android:windowSoftInputMode="stateAlwaysVisible" />
    <activity android:name="com.pmangplus.ui.dialog.PPCheckPasswd"
android:theme="@style/pp_dialog"
    android:configChanges="orientation|keyboardHidden"
android:windowSoftInputMode="stateAlwaysVisible" />
    <activity android:name="com.pmangplus.ui.dialog.PPTwtLogin"
android:theme="@style/pp_dialog"
    android:configChanges="orientation|keyboardHidden"
android:windowSoftInputMode="stateAlwaysVisible"
    android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.dialog.PPFbLogin" android:theme="@style/pp_dialog"
    android:configChanges="orientation|keyboardHidden"
android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.dialog.PPMemberMerge"
android:theme="@style/pp_dialog"
    android:configChanges="orientation|keyboardHidden"
android:windowSoftInputMode="stateAlwaysVisible"
    android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.dialog.PPMemberMergeOk"
android:theme="@style/pp_dialog"
    android:configChanges="orientation|keyboardHidden"
android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.dialog.PPOptionalInfo"
android:theme="@style/pp_dialog"
    android:configChanges="orientation|keyboardHidden"
android:screenOrientation="behind" />
    <activity android:name="com.pmangplus.ui.dialog.PPNotiDialog"
android:theme="@style/pp_dialog"
    android:configChanges="orientation|keyboardHidden" />

</application>
</manifest>

```

리스트 4. AndroidManifest.xml 전체 (주의: 위 내용은 PP\_SDK 버전에 따라 달라질 수 있음)

## 6 Troubleshooting

다음은 자주 발생하는 문제에 대한 해결책이다.

- R 파일이 없거나, gen 디렉터리가 없다는 error가 발생하면 전체 프로젝트를 clean해 본다.
- 간혹 class가 있는데도 eclipse상에서 compile이 실패했다는 에러가 발생할 수 있다. 이 경우 project를 clean하거나, Eclipse를 재시동해서 확인한다.
- @Override 때문에 컴파일에 실패할 경우, Java Compiler의 JDK Compliance 세팅을 1.6으로 수정한다. 다음은 JDK 설정 예시이다.

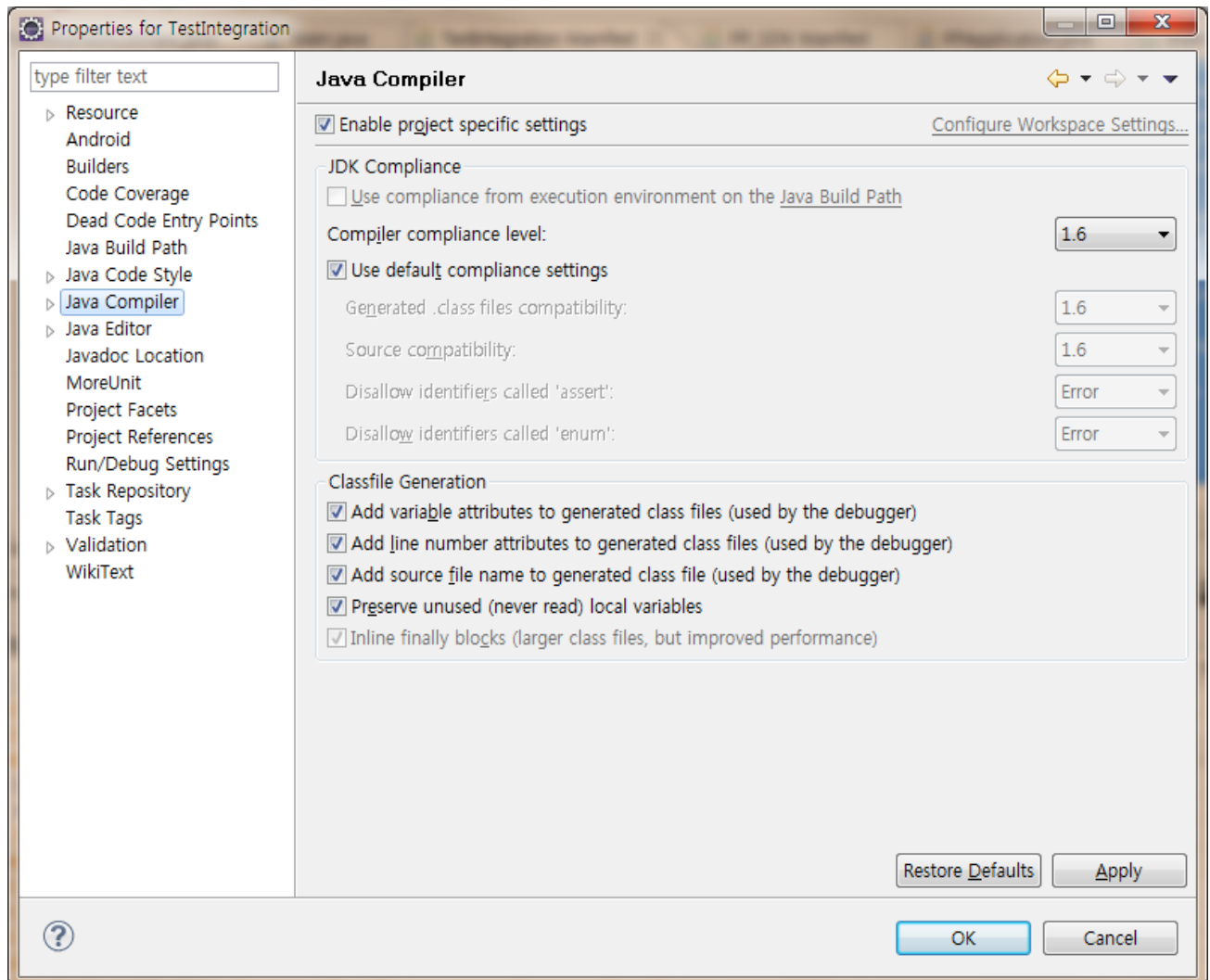


그림 29 JDK 1.6 Compliance 한 Java Compiler 설정