

TOP Android SDK 使用指南

SDK Version: android1.0.1

2012-11-7

目 录

准备工作.....	1
创建应用.....	1
准备开发环境.....	1
准备编码.....	2
创建 Android 工程.....	2
注册 APK 签名	5
开始开发.....	6
启用 App Key	6
用户授权.....	6
调用 API	7
1. 调用 REST API.....	7
2. 调用 TQL	9
Refresh Token	10
Sample 使用说明.....	10
Android 平台支持.....	11
技术支持.....	12
Change Log.....	12

准备工作

创建应用

创建一个新的应用，参考文档：<http://open.taobao.com/doc/detail.htm?id=66>

应用名称：
不可超过15个中文字符

应用标签：应用标签是应用的关键标识，在应用创建成功后将无法修改，根据以下标签说明选择正确的标签，有助于应用审核人员快速定位和审核您的应用

▼ 不需要发布到淘宝平台进行推广：

☐ 淘宝客网站

☒ **无线买家应用**（面向消费者的无线应用，主要用于提供商品导购服务，比如：**果库**。面向卖家的无线应用，请选择**在线订购应用**接入，上架类目选择**无线专区**）

▼ 需要发布到淘宝平台进行推广👉：

☐ 买家应用

☒ **在线订购应用**

☐ 店铺模块应用

如果此无线应用针对买家需求开发上图的应用标签请选择“**无线买家应用**”。如果是针对卖家需求开发请选择“**在线订购应用**”，发布服务时类目选择“无线专区”。

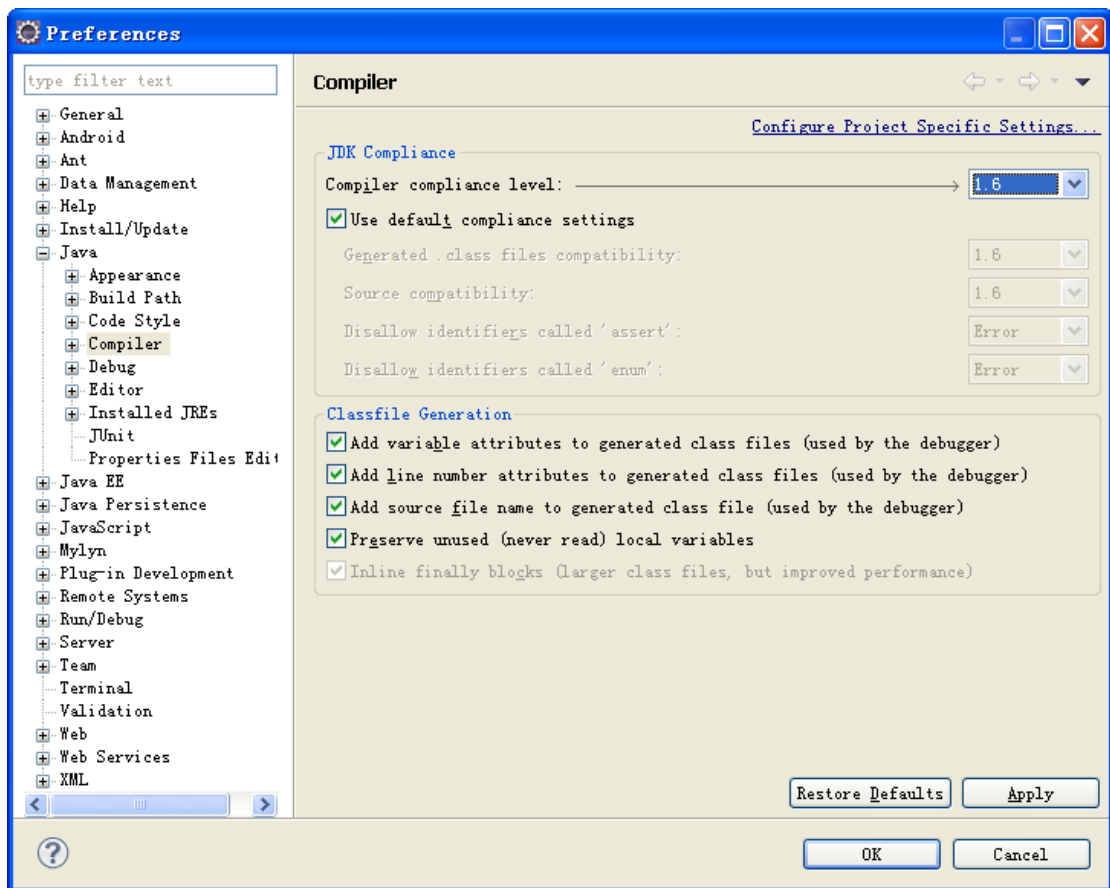
创建完应用后需要在“开发测试”页面设置应用属性，这里需要提及的是“回调页面 URL”的配置

回调页面URL：
即供用户访问的应用地址。通过该地址，应用可以获得用户nick、SessionKey等信息。
填好URL后，请点击“测试”来检查该URL是否可以正常获取淘宝用户授权，及被访问。
详见：[通过回调页面URL进行用户验证和授权](#)

这个很重要，这个会影响到用户授权后浏览器对 APP 的回调。建议使用应用的包名作为 scheme，这样可以尽量保证 scheme 的唯一性。

准备开发环境

1. 安装 JDK (1.6 及以上版本)
2. 安装 Eclipse
3. 检查 Eclipse 的编译级别，请选择 1.6

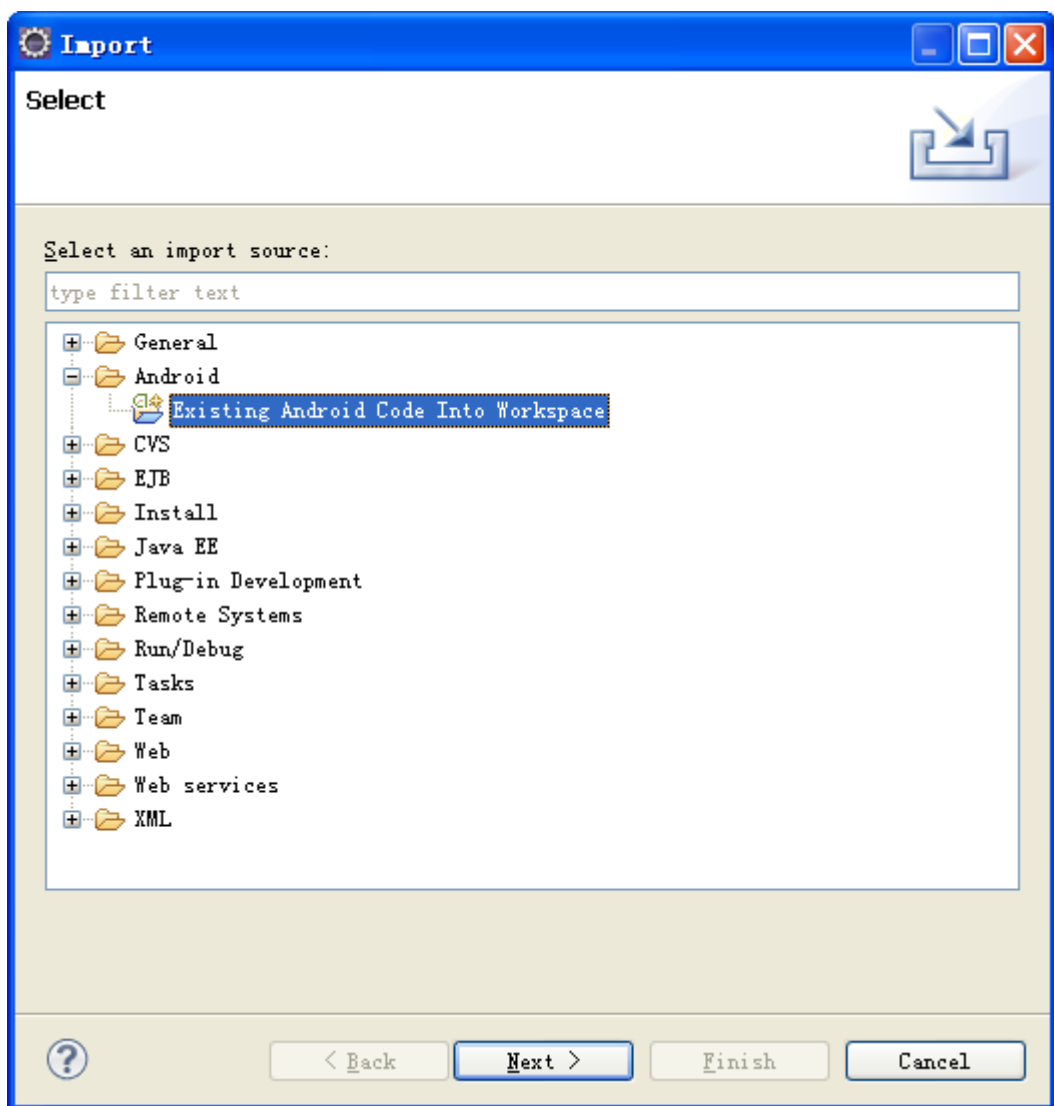


4. 安装 Android SDK 及 Eclipse 插件。具体参考：
<http://developer.android.com/sdk/installing/index.html>
5. TOP Android SDK 需要 Android 2.1 支持，请先使用 Android SDK Manager 进行下载。缺少 Android 2.1 时 TOP Android SDK 导入 Eclipse 后会出现红叉。
6. 下载 TOP Android SDK
<http://open.taobao.com/doc/detail.htm?spm=0.0.0.21.580f74&id=34>

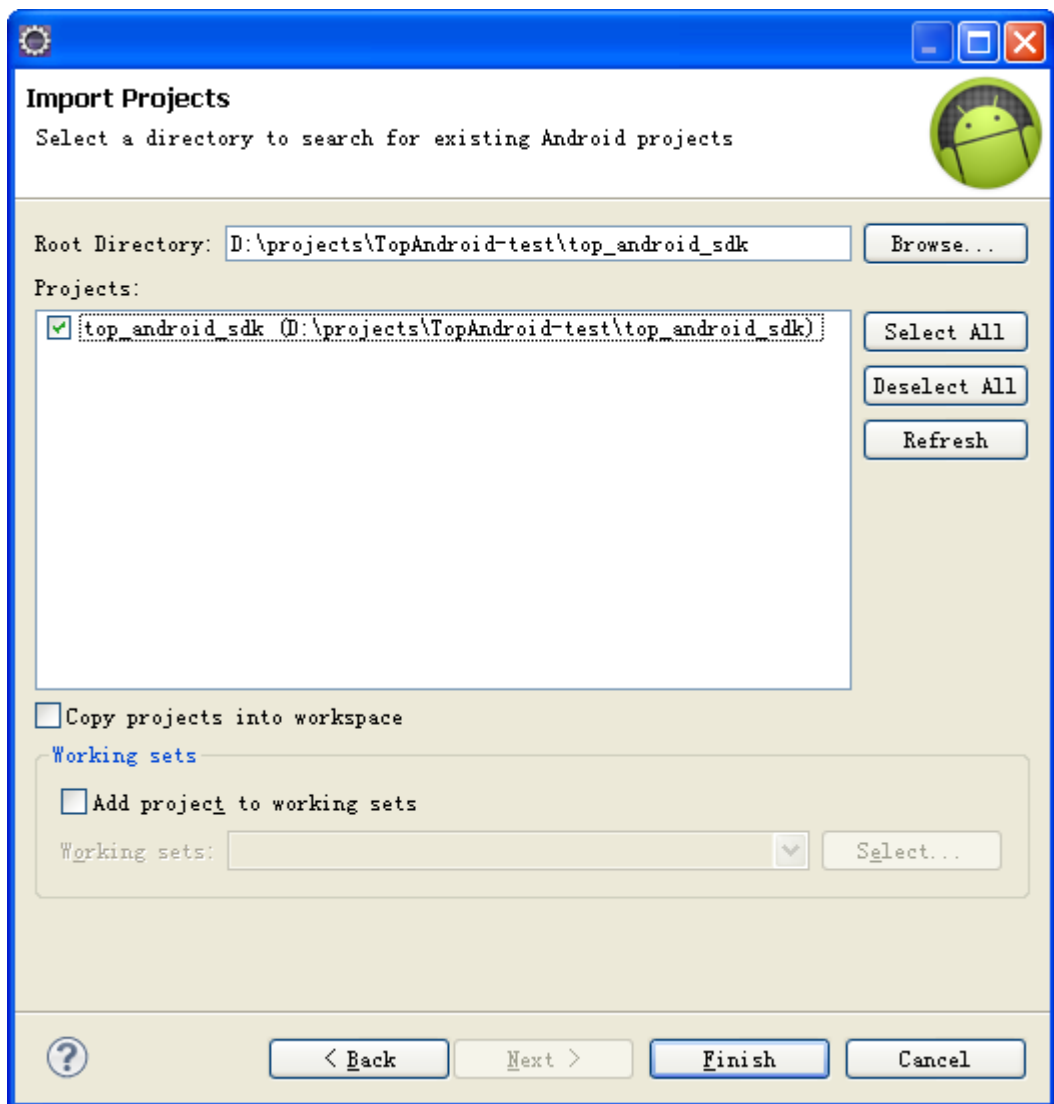
准备编码

创建 Android 工程

1. 将 TOP Android SDK 导入 Eclipse。导入之前请不要把 TOP Android SDK 代码放在 Eclipse workspace 目录下。
Eclipse->File->Import



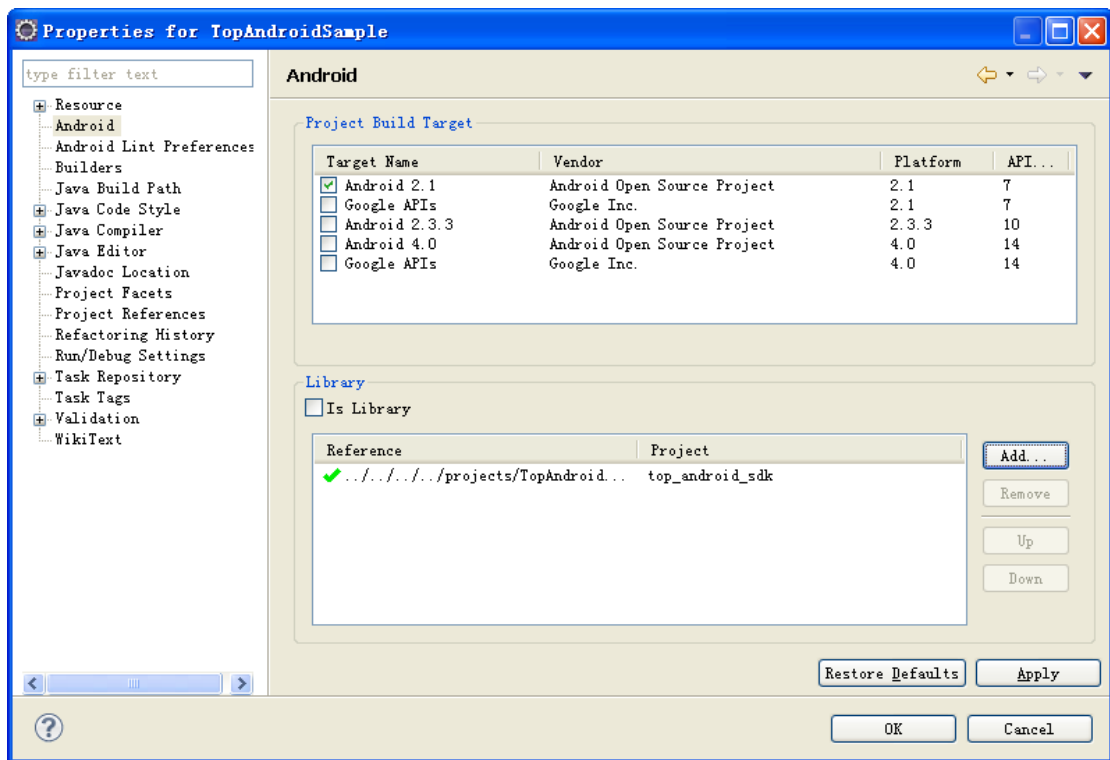
选择 SDK 下载路径，导入 SDK 到 Eclipse



将工程代码文件编码设为 UTF-8。(Project->Properties->Resource)

2. 创建一个新的 Android 工程，添加对 TOP Android SDK 的引用。

Project->Properties->Android->Library



注册 APK 签名

出于安全考虑开放平台要求你在“开发者中心”注册你的应用的签名。可以在 SDK 下载包里找到获取签名的工具：apk-parser。

使用方式：执行 parse.bat 或 parse.sh 后面添加 apk 文件路径参数。

```
D:\wbox_mount\apk-parser>parse.bat D:\wbox_mount\TopAndroid.apk
apk签名: -758972822
```

在“Android 应用认证”页面将上面工具中得到签名填入文本框中。



注意：当你的应用正式发布的时候，一般都会重新用 RSA 私钥对 APK 进行签名，这个时候一定不要忘了重新获取一下 APK 的签名并注册到“开发者中心”。

开始开发

启用 App Key

TopAndroidClient 是使用 TOP Android SDK 的入口，在前面申请得到的 App Key 需要注册到此类中才能进行后续的授权及 API 调用操作。

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

        TopAndroidClient.registerAndroidClient(getApplicationContext(),
            "519255", "988d57871c1fb8767a9b0875b28e5c17",
            "callback://authresult");
    }
}
```

PS: TopAndroidClient 支持同时注册多个 App Key。

获得 TopAndroidClient 实例:

```
TopAndroidClient
client=TopAndroidClient.getAndroidClientByAppKey("519255");
```

用户授权

出于安全考虑第三方应用进行用户授权操作时必须在浏览器上进行。只需要调用以下代码即可实现:

```
client.authorize(MainActivity.this);
```

TopAndroidClient 对象需要传入一个 Activity 实例来完成对浏览器的启动。

无论用户授权成功或者失败浏览器都会回调你的应用。为了接收回调的数据请按以下步骤进行操作:

1. 继承 com.taobao.top.android.auth.AuthActivity 实现一个 Activity 完成授权以后的后续操作。这个子类需要实现两个方法，这两个方法的作用请见方法注释和原代码。

```
/**
 * 待子类实现。需要返回{@code TopAndroidClient}对象，用来接收授权成功后的
access token
 * @return
 */
protected abstract TopAndroidClient getTopAndroidClient();

/**
 * 待子类实现。需要返回一个处理授权结果的监听器
 * @see AuthorizeListener
 * @return
```



```
*/
protected abstract AuthorizeListener getAuthorizeListener();
```

2. 在 AndroidManifest.xml 中配置这个 Activity

```
<activity
    android:name="xxxActivity"
    android:configChanges="keyboardHidden|orientation"
    android:label="@string/title_activity_main" >
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT"
    />

    <category
android:name="android.intent.category.BROWSABLE" />
<!-- 注意: scheme的值一定要和开发者中心注册的回调url的"://"之前部分保持一致 -->
        <data android:scheme="callback"/>
    </intent-filter>
</activity>
```

另外需要申请以下三个权限

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

调用 API

1. 调用 REST API

```
/**
 * 调用TOP API
 *
 * @see TopApiListener
 * @param params
 *         系统及业务参数
 * @param userId
 *         需要使用哪个用户授权的access token来调用api, 当API不需要
session key时此参数可以为null
 * @param listener
 *         api调用回调处理监听器, 不能为null
 * @param async
 *         true:异步调用; false:同步调用。Android 3.0以后会限制在UI主线
```

程中同步访问网络，使用同步方式需谨慎

```
* @throws IllegalArgumentException
*      当参数<code>params</code>或<code>listener</code>为null
时
*/
public void api(final TopParameters params, final Long userId,
               final TopApiListener listener, final boolean async)
```

示例代码：

```
TopParameters params=new TopParameters();
params.setMethod("taobao.user.buyer.get");//API方法
params.addFields("nick","sex");//返回字段
client.api(params, userId, new TopApiListener(){

    @Override
    public void onComplete(JSONObject json) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onError(ApiError error) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onException(Exception e) {
        // TODO Auto-generated method stub
    }}, true);
```

注意：在 Android3.0 以后对于在 UI 主线程中同步访问网络默认情况下会导致访问失败。所以请慎用同步方式调用 API

附件上传示例代码：

```
TopParameters params=new TopParameters();
params.setMethod("taobao.item.img.upload");//api方法
params.addParam("num_iid", "1500009902569");//业务参数
File file=new File(getFilesDir(), "test.jpg");
FileItem image=new FileItem(file);
params.addAttachment("image", image);//设置附件
client.api(params, userId, new TopApiListener(){

    @Override
    public void onComplete(JSONObject json) {
        // TODO Auto-generated method stub
    }

    @Override
```

```

    public void onError(ApiError error) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onException(Exception e) {
        // TODO Auto-generated method stub
    }, true);

```

2. 调用 TQL

```

/**
 * 调用TQL服务
 *
 * @see TopApiListener
 * @param ql
 * @param userId
 *      需要使用哪个用户授权的access token来调用api，当API不需要
session key时此参数可以为null
 * @param listener
 *      tql调用回调处理监听器，不能为null
 * @param async
 *      true:异步调用；false:同步调用。Android 3.0以后会限制在UI主线
程中同步访问网络，使用同步方式需谨慎
 * @throws IllegalArgumentException
 *      当参数<code>ql</code>或<code>listener</code>为null时
 */
public void tql(final String ql, final Long userId,
                final TopTqlListener listener, final boolean async)

```

示例代码：

```

String ql="{select num,price,type,stuff_status,title
from item where num_iid=15435709307}";
ql+="{select num,price,type,stuff_status,title from item
where num_iid=17546088521}";
client.tql(ql, userId, new TopTqlListener() {
    @Override
    public void onComplete(String result) {
    }

    @Override
    public void onException(Exception e) {
    }}, true);

```

Refresh Token

```
/**
 * 刷新access token的操作。
 *
 * @see #authorize(Activity)
 * @see #addAccessToken(AccessToken)
 * @see AuthorizeListener
 * @param userId
 *         不能为null, 通过userId确定需要刷新哪个access token。如果对应的access
 *         token不存在会抛出IllegalArgumentException
 * @param listener
 *         不能为null, 处理刷新结果的回调函数。
 * @param async
 *         是否是异步执行刷新操作
 * @throws IllegalArgumentException
 */
public void refreshToken(Long userId, final AuthorizeListener listener,
                        boolean async)
```

关于何时进行刷新? 使用 TOP Android SDK 进行开发的应用安全等级为 2 级, R2 级别的授权时长为 3 天, 并且可以通过刷新的方式延长 R2 的授权时长。具体请参考安全等级说明文档:

<http://open.taobao.com/doc/detail.htm?id=1002>

每次授权或者 Refresh Token 完成时通过 AuthorizeListener 都可以获得 AccessToken 对象, 通过此对象的 `getStartDate()` 方法可以获得这个 token 的失效计时的开始时间点。这个时间点是取自淘宝系统时间因为网络传输的问题会有一些时间差。

通过下面的方式可以得到某个级别授权的过期时间点。

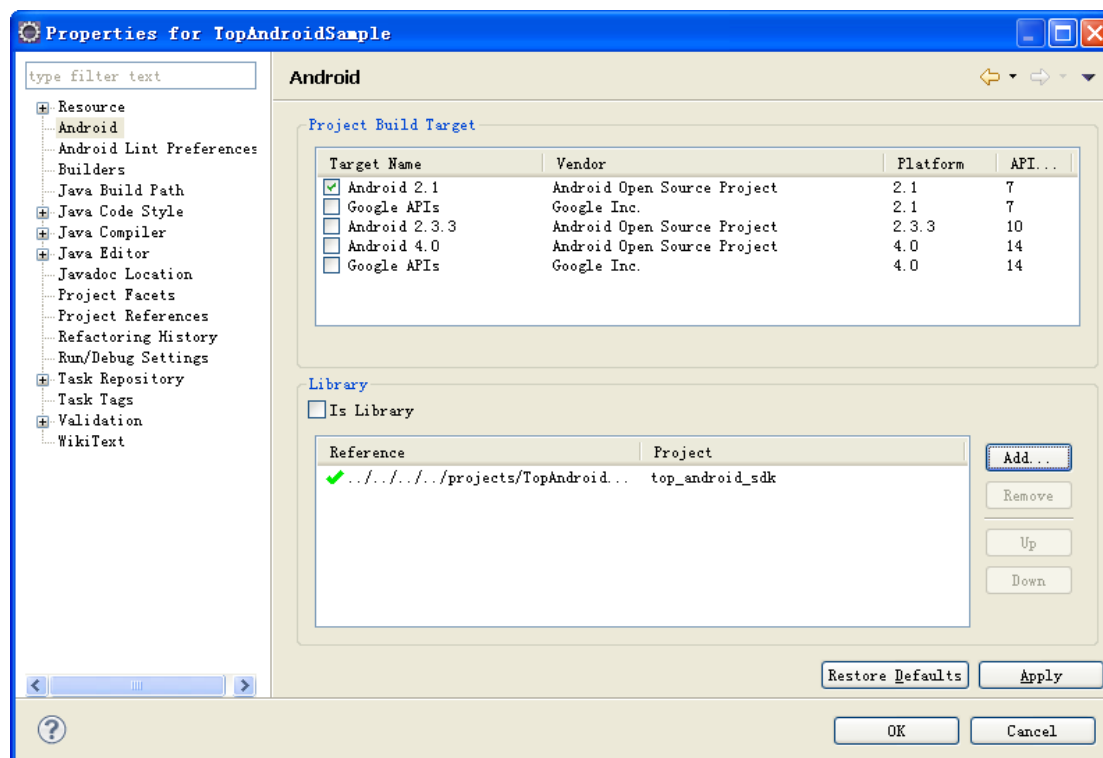
```
String r2_expires = accessToken.getAdditionalInformation().get(
    AccessToken.KEY_R2_EXPIRES_IN);
Date start = accessToken.getStartDate();
Date end = new Date(start.getTime() + Long.parseLong(r2_expires) * 1000L);
```

建议大家在过期时间到来之前提前进行刷新操作。

强烈建议采用异步方式刷新 Token。

Sample 使用说明

下载包解压后 TopAndroidSample 目录下是使用样例。将此工程代码导入 Eclipse 之后请添加对 TOP Android SDK 的依赖。



请修改 **SampleApplication** 类中注册 App Key 的相关代码，请填入你自己的 APP 相关信息。

MainActivity 获得 TopAndroidClient 实例的代码中请填入你的 App Key。

使用 demo 前记得注册 APK 签名哦!!!

在测试 demo 上的功能前请先做一次授权。Android 3.0 以上测试同步调用会失败，失败原因前面已经描述过。



Android 平台支持

CPU 架构	SDK 依赖 Android 系统版本
--------	---------------------

ARM, ARM-NEON	Android 2.1 (API Level 7) and higher
x86	Android 2.3 (API Level 9) and higher
MIPS	Android 2.3 (API Level 9) and higher

技术支持

有问题可以到开放平台支持中心提问：<http://open.taobao.com/support/index.htm>
或加入旺旺群：590851148

Change Log

2012-11-7 Version: android1.0.1

修复 TQL 串行化批量操作模式下返回结果解析错误，同时 TQL 回调 Listener 类型重新定义。
修复未授权情况下调用不需要授权的 TQL 时出现的空指针。