
人教数字教材 SDK 集成说明

修订说明

版本	修订日期	修订人	修订说明
1.0	2018-06-05		新建
1.1	2018-06-12		补充品类目录相关接口

1. 概述

本文档描述了人教数字教材 SDK 对接的一般流程、规则及相关接口的调用方式方法。

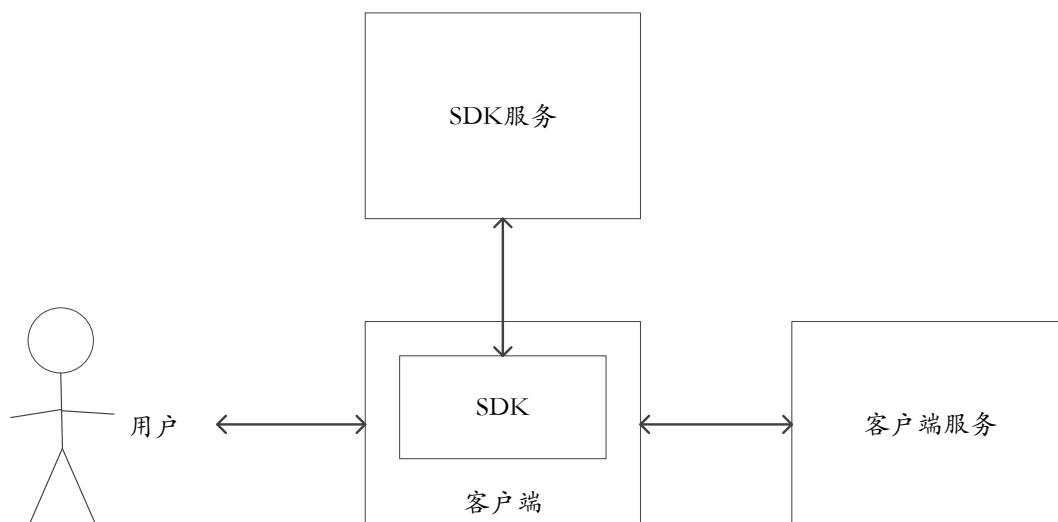
2. 接入流程

- (1) SDK 开发者向人教数字公司提出授权许可申请，审核通过后，将获取 ClientId。ClientId 是使用 SDK 及其相关服务的唯一凭证，请妥善保管。
- (2) 对接方向人教放提交订单，并提供订单编号，有人教后台管理员对订单进行授权，对接方通过订单标号，向人教数字公司通过接口同步授权用户（接口 6.1.3）
- (3) SDK 开发者获取相关软件包及开发说明文档。如采用私有服务器部署方式，SDK 开发者将协助人教数字公司进行相关环境的部署工作。
- (4) SDK 开发者根据相关文档进行 SDK 接入开发、测试及上线等一系列工作。

3. 业务流程

3.1 SDK 与集成系统的关系

下图显示了用户、SDK 与集成系统等角色间的相互关系。



可以看出，纵向的 SDK、SDK 服务与横向的客户端、客户端服务相互依存但又各自独立，每个角色输出特有功能，共同配合以满足用户多样化的需求。

角色说明如下：

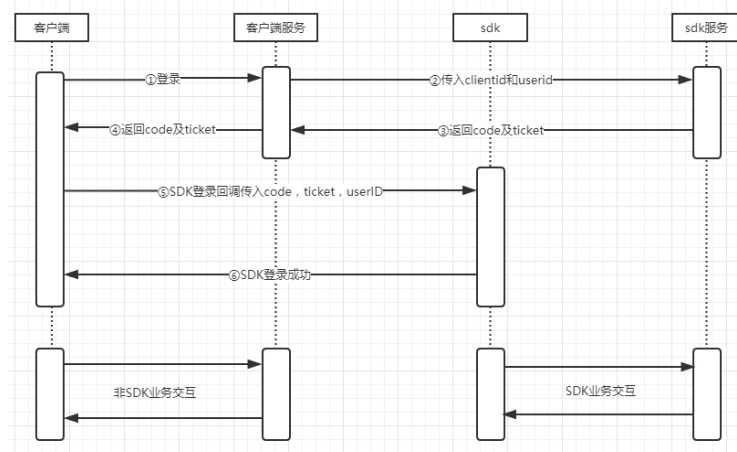
客户端：集成了人教数字教材 SDK 的用户终端软件

客户端服务：为集成了人教数字教材 SDK 的用户终端软件提供接口等服务的后台系统

SDK：人教数字教材 SDK，提供了书架展示、教材阅读、资源调用等相关功能

SDK 服务：为人教数字教材 SDK 提供数据支持、权限认证等服务的后台系统

3.2 SDK 与集成系统的交互



结合上图，对各角色间主要的交互说明如下：

步骤 1：通过 6.1（1）接口获取授权 code（图上返回 ticket 已去掉）

步骤 5：通过 6.1（2）接口获取授权 token，参数 ticket 为当前时间戳

取到 token，即为 SDK 鉴权成功

4. 相关规则

4.1 使用公钥生成 code 的规则说明

(1) code 是通过 RSA 算法使用公钥对“clientId、userId 及有效时间戳”进行加密后再进行 base64 编码生成的字符串；

(2) “clientId、userId 及有效时间戳”之间采用“|”拼接，如：

XXXXX|YYYYY|ZZZZZ

4.2 客户端服务返回某用户教材授权列表的数据规则说明

(1) 请按照下述 json 结构返回：

{

```
"state": "0",

"msg": "success",

"result": {

    "textbook_list": [

        "1211001102161",

        "1211001202171"

    ]

}

}
```

5. 客户端 SDK 接口说明

5.1 PC 端集成

(1) 运行环境

开发语言	.NET 4 WPF
支持 OS	WIN7, WIN8, WIN10
进程类型	目前仅支持 X86 进程, X64 暂不支持

(2) 输出模块

名称	类名	类型	说明
SDK Context	PepSdkContext	普通类	提供对外接口
书城	BooksCenterControl	用户控件	
书架	MyTextBooksControl	用户控件	
期刊中心	JournalsListControl1	用户控件	CommonParamter.Instance.IsJournalEnabled && CommonParamter.Instance.IsTeacher为真是有效
我的期刊	MyJournalsControl1	用户控件	CommonParamter.Instance.IsJournalEnabled && CommonParamter.Instance.IsTeacher为真是有效

(3) 整体样式

需要在程序集中添加 JXP.Resources 引用，并在程序样式中加入以下代码：

```

<Application.Resources>

    <ResourceDictionary>

        <ResourceDictionary.MergedDictionaries>

            <ResourceDictionary

                Source="pack://application:,,,/JXP.Resources;component/Styles/Index.xaml"

            />

        </ResourceDictionary.MergedDictionaries>

    </ResourceDictionary>

</Application.Resources>

```

(4)

① 初始化

◇ 接口名称

```
public bool Init(string serverUrl, string clientId, string productId = "", bool
useSoftRender = false)
```

◇ 参数

类型	名称	说明
System.String	<i>serverUrl</i>	后台服务地址
System.String	<i>clientId</i>	后台分配给客户端的 Id

System.String	productId	产品 Id
System.bool	useSoftRender	是否使用 CPU 方式进行渲染.默认为 false

◇返回值

类型	说明
System.Boolean	true: 成功 false:初始化过程中发生错误 具体错误信息, 请参考 sdk 日志

② 登录

◇接口名称

```
public int Login(string code, string ticket, string userId)
```

◇参数

类型	名称	说明
System.String	<i>code</i>	认证 code
System.String	<i>ticket</i>	请求 ticket
System.String	<i>userId</i>	用户 Id

◇返回值

类型	说明
----	----

System.Int32	0:登录成功 1:获取 Token 信息失败 2:获取教材列表失败 99:程序异常 具体错误信息, 请参考 sdk 日志
--------------	--

③ 打开教材

◇接口名称

```
public int OpenBook(
    string bookId,
    int pageNum,
    Window owner = null
)
```

◇参数

类型	名称	说明
System.String	<i>bookId</i>	书籍 Id
System.Int32	pageNum	书籍页码
System.Windows.Window	owner	阅读窗口的 Owner.为空时, 以程序主窗体为 Owner

◇返回值

0:打开成功

1:用户未登录

2:书籍没有下载或者下载未完成

3:书籍权限过期

99:程序异常

<remarks>具体错误信息, 请参考 sdk 日志</remarks>

④ 登出

◇接口名称

```
public void Logout()
```

◇参数

无

◇返回值

无

⑤ 销毁

◇接口名称

```
public void Dispose()
```

◇参数

无

◇返回值

无

⑥ 查询教材列表

◇接口名称

```
public string GetTextbookList(  
  
    string rkxd,  
  
    string zxxkc,  
  
    string nj,  
  
    string name,  
  
    int pageno,  
  
    int pagesize,  
  
    int detail  
  
)
```

◇参数

参数名	含义	类型	必需	备注
rkxd	学段	String	否	
zxxkc	学科	String	否	
nj	年级	String	否	

name	教材名称	String	否	支持模糊搜索
pageno	分页页码	int	否	默认 1
pagesize	分页大小	int	否	默认 10
detail	是否返回详 情	int	否	0/1; 默认 0 不返回

◇返回值

参数名				含义	类型	备注
errno				响应状 态码	String	0: 成功 其他: 失败
errmsg				响应状 态信息	String	
data				响应数 据	Object	
	total			查询结 果总数	int	
	page			分页信 息	Object	

		total		总页数	int	
		current		当前第 几页	int	
	list			本页列 表	Object	
		id		教材 ID	String	
		name		教材名 称	String	
		cover_img_url		封面地 址	String	
		detail		教材详 情	Object	detail 为 true 返 回
			rkxd	学段编 号	String	
			rkxd_name	学段名 称	String	

			nj	年级编号	String	
			nj_name	年级名称	String	
			zxxkc	学科编号	String	
			zxxkc_name	学科名称	String	
			fascicule	册别编号	String	
			fascicule_name	册别名称	String	
			edition	编著版本	String	
			pub_version	数字教材发布版本	String	

			create_time	数字教材制作时间	Date	
			pub_time	数字教材发布时间	Date	
			size	教材大小	int	单位 Byte
			publisher	出版社	String	
			publish_time	出版时间	String	
			folio	开本	String	
			series	丛书	String	
			isbn	ISBN	String	
			res_count	资源数量	int	

◇返回示例

{

“errno” : “0” ,

```
“errmsg” : “success” ,

“data” : {

    “total” : “15” ,

    “page” : {

        “total” : “3” ,

        “current” : “1”

    },

    “list” : [

        {

            “id” : “1211001101161” ,

            “name” : “语文 一年级上册” ,

            “cover_img_url” :

                “http://127.0.0.1/1001.jpg”

        }

    ]

}

}
```

⑦ 查询教材详情

◇接口名称

```
public string GetTextbookDetail(
    string id,
    int chapter
)
```

◇参数

参数名	含义	类型	必需	备注
id	教材 ID	String	是	
chapter	是否返回章节	int	否	0/1; 默认 0 不返回

◇返回值

参数名	含义	类型	备注
errno	响应状态码	String	0: 成功 其他: 失败
errmsg	响应状态信息	String	
data	响应数据	Object	

	id		教材 ID	String	
	name		教材名称	String	
	cover_img_url		封面地址	String	
	rkxd		学段编号	String	
	rkxd_name		学段名称	String	
	nj		年级编号	String	
	nj_name		年级名称	String	
	zxxkc		学科编号	String	
	zxxkc_name		学科名称	String	
	fascicule		册别编号	String	
	fascicule_name		册别名称	String	
	edition		编著版本	String	
	pub_version		数字教材 发布版本	String	
	create_time		数字教材 制作时间	Date	
	pub_time		数字教材	Date	

			发布时间		
	size		教材大小	int	单位 Byte
	publisher		出版社	String	
	publish_time		出版时间	String	
	folio		开本	String	
	series		丛书	String	
	isbn		ISBN	String	
	res_count		资源数量	int	
	chapter		章节列表	List	
		id	章节 ID	String	
		name	章节名称	String	
		index_s	开始页码	int	
		index_e	结束页码	int	
		is_chapter	是否为章节	int	
		level	级别	int	
		pid	父节点 ID	String	
		sort_num	排序编码	int	

◇返回示例

```
{  
  
  "errno" : "0" ,  
  
  "errmsg" : "success" ,  
  
  "data" : {  
  
    "id" : "1211001101161" ,  
  
    "name" : "语文 一年级上册" ,  
  
    "cover_img_url" : "http://127.0.0.1/1001.jpg" ,  
  
    "rkxd" : "2" ,  
  
    "rkxd_name" : "小学" ,  
  
    "nj" : "21" ,  
  
    "nj_name" : "一年级" ,  
  
    "zxxkc" : "11" ,  
  
    "zxxkc_name" : "语文" ,  
  
    "fascicule" : "01" ,  
  
    "fascicule_name" : "上册" ,  
  
    "pub_version" : "16" ,  
  
    "create_time" : "2018-06-01 10:12:00" ,  
  
    "pub_time" : "2018-06-01 10:12:00" ,
```

```

“size” : “12658410” ,

“publisher” : “人民教育出版社” ,

“publish_time” : “2016 年 11 月第 1 版” ,

“folio” : “787 毫米 × 1092 毫米 1/16” ,

“series” : “第十一套义务教育教科书” ,

“isbn” : “978-7-107-31520-6” ,

“res_count” : “456”

    }

}
    
```

⑧ 查询我的教材列表

◇接口名称

```

public string GetMyTextbookList(

    int detail

)
    
```

◇参数

参数名	含义	类型	必需	备注
detail	是否返回详情	int	否	0/1; 默认 0 不返回

◇返回值

参数名				含义	类型	备注
errno				响应状态码	String	0: 成功 其他: 失败
errmsg				响应状态信息	String	
data				响应数据	Object	
	list			本页列表	Object	
		id		教材ID	String	
		name		教材名称	String	
		cover_img_url		封面地址	String	
		download_state		下载状态	int	0: 未下载

						1: 已下载
		detail		教材详情	Object	detail 为 true 返回
			rkxd	学段编号	String	
			rkxd_name	学段名称	String	
			nj	年级编号	String	
			nj_name	年级名称	String	
			zxxkc	学科编号	String	
			zxxkc_name	学科名称	String	
			fascicule	册别编号	String	

			fascicule_name	册别名称	String	
			edition	编著版本	String	
			pub_version	数字教材发布版本	String	
			create_time	数字教材制作时间	Date	
			pub_time	数字教材发布时间	Date	
			size	教材大小	int	单位 Byte
			publisher	出版社	String	
			publish_time	出版时间	String	
			folio	开本	String	

			series	丛书	String	
			isbn	ISBN	String	
			res_count	资源数 量	int	

◇返回示例

```
{  
  
  "errno" : "0" ,  
  
  "errmsg" : "success" ,  
  
  "data" : {  
  
    "list" : [  
  
      {  
  
        "id" : "1211001101161" ,  
  
        "name" : "语文 一年级上册" ,  
  
        "download_state" : "0" ,  
  
        "cover_img_url" : "http://127.0.0.1/100  
1.jpg"  
      }  
  
    ]  
  
  }  
  
}
```

```
        }  
    ]  
}
```

}使用示例

① 初始化

```
protected override void OnStartup(StartupEventArgs e)  
  
{  
  
    // 初始化SDK  
  
    PepSdkContext.Instance.Init(Constants.SERVER_URL,  
Constants.CLIENT_ID);  
  
    base.OnStartup(e);  
  
}
```

② 登录

```
private void LoginCtrl_OnLoginClick(object sender, DemoLoginEventArgs  
e)  
  
{
```

```
PepSdkContext.Instance.Login(e.LoginData.Code, e.LoginData.Ticket,  
e.LoginData.UserId);  
  
}
```

③ 登出

```
private void LoginCtrl_OnLogoutClick(object sender,  
DemoLoginEventArgs e)  
  
{  
  
    PepSdkContext.Instance.Logout();  
  
}
```

④ 销毁

```
protected override void OnExit(ExitEventArgs e)  
  
{  
  
    // 销毁SDK  
  
    PepSdkContext.Instance.Dispose();  
  
    base.OnExit(e);  
  
}
```

⑤ 我的教材中点击添加教材处理

定制我的教材中点击添加教材事件，如：

```
MyTextBooksControl.OnAddTextbook += BtnAddTextBook_Click;
```

定义处理事件，在事件中完成跳转到教材中心的逻辑，如：

```
/// <summary>
    /// 我的教材，按钮“添加教材”点击事件
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void BtnAddTextBook_Click(object sender,
System.EventArgs e)
    {
        //跳转到教材中心
        //……在这里写你的代码
    }
}
```

⑥ 我的期刊中点击添加期刊处理

定制我的期刊中点击添加期刊事件，如：

```
MyJournalsControl.OnAddTextbook += BtnAddTextBook_Click;
```

定义处理事件，在事件中完成跳转到教材中心的逻辑，如：

```
/// <summary>
    /// 我的期刊，按钮“添加期刊”点击事件
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void BtnAddTextBook_Click(object sender,
System.EventArgs e)
    {
        //跳转到期刊中心
        //……在这里写你的代码
    }
}
```

5.2 Android 端集成

(1)运行环境

(2)注意

开发语言	Java
支持 Android 版本	Android 5.0 及以上

1,注意

意:升级 App 的时候,请提示用户先将安卓端添加的用户资源同步到云端,避免因升级造成数据丢失

2,app 的 build.gradle 文件中, defaultConfig 需要配置 applicationId

(3)集成

1, 将 pepsdk.zip 解压到电脑的指定目录 ,例如直接解压到 d 盘

2,在项目的 build.gradle 中, allprojects 中的 repositories 下添加:

```
maven { url 'https://jitpack.io' }
```

3, 在 app 的 build.gradle 文件的 android{ }中添加

```
repositories {
    maven { url "D:\\pepsdk" }
}
```

注:这里的 url 修改为 pepsdk.zip 的解压路径

4, 在 app 的 build.gradle 文件的 dependencies 中添加

```
implementation 'com.pep.sdk:pepsdk:1.6.0'
```

(4)调用

1.初始化

应用启动后进行初始化操作，在 Application 的 onCreate 中初始化

```
/**
 * SDK 初始化
 * @param this 上下文
 * @param baseUrl 域名
 * @param APP_KEY 后台分配给客户端的 id(通常由商务提供)
 */
PepApp.init(this, "baseUrl", "APP_KEY")
```

2.登录和登出

◇登录:

App 在登录服务器后，调用 SDK 的 login

```
/**
 * SDK 登录操作
 *
 * @param code code
 * @param userId 用户 id
 * @param isAutoLogin 是否下次自动登录
 * @param onLoginCallbackListener 回调
 */
PepApp.netAccessTokenLogin(String code, String userId,
final boolean isAutoLogin, final OnLoginCallbackListener
onLoginCallbackListener)
```

◇登出:

退出登录时调用

```
PepApp.logout()
```

3.书架

书架组件为了方便调用和管理使用 Fragment

书架的 Framgent 为 **BookFragment**

正常使用 fragment 即可

例如在布局中直接使用

```
<fragment
android:name="com.pep.core.foxitpep.fragment.BookFragment"
android:layout_width="match_parent"
android:layout_height="match_parent"
/>
```

4. 小学 ui 的打开和关闭

```
PepApp.setPrimaryUi(true);
```

不调用方法,默认是 **true**

参数为 **true**: 自动根据学段判断是否使用小学 **ui**

参数为 **false**: 关闭小学 **ui**

5. 监听 token 过期

```
PepApp.setOnTokenExpiredListener(new PepApp.OnTokenExpiredListener() {
    @Override
    public void onTokenExpired() {
        //token 过期
    }

    @Override
    public void onNeedLogin() {
//一般用不到
        //目前提供的 BookFragment, 未登录点击我的教材, 会触发此回调
    }
});
```

(5) 混淆配置

```
-keep class com.pep.core.** { *; }
```

```
-keep class com.ckl.** { *; }
```

```
-keep class com.creative.** { *; }
-keep class com.gumi.** { *; }
-keep class com.foxit.sdk.** { *; }
-keep class com.foxit.uiextensions.** { *; }
-dontwarn com.foxit.sdk.**
-dontwarn com.foxit.uiextensions.**
-keepattributes Exceptions, Signature, InnerClasses
-keepattributes SourceFile,LineNumberTable
```

5.3 iOS 接入

1 在 pod 中加入

```
pod 'YYModel'
pod 'Masonry'
pod 'SSZipArchive'
pod 'SAMKeychain'
pod 'PEPNetworking'
pod 'PEPBigData'
pod 'SDWebImage'
pod 'MBProgressHUD'
pod 'MJRefresh'
pod 'FMDB'
pod 'ZSSRichTextEditor', :git =>
'https://github.com/nnhubbard/ZSSRichTextEditor.git'
```

2 加入 PDFReaderSDKTextBook, CKCEvaluationSDK , framework

3 在跳转页面


```
#import <PDFReaderSDKTextBook/PDFReaderSDKTextBook.h>

#import <AFNetworking.h>

#import <PDFReaderSDKTextBook/PEPHttpUtil.h>

#import <PDFReaderSDKTextBook/PEPURLPath.h>

NSMutableDictionary *dic = [NSMutableDictionary dictionary];

[dic setObject:@"" forKey:@"ticket"];

[dic setObject:@"" forKey:@"code"];

[dic setObject:@"" forKey:@"user_id"];

[dic setObject:@"" forKey:@"client_id"];

[dic setObject:@"2" forKey:@"device_type"];

PEPReaderTextBookManager *manager =
[PEPReaderTextBookManager
PEPShareManagerWithAPPID:SDKClientAppID]; // [manager
setBaseURL:@"https://testczpt.mypep.cn/jxw-sdk-web"];

[manager setBaseURL:nil];

// [manager setBaseURL:@"https://testczpt.mypep.cn/jxw-sdk-
web"];

[manager PEPLoginWithAuthCode:dic andCallBack:^(id response) {

    BookListViewController *bookListVC = [[BookListViewController
alloc] init];

    bookListVC.hidesBottomBarWhenPushed = YES;

    self.tabBarController.tabBar.hidden = YES;
```

```
[self.navigationController pushViewController:bookListVC  
animated:YES];  
    }];
```

6. SDK 服务端相关接口说明

6.1 服务端鉴权

(1) 获取动态授权 code——authorizeCode

① 接口名称

获取动态授权 code——authorizeCode

② 请求类型

POST

③ 请求 Url

saas/auth/authorizeCode.json

④ 请求参数

Name	Identifier	Type	Remark
接入方用 户 ID	user_id	string	
接入方 client_id	client_id	string	

⑤ 响应参数

Name	Identifier	Type	Remark
状态码	statusCode	string	
	dataMap	object	
---到期具体时间	expires_time	number	
---接入方 client_id	client_id	string	
---接入方用户 id	user_id	string	
---有效时间长度	expires_in	number	
---接入方认证 code	authorize_code	string	
---访问时间，为规避用户使用过程中掉线，将时间最终设定为当日凌晨两点	s_create_time	string	
---最后访问时间	last_request_time	number	
状态文本	msg	string	

(2) 授权用户导入

① 接口名称

授权用户导入

② 请求类型

POST

③ 请求 Url

saas/auth/userImport.json

④ 请求参数

Name	Identifier	Type	Remark
data 长度	dataCount	number	
接入方 client_id	client_id	string	
接入方 access_token	access_token	string	给出固定字符串 “400X587ASG”，desc 加密
订单编号	orderNumber	string	
用户信息数组	data	array<object>	desc 加密
---用户实体	user	object	

-----角色 1: 学生; 2: 老师	role	number	
-----用户名	name	string	
-----用户 id	user_id	string	
-----学科	zxxkc	string	
-----学校	school	string	
-----学段	rkxd	string	

⑤ 响应参数

Name	Identifier	Type	Remark
状态 (500110: 成功)	state	number	

文本（根据状态值做出相应文字提示）	msg	string	
-------------------	-----	--------	--

7. 附录

注意：SDK 服务分为测试服务和正式服务，分别对应的域名为：

testczpt.mypep.cn 和 czpt.mypep.cn

建议对接时使用测试域名，由于 sdk 客户端中配置的是正式域名
