
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。

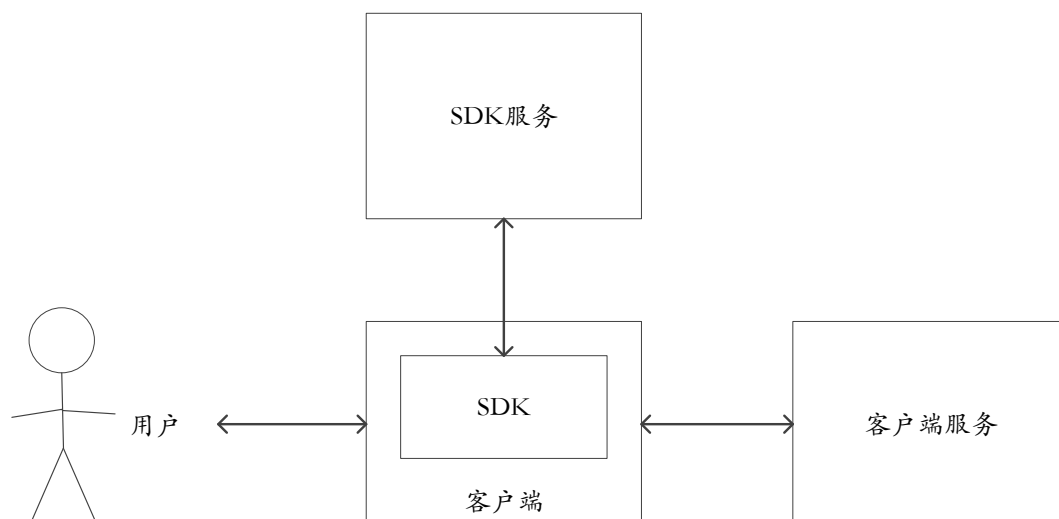
2. 接入流程

- (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：

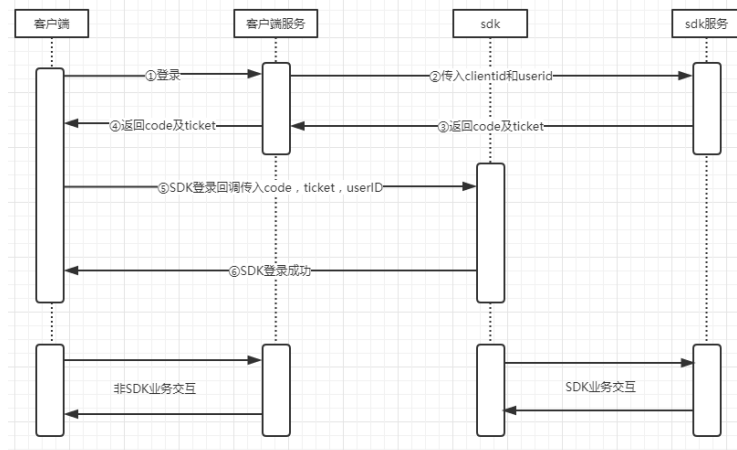
客户端：集成了人教数字教材 SDK 的用户终端软件

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：

XXXXXX|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

5.1PC 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
SD K Context	PepSdkContext	普 通类	提供对外接口
书 城	BooksCenterContr ol	用 户控 件	
书 架	MyTextBooksCon trol	用 户控 件	
期 刊中心	JournalsListContro l	用 户控 件	CommonParamter. Instance. IsJournalE nable && CommonParamter. Instance. IsTeacher 为真 是有效
我 的期刊	MyJournalsContro l	用 户控 件	CommonParamter. Instance. IsJournalE nable && 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"
```

```
        }  
    }  
}  
  
}
```

⑨ 添加教材下载任务（该接口需要后台授权）

◇接口名称

```
public bool AddNewDownloadTask(string tbId, int downType, ChaptersModellist  
chaptersTree, bool isDownloadResource =true)
```

◇参数

参数名	含义	类型	必需	备注
tbId	书籍 Id	string	是	
downType	下载类 型	int	是	0 整本下载，1 按章节下 载
chaptersTree	要下载 的章节	ChaptersM odellist	是	按章节下载时，该参数不 能为 null
isDownloadResource	是否下 载资源	bool	否	默认下载资源，如果只下 载教材，可设置为 false

				注意：如果没有资源下载 权限该参数无效
--	--	--	--	------------------------

◇返回值

参数名	含义	类型	备注
	是否成功加入 下载队列	bool	true：成功 false：失败

◇返回示例

略

⑩ 启动、暂停下载任务（该接口需要后台授权）

◇接口名称

```
public bool StartPauseDownloadTB(string strId)
```

◇参数

参数名	含义	类型	必需	备注
strId	书籍 Id	string	是	

◇返回值

参数名	含义	类型	备注
-----	----	----	----

	是否成功加入下载队列	bool	true: 成功 false: 失败
--	------------	------	-----------------------

◇返回示例

略

⑪ 继续下载剩余章节（分章节下载后续操作选项 该接口需要后台授权）

◇接口名称

```
public bool ContinueDownloadTextBook(string tbId)
```

◇参数

参数名	含义	类型	必需	备注
tbId	书籍 Id	string	是	

◇返回值

参数名	含义	类型	备注
	是否成功加入下载队列	bool	true: 成功 false: 失败

◇返回示例

略

⑫ 删除教材（包含文件 该接口需要后台授权）

◇接口名称

```
public bool DeleteTextBook(string tbId)
```

◇ 参数

参数名	含义	类型	必需	备注
tbId	书籍 Id	string	是	

◇ 返回值

参数名	含义	类型	备注
	是否成功加入 下载队列	bool	true: 成功 false: 失败

◇ 返回示例

略

⑬ 取消下载任务（该接口需要后台授权）

◇ 接口名称

```
public bool CancelDownloadTextBookByTBId(string tbId)
```

◇ 参数

参数名	含义	类型	必需	备注
tbId	书籍 Id	string	是	

◇ 返回值

参数名	含义	类型	备注
	是否成功加入下载队列	bool	true: 成功 false: 失败

◇返回示例

略

⑭ 更新教材（该接口需要后台授权）

◇接口名称

```
public bool UpdateTextBook(string tbId)
```

◇参数

参数名	含义	类型	必需	备注
tbId	书籍 Id	string	是	

◇返回值

参数名	含义	类型	备注
	是否成功加入下载队列	bool	true: 成功 false: 失败

◇返回示例

略

⑮ 激活教材（适用包含激活业务）

◇接口名称

```
public bool ActiveTextbook(string bookId, string code, out string msg)
```

◇参数

参数名	含义	类型	必需	备注
bookId	书籍 Id	string	是	
code	激活码	string	是	
msg	返回提示信息	string	是	

◇返回值

参数名	含义	类型	备注
	是否成功加入下载队列	bool	true: 成功 false: 失败

◇返回示例

略

注意：对于包含激活逻辑的第三方客户，如果使用默认的激活处理，则不用特殊设置；

如果使用教材中心和我的教材控件，希望程序自动为用户输入激活码这种，需要设置 `SdkConstants.IsAutoActiveTextbook=true`，并订阅教材中心控件的 `UserActiveHandler` 事件，自己处理激活前置逻辑，如果满足激活条件，请调用该接口进行激活；

对于未登录教材中心显示激活，用户点击时的逻辑处理，请订制：

`BookCenterControl.UserToLoginHandler` 事件，在事件中添加相应跳转到登录页或相关提示逻辑；只使用单册教材功能相关的，直接调用激活方法自行处理。

⑩ 打开我的教材（翻页功能不受限制 该接口需要后台授权）

◇接口名称

```
public int OpenMyTextBook(string id,Window owner=null)
```

◇参数

参数名	含义	类型	必需	备注
id	书籍 Id	string	是	
owner	所属窗体	Window	否	一般指调用该接口的窗体，建议设置

◇返回值

参数名	含义	类型	备注
	打开教材是否成功	int	0 打开成功，-1 教材不存在，需要先下载，1 其它异常,2 该接口未授权

◇返回示例

略

⑪ 获取登录用户本地已下载的教材列表（该接口需要后台授权）

◇接口名称

```
public List<TransferTextbookModel> GetLocalMyTextbookList()
```

◇参数

参数名	含义	类型	必需	备注
无				

◇返回值

参数名	含义	类型	备注
	获取本地已 下载教材列 表	List<TransferTextbookModel>	

◇返回示例

略

⑱ 获取常量列表（该接口需要后台授权）

◇接口名称

```
public MetaList GetConstDataByType(int dataType)
```

◇参数

参数名	含义	类型	必需	备注
dataType	常量类 型	int	是	0 学段，1 学科，2 出版社

◇返回值

参数名	含义	类型	备注
略			

◇返回示例

略

⑲ 清除所有下载任务

◇接口名称

```
public void DisposeDownloadTask()
```

◇参数

参数名	含义	类型	必需	备注
无				

◇返回值

参数名	含义	类型	备注
无			

◇返回示例

略

⑳ 教材导出获取教材大小接口（需要后台授权）

◇接口名称

```
public long GetExportBookSize(TransferTextbookModel book)
```

◇参数

参数名	含义	类型	必需	备注
book	要导出的教材实体	TransferTextbookModel	是	

◇返回值

参数名	含义	类型	备注
教材+资源占用空间			

21 教材导出接口（需要后台授权）

◇接口名称

```
public bool ExportBook(TransferTextbookModel book, string targetDir,
    SetProcess setProcessCallback=null,
    CancelExportBook cancelExportBookCallBack=null)
```

◇参数

参数名	含义	类型	必需	备注
-----	----	----	----	----

book	要导出的教材实体	TransferTextbookModel	是	
targetDir	导出保存文件夹路径	string	是	
setProcessCallback	导出进度回调函数	SetProcess	否	
cancelExportBookCallBack	导出取消回调	CancelExportBook	否	导出时会频繁调用此回调，以检查当前的导出是否取消，取消返回 true

◇返回值

参数名	含义	类型	备注
bool			

22 教材导入接口（需要后台授权）

◇接口名称

```
public void InportBook(string[] strFilePaths, DoWorkEventArgs doWorkEventArgs = null,
```

```
ShowQuestionMessgae showQuestionMessage = null,
EventHandler<StringEventArgs> showCurrentImportFilePath = null,
JXP.PepDtp.PackPorts.AddMyTextBook addMyTextBookCallback = null,
SetProcessValue setImportProgressValueCallBack = null)
```

◇ 参数

参数名	含义	类型	必需	备注
strFilePaths	要导出的教材实体	string[]	是	要导入的教材包路径集合
doWorkEventArgs	导入取消控制参数	DoWorkEventArgs	否	导入取消时，设置此参数 doWorkEvent.Cancel = true;
showQuestionMessage	导入过程询问是否覆盖回调，如果本地已存在该教材	ShowQuestionMessgae		
showCurrentImportFilePath	导入时当前处理路径回调函数	EventHandler<StringEventArgs>	否	用于导入时显示当前处理信息
addMyTextBookCallback	导入成功回调	JXP.PepDtp.PackPorts.AddMyTextBook	否	导入成功后会触发此回调，便于及时更新

				教材列表，也可以导入完成后统一刷新
setImportProgress ValueCallBack	导入进度回调	SetProcess Value		

◇返回值

参数名	含义	类型	备注
无			

23 单册回调相关

打开教材回调：

◇接口名称

添加回调	
设置教材打开、关闭触发回调	<code>public void SetOwnerVisibilityCallBack(SetOwnerVisibility setOwnerVisibilityCallBack)</code>
教材打开结果回调	<code>public void SetNotifyOpenBookReusltCallBack(NotifyOpenBookReuslt notifyOpenBookReusltCallBack)</code>
移除回调	
移除设置教材打开、关闭触发回调	<code>public void RemoveOwnerVisibilityCallBack(SetOwnerVisibility setOwnerVisibilityCallBack)</code>
移除教材打开结果回调	<code>public void RemoveNotifyOpenBookReusltCallBack(NotifyOpenBookReuslt notifyOpenBookReusltCallBack)</code>

下载教材回调：

◇接口名称

设置删除教材完成时回调	<code>public void SetDelBookFinishCallBack(DelBookFinish delBookFinish)</code>
设置刷新封皮回调函数	<code>public void SetRefreshThumbCallBack(RefreshThumb refreshThumbCallback)</code>
设置添加下载任务成功后回调函数	<code>public void SetAddMyTextBookCallBack(AddMyTextBook addMyTextBookCallBack)</code>
设置添加到我的教材回调函数	<code>public void SetNotifyDownloadCountCallBack(NotifyDownloadCount notifyDownloadCountCallBack)</code>
教材下载失败从我的教材中移除回调	<code>public void SetRemoveTBFromMyTextBookCallBack(RemoveTBFromMyTextBook removeTBFromMyTextBookCallBack)</code>
设置下载过程消息回调	<code>public void SetNotificationMessageCallBack(NotifyDownloadMessage notificationMessageCallBack)</code>
设置下载过程进度回调	<code>public void SetNotificationDownloadProgressCallBack(NotifyDownloadProgress notificationDownloadProgressCallBack)</code>
设置资源下载失败询问弹窗回调函数	<code>public void SetQuestionRetryResourceDownloadFailCallBack(QuestionRetryResourceDownloadFail questionRetryResourceDownloadFailCallBack)</code>
设置教材下载失败询问是否重试	<code>public void SetQuestionRetryDownloadFailCallBack(QuestionRetryDownloadFail questionRetryDownloadFailCallBack)</code>

(5) 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: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:1.6.3'
```

(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(netAccessTokenLogin(String code, String userId,
final boolean isAutoLogin, final OnLoginCallbackListener
onLoginCallbackListener)
```

◇登出:

退出登录时调用

```
PepApp.loginOut()
```

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
```

(6) 单册下载

注:本文档内容需要授权才可使用

1,打开书籍

```
/**  
 * @param context 上下文  
 * @param bookId 教材id  
 */
```

```
PEPBookActivity.invoke(Context context, String bookId);
```

2, 下载教材和资源

```
/**  
 * @param bookId 教材id  
 */
```

```
BookDownloadManager.getInstance().downloadBookSection(String  
bookId);
```

3,只下载教材pdf

```
/**  
 * @param bookId 教材id  
 */
```

```
BookDownloadManager.getInstance().downloadBook(String bookId);
```

4, 获取书籍状态

```
/**  
 * @param context 上下文  
 * @param bookId 教材id  
 */
```

```
Int status =  
BookDownloadManager.getInstance().getBookStatus(Context context,  
String bookId);
```

返回值:

status == 0 未下载

status == 1 下载中

status == 2 下载成功

status == 3 下载中, 下载成功了部分, 需要继续下载

status == 4 下载成功, 但资源并未完全下载

5, 删除下载信息

```
/**  
 * 下载成功的时候调用, 删除临时存放的下载进度  
 * @param bookId 教材id  
 */
```

```
BookDownloadManager.getInstance().removeBookDownload(String  
bookId);
```

6,更新教材

```
/**  
 * @param bookId 教材id  
 */
```

```
BookDownloadManager.getInstance().updateBook(String bookId);
```

7,删除教材

```
/**  
 * @param bookId 教材id  
 */
```

```
BookDownloadManager.getInstance().deleteBooks(String bookId);
```

8,获取筛选信息

```
StaticCascadeConfig.getStaticMetadataBean(new  
StaticCascadeConfig.OnStaticCascadeListener() {  
    @Override  
    public void onData(StaticCascadeModel staticCascadeModel) {
```

```
    }  
});
```

返回值：StaticCascadeModel

注：ContentBean 为通用实体：

contentBean.id id

contentBean.value 名称

(1)可选的出版社列表

```
List<ContentBean> publisherContent =  
staticCascadeModel.static_metadata.publisher.content;
```

(2)可选的学段列表

```
List<ContentBean> PhaseContent =  
staticCascadeModel.static_metadata.phase.content;
```

```
for (int i = 0; i < PhaseContent .size(); i++) {  
    String phaseId = contentPhase.get(i).id;  
  
    if ("2".equals(phaseId )){  
        //小学学段  
    }else if ("3".equals(id)){
```

```
//初中学段
}else if ("4".equals(id)){
//高中学段
}
}
```

(3) 可选学科

注:学科数据需要根据当前所选的学段进行获取

①所有学段可选学科

```
List<ContentBean> content =
staticCascadeModelData.static_metadata.subject.content;
```

②小学学段可选学科

```
List<StaticCascadeModel.CascadeMetadada.CascadeBean.Content
Bean> content =
staticCascadeModelData.cascade_metadata.primary.content;
```

③初中学段可选学科

```
List<StaticCascadeModel.CascadeMetadada.CascadeBean.Content
Bean> content =
staticCascadeModelData.cascade_metadata.junior.content;
```

④高中学段可选学科

```
List<StaticCascadeModel.CascadeMetadada.CascadeBean.Content  
Bean> content =  
staticCascadeModelData.cascade_metadata.senior.content;
```

9, 获取下载进度

```
/**  
* @param bookId 教材id  
*/  
  
BookDownloadManager.BookProgress bookProgress =  
BookDownloadManager.getInstance().getBookProgress(String  
bookId);
```

```
进度:Int progress = bookProgress.progress;
```

```
速度:String speed = bookProgress.speed;
```

```
状态:Int status = bookProgress.status;
```

```

BookDownloadCallable.BOOK_DOWN_LOAD      == status  下载中
BookDownloadCallable.BOOK_DOWN_COMPLETE == status  下载完成
BookDownloadCallable.BOOK_DOWN_ERROR     == status  下载失败

```

10, 添加下载进度监听

```

/**
 * 添加下载监听
 * @param bookId      书籍id
 * @param onDownloadListener 监听
 * @param tag          标记
 */

```

```

BookDownloadManager.getInstance().addOnDownloadListener(String bookId, BookDownloadManager.OnDownloadListener onDownloadListener, String tag);

```

例:

```

BookDownloadManager.getInstance().addOnDownloadListener(bookId, new BookDownloadManager.OnDownloadListener() {
    @Override
    public void onProgressStatus(String bookId, long progress, String speed, int status, Throwable exception) {

```

```
//bookId 教材 id

//progress 进度

//speed 下载速度


//status 状态

BookDownloadCallable.BOOK_DOWN_LOAD == status 下载中

BookDownloadCallable.BOOK_DOWN_COMPLETE == status 下载完成

BookDownloadCallable.BOOK_DOWN_ERROR == status 下载失败

BookUnZipCallable.BOOK_DOWN_UNZIP_LOAD == status 解压中

BookUnZipCallable.BOOK_DOWN_UNZIP_COMPLETE == status 解压成功

}

}, "myBookAdapter");
```

11, 下载异常

Exception	原因
FileDownloadHttpException	在发出请求以后，response-code 不是 200(HTTP_OK)，也不是

Exception	原因
	206(HTTP_PARTIAL)的情况下会抛出该异常; 在这个异常对象会带上 response-code、response-header、request-header。
FileDownloadGiveUpRetryException	在请求返回的 response-header 中没有带有文件大小(content-length), 并且不是流媒体(transfer-encoding)的情况下会抛出该异常; 出现这个异常, 将会忽略所有重试的机会
	(BaseDownloadTask#setAutoRetryTimes)
	. 你可以通过
	在 filedownloader.properties 中添加 http.lenient=true 来忽略这个异常, 并且在该情况下, 直接作为流媒体进行下载。
FileDownloadOutOfSpaceException	当将要下载的文件大小大于剩余磁盘大小时, 会抛出这个异常。
java.net.SocketException	网络断开时可能会出现

Exception	原因
和 <code>javax.net.ssl.SSLException</code> <code>ion</code>	
<code>FileDownloadNetworkPolicyException</code>	设置了 <code>BaseDownloadTask#setWifiRequired(true)</code> ，在下载过程中，一旦发现网络情况转为非 Wifi 环境，便会抛回这个异常
<code>PathConflictException</code>	当有一个正在下载的任务，它的存储路径与当前任务的存储路径完全一致，为了避免多个任务对同一个文件进行写入，当前任务便会抛回这个异常

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	orderId	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	

6.2单册教材调用接口

6.2.1. 取服务器授权 token

① URL

/saas/auth/serviceAuthCode.json

② Method

GET

③ Description

请求服务器需与使用该 code 的服务器出口 ip 保持一致

Request Param List

Name	Identifier	Type	Remark
client_id	client_id	string	对接项目渠道 id
appId	appId	string	附加参数，格式为：w+32 为随机数+13 位时间戳，只一次请求有效（java 请参考示例代码中的 main 方法）
sign	sign	string	附加参数，格式为：根据 appId 使用指定算法生成，只一次请求有效（java 请参考示例代码中的 main 方法）

Response Param List

Name	Identifier	Type	Remark
请求 状态	statusCode	number	
状态 说明	msg	string	

Name	Identifier	Type	Remark
请求 返回解密 key	key	string	用于使用此 token 请求的返回 值解密使用
服务 器授权 token	token	string	仅一次有效

6.2.2. 资源列表

URL

/saas/out/auth/getResource.json

Method

POST

Description

Request Param List

参数名称	参数类型	参数说明
id	string	资源 id
pageNo	string	示例：单页 2；双页 2, 3（双页用逗号 隔开）
access_token	string	访问凭证

参数名称	参数类型	参数说明
appld	string	附加参数，格式为：w+32 为随机数 +13 位时间戳，只一次请求有效（java 请参考示例代码中的 main 方法）
sign	string	附加参数，格式为：根据 app_id 使用 指定算法生成，只一次请求有效（java 请参考示例代码中的 main 方法）

Response Param List

Nam e	Identifier	Type	Remark
5001 00 代表成功， 500192 代表参数不合法	statusCode	string	@mock=500110
提示信息	msg	string	@mock=request success

Nam e	Identifier	Type	Remark
	dataList 9	array<obje ct>	
---资 源格式	file_for mat	string	@mock=\$order('.jpg','.jpg','.mp3','.mp3','.mp3')
---资 源名称	title	string	@mock=\$order('2 花的学校','3 不懂就要问-2','3 不懂就要问-3')
---资 源 id	id	string	@mock=\$order('12110013011810102001535348292873','12110013011810103001535348292873')

6.2.3. 教材列表接口

URL

/saas/out/auth/allTextBookList.json

Method

GET

Description

获取全部教材列表信息接口,返回结果 des 动态加密，加密 key 会在取 token 的时候返回

Request Param List

参数名称	Type	Remark
access_token	string	serviceAuthCode 取回来的服务器授权 token（发送请求的服务器出口地址需跟取服务器授权 token 的服务器一致）
client_id	string	对接渠道 id
appId	string	附加参数，格式为：w+32 为随机数+13 位时间戳，只一次请求有效（java 请参考示例代码中的 main 方法）
sign	string	附加参数，格式为：根据 app_id 使用指定算法生成，只一次请求有效（java 请参考示例代码中的 main 方法）

Response Param List

字段描述	字段名称	字段类型
请求状态	statusCode	number
状态描述	msg	string
	dataList	array<object>
副标题	sub_heading	string

字段描述	字段名称	字段类型
教材出版时间	publication_time	string
学段	rkxd_name	string
年级	nj_name	string
教材名称	name	string
册别	fascicule_name	string
教材内容简介	content	string
章节 json	chapter_json	string
出版社	publisher	string
教材的丛书	series	string
学科	zxxkc_name	string
教材 id	id	number

6.3 数据统计（需走 6.2 服务授权过程）

1. 数据统计

1. 接口名称

获取统计指标数据

2. 请求类型

POST

3. 请求 Url

/saas/out/auth/quota.json

4. 请求参数

参数名	含义	类型	必需	备注
id	指标 ID	String	是	见指标 id 表
group	分组依据	String	否	目前分组数据不完善，此字段预留，但未使用
access_token	服务器授权 token	String	是	接口 6.2.1 返回的 token 字段，需要 urlencode
client_id	渠道 id			

5. 响应参数 返回结果 des 动态加密，加密 key 会在取 token（接口

6.2.1）的时候返回

参数名				含义	类型	备注
err				响应状态码	String	

err msg				响应状态 信息	Stri ng	
da ta				响应数据	Obj ect	
	id			指标 ID	Stri ng	
	na me			指标名称	Stri ng	
	gr oup			指标分组 信息	Obj ect	
		li st		分组数据 列表	List	

6. 响应示例

```
{
  "errno" : "0" ,
  "errmsg" : "success" ,
  "data" :{
    "quota" : "1001" ,
    "name" : "教材数量" ,
    "value" : "12345" ,
    "group" :{
      "list" :[
        {
          "id":1211001101161,
```

```
        "resourceCount": 2425
      },
      {
        "id": 1211001102161,
        "resourceCount": 990
      } ],
    }

  }

}
```

7. 指标 ID 列表

ID	含义	备注
1002	教材使用次数	
1003	教材使用时长	
1101	资源数量	

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