数字教材PC端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 |
| 3.3 | 2025-12-12 |  | 补充单册及工具栏定制接口 |
|  |  |  |  |

# 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。

# 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。

# 业务流程

## 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：

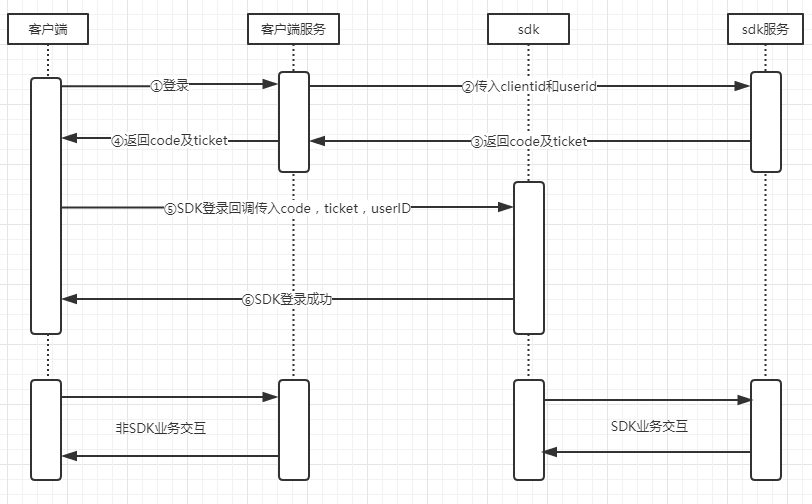
客户端：集成了人教数字教材SDK的用户终端软件

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

## 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

# 相关规则

## 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

## 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"

]

}

}

# 客户端SDK接口说明

## 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 |

1. 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 | JournalsListControl | 用户控件 | CommonParamter.Instance.IsJournalEnable && CommonParamter.Instance.IsTeacher为真是有效 |
| 我的期刊 | MyJournalsControl | 用户控件 | CommonParamter.Instance.IsJournalEnable && CommonParamter.Instance.IsTeacher为真是有效 |

1. 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>

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 | *serverUrl* | 后台服务地址 |
| System.String | *clientId* | 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 |

1. 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 | *code* | 认证code |
| System.String | *ticket* | 请求ticket |
| System.String | *userId* | 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 |

1. 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 | *bookId* | 书籍Id |
| [System.Int32](http://msdn2.microsoft.com/en-us/library/td2s409d) | pageNum | 书籍页码 |
| [System.Windows.Window](http://msdn2.microsoft.com/en-us/library/ms590112) | 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>

1. 登出

◇接口名称

|  |
| --- |
| public void Logout() |

◇参数

无

◇返回值

无

1. 销毁

◇接口名称

|  |
| --- |
| public void Dispose() |

◇参数

无

◇返回值

无

1. 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”

}

]

}

}

1. 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  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 |  |
|  | 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”

}

}

1. 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://127.0.0.1/1001.jpg”

}

]

}

}

1. 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 | ChaptersModelList | 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

1. 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

1. 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

1. 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

1. 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

1. 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

1. 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。**

1. 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

1. 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

1. 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

1. 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

1. 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 | | | |

1. 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 | | | |

1. 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<StringEventArg> 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<StringEventArg> | 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 |
| setImportProgressValueCallBack | 导入进度回调 | SetProcessValue |  |  |

◇返回值

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 含义 | 类型 | 备注 |
| 无 | | | |

1. 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 | public void SetOwnerVisibilityCallBack(SetOwnerVisibility setOwnerVisibilityCallBack) |
| 教材打开结果回调 | public void SetNotifyOpenBookReusltCallBack(NotifyOpenBookReuslt notifyOpenBookReusltCallBack) |
| 移除回调 |  |
| 移除设置教材打开、关闭触发回调 | public void RemoveOwnerVisibilityCallBack(SetOwnerVisibility setOwnerVisibilityCallBack) |
| 移除教材打开结果回调 | public void RemoveNotifyOpenBookReusltCallBack(NotifyOpenBookReuslt notifyOpenBookReusltCallBack) |

下载教材回调：

◇接口名称

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

1. **自定义工具条对应接口：**
2. 翻页

◇接口名称public void TurnPage(bool isRightTurn)

### 参数

| 参数名 | 含义 |  |  | 类型 | 必需 | 备注 |
| --- | --- | --- | --- | --- | --- | --- |
| isRightTurn | 翻页方向 |  |  | bool | 是 | true: 右翻页 false: 左翻页 |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 获取目录数据（目录树数据源取得）

◇接口名称public List<ChaptersModel> GetChapterListData(string strTbid)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| strTbid | 教材 id | string | 是 | - |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 返回值 | 目录树数据列表 | List<ChaptersModel> | - |

◇返回示例略

1. 根据目录信息跳转到目录对应的初始页码

◇接口名称public void GotoBookPage(ChaptersModel chapter)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| chapter | 目录信息 | ChaptersModel | 是 | - |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 获取章节

◇接口名称public ChaptersModel GetChapterById(string strTbId, string chapterId)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| strTbId | 教材 ID | string | 是 | - |
| chapterId | 章节 ID | string | 是 | - |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 返回值 | 章节信息 | ChaptersModel | - |

◇返回示例略

1. 资源开关

◇接口名称public void ResSwitch()

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 | - | - | - |

◇备注内置开关标识字段：CommonParamter.Instance.ResSwitch，调用一次该方法，该字段反转一次值

◇返回示例略

1. 热区开关

◇接口名称public void HotZoneSwitch()

### 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 | - | - | - |

◇备注

1. 内置开关标识字段：CommonParamter.Instance.HotZoneSwitch，调用一次该方法，该字段反转一次值；

2. 资源开关开的状态下才能控制热区开关

◇返回示例

略

1. 放大教材

◇接口名称public void TextbookZoomIn()

### 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 | - | - | - |

◇备注每调用一次，放大原来的 25%

◇返回示例略

1. 还原教材

◇接口名称public void TextbookRestore()

### 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 | - | - | - |

◇备注还原到放大前的自适应状态

◇返回示例略

1. 单双页切换

◇接口名称public void ChangeBookPageLayout()

### 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

1. 获取当前页面的单双页状态

◇接口名称public TextbookPageLayout GetCurrentBookPageLayout()

### 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** |
| --- | --- | --- | --- |
| 返回值 | 当前页面的单双页状态 | TextbookPageLayout | - |

◇返回示例略

1. 显示笔记列表

◇接口名称public void ShowNote()

### 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 | - | - | - |

◇备注内置功能◇返回示例略

1. 显示或隐藏批注工具条

◇接口名称public void ShowOrHidePdfCommentToolbar(bool isShow = true)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| isShow | 是否显示批注工具条 | bool | 否 | 默认值：true |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 长截图

◇接口名称public void LongClipTextbook(Action<FillRectDataRoot, ChaptersInfo> SavedStaticClips)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| SavedStaticClips | 截图保存数据的回调方法 | Action<FillRectDataRoot, ChaptersInfo> | 是 | 接收长截图保存的数据 |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 静态截图

◇接口名称public void ClipTextbook(Action<FillRectDataRoot, ChaptersInfo> SaveAction)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| SaveAction | 截图保存数据的回调方法 | Action<FillRectDataRoot, ChaptersInfo> | 是 | 1. 接收截图保存的数据；  2. 仅限于截取教材对应的内容图片 |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 动态截图

◇接口名称public void DynamicClipsTextbook(Action<FillRectDataRoot, ChaptersInfo> SavedDynamicClips)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| SavedDynamicClips | 截图保存数据的回调方法 | Action<FillRectDataRoot, ChaptersInfo> | 是 | 1. 接收动态截图保存的数据；  2. 仅限于截取教材对应的内容图片 |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 云资源获取方法

◇接口名称public EmbedResourceReponse GetNQCloudRes(ShareResParam param)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| param | 云资源查询参数 | ShareResParam | 是 | 使用示例：  var param = new ShareResParam  {  Dzwjlx = QueryParam.EmbedResFormat.Id,  PageNo = 1,  PageSize =14  };  param.OriTreeCode = 章节 id;  param.Keywords = "搜索关键字"; |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 返回值 | 云资源查询结果 | EmbedResourceReponse | - |

◇返回示例略

1. 获取云资源格式列表

◇接口名称public List<EmbedResFormatModel> GetResFormatList()

### 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** |
| --- | --- | --- | --- |
| 返回值 | 云资源格式列表 | List<EmbedResFormatModel> | - |

◇返回示例略

1. 打开云资源

◇接口名称public void OpenCloudRes(string resId)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| resId | 云资源 ID | string | 是 | EmbedResItemModel 下的 Id |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 获取当前打开教材对应的更多工具列表

◇接口名称public MetaList GetMoreTools()

### 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** |
| --- | --- | --- | --- |
| 返回值 | 更多工具列表 | MetaList | - |

◇返回示例略

1. 导出教材资源包

◇接口名称public void ExportTextbook()

### 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

1. 打开边栏

◇接口名称public void OpenSideBar()

### 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

1. 切换到演示模式

◇接口名称public void SwitchToShowModel(bool isOpen = true)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| isOpen | 是否打开演示模式 | bool | 否 | 1. true: 打开演示模式，false: 关闭演示模式；  2. 默认值：true |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 25 退出教材

◇接口名称public void ExitTextbook()

### 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

1. 26 打开演示模式下白板功能

◇接口名称public void OpenWhiteBoardFun()

### 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 | - | - | - |

◇备注

1. 演示模式下可用；

2. 前提是需要开启演示模式

◇返回示例

略

1. 打开画笔

◇接口名称public void OpenDrawModel()

### 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 | - | - | - |

◇备注演示模式下可用◇返回示例略

1. 设置画笔模式

◇接口名称public void SetDrawModel(int model)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| model | 画笔模式 | int | 是 | 0 画笔，1 激光笔，2 荧光笔 |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 设置画笔颜色

◇接口名称public void SelectPenColor(SolidColorBrush brush)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| brush | 画笔颜色 | SolidColorBrush | 是 | - |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 打开橡皮功能

◇接口名称public void OpenEraseModel()

### 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 | - | - | - |

◇备注演示模式下可用

◇返回示例略

1. 清除所有画笔内容

◇接口名称public void ClearDrawContent()

### 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

1. 撤销画笔动作

◇接口名称public void RevertDrawAction()

### 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 | - | - | - |

◇备注演示模式下可用

◇返回示例略

1. 退出画笔

◇接口名称public void ExitDrawModel()

### 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

1. 切换到鼠标

◇接口名称public void SwitchToMouse()

### 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

1. 切换选择模式

◇接口名称public void SwitchToSelectModel()

### 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

1. 添加自定义图形

◇接口名称public void AddUserPath(Point p, UserDrawOpreateType opreateType, int dataType)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| p | 坐标点 | Point | 是 | - |
| opreateType | 操作类型 | UserDrawOpreateType | 是 | - |
| dataType | 数据类型 | int | 是 | - |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 打开计时器

◇接口名称public void OpenTimerWidow()

### 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

1. 打开放大镜

◇接口名称public void OpenSpotlight()

### 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

1. 设置自定义工具条

◇接口名称public void SetToolBar(TextbookBaseToolBar baseToolBar)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| baseToolBar | 基础工具条对象 | TextbookBaseToolBar | 是 | sdk 初始化后调用 |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

1. 设置演示模式工具条

◇接口名称public void SetShowModelToolBar(TextbookBaseToolBar bar)

### 参数

| **参数名** | **含义** | **类型** | **必需** | **备注** |
| --- | --- | --- | --- | --- |
| bar | 基础工具条对象 | TextbookBaseToolBar | 是 | - |

### 返回值

| **参数名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| 无 | - | - | - |

◇返回示例略

注意：以上自定义工具条功能与后台授权与否联动，不授权，功能不可用。

对应使用示例请参考DEMO中的对应代码，编辑工具对应UserBottomBar.xaml，演示工具对应UserShowModelToolBarControl.xaml，此处略。

1. 使用示例
2. 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);  } |

1. 登录

|  |
| --- |
| private void LoginCtrl\_OnLoginClick(object sender, DemoLoginEventArgs e)  {  PepSdkContext.Instance.Login(e.LoginData.Code, e.LoginData.Ticket, e.LoginData.UserId);  } |

1. 登出

|  |
| --- |
| private void LoginCtrl\_OnLogoutClick(object sender, DemoLoginEventArgs e)  {  PepSdkContext.Instance.Logout();  } |

1. 销毁

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

1. 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          } |

1. 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          } |

## Android端集成

### **(1)运行环境**

|  |  |
| --- | --- |
| 开发语言 | Java |
| 支持Android版本 | Android 5.0及以上 |

### **(2)注意**

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:1.6.2'**

**注:1.6.2是版本号, 接入时请以sdk的实际版本为准**

### **(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.netAccessTokenLoginnetAccessTokenLogin(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"  /> |

#### 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

#### 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  }  }); |

### 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,停止下载 (删除下载信息)

*/\*\*  
 \** ***@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)可选择的出版社列表

List<ContentBean> publisherContent = staticCascadeModel.static\_metadata.publisher.content;

###### (2)可选择的学段列表

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*  
 }

}

###### 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.ContentBean> content = **staticCascadeModelData**.**cascade\_metadata**.**primary**.**content**;

③初中学段可选学科

List<StaticCascadeModel.CascadeMetadada.CascadeBean.ContentBean> content = **staticCascadeModelData**.**cascade\_metadata**.**junior**.**content**;

④高中学段可选学科

List<StaticCascadeModel.CascadeMetadada.CascadeBean.ContentBean> 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  和  javax.net.ssl.SSLException | 网络断开时可能会出现 |
| FileDownloadNetworkPolicyException | 设置了BaseDownloadTask#setWifiRequired(true)，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 |
| PathConflictException | 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 导入资源文件

直接拖到新项目中

1 framework : 3个:

PDFReaderSDKTextBook.framework

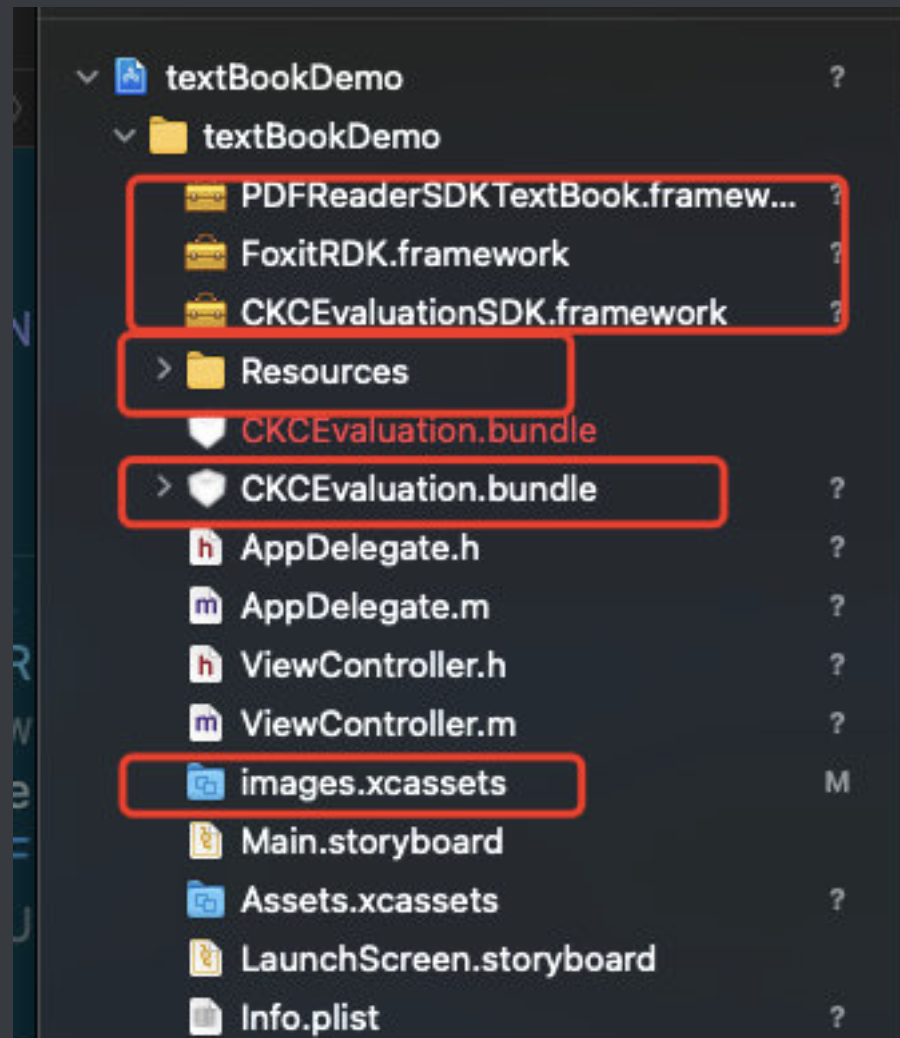
FoxitRDK.framework

CKCEvaluationSDK.framework

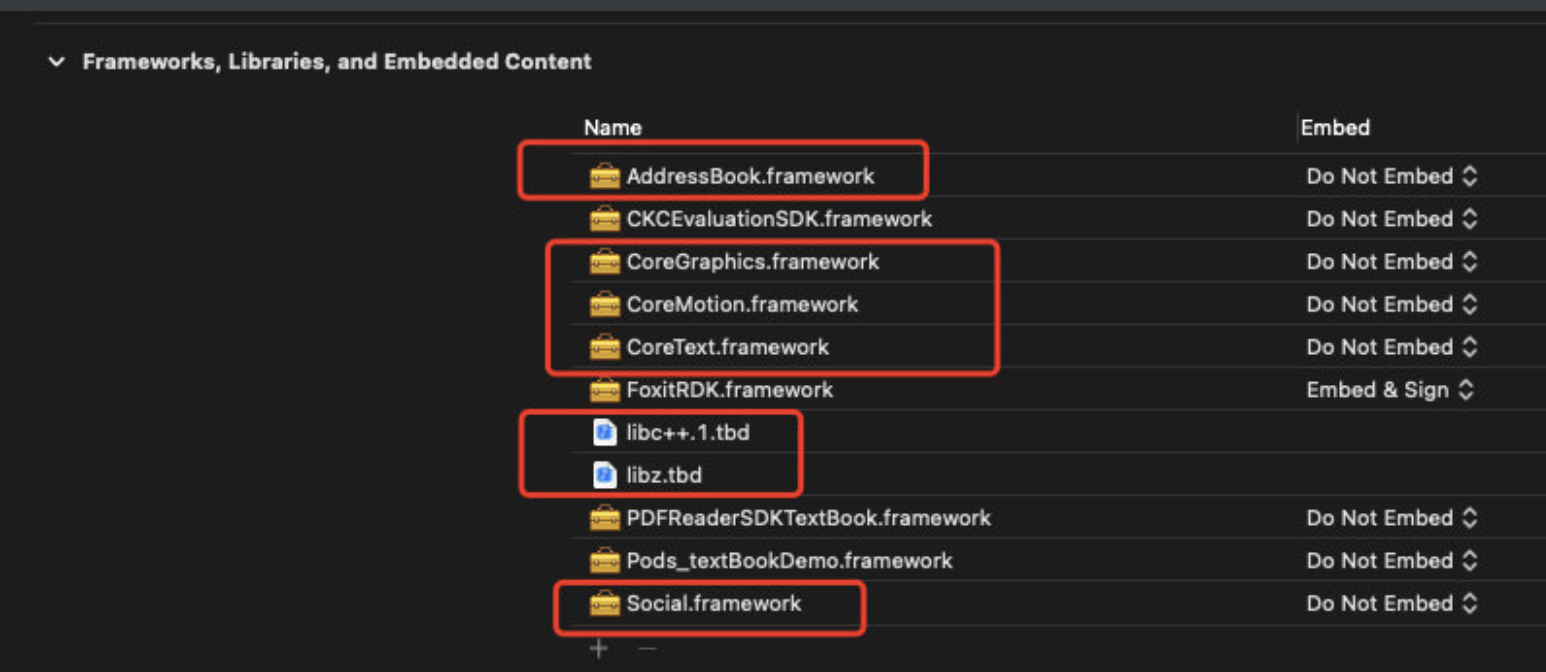
2 Resources 文件夹

3 CKCEvaluation.bundle

4 images.xcassets

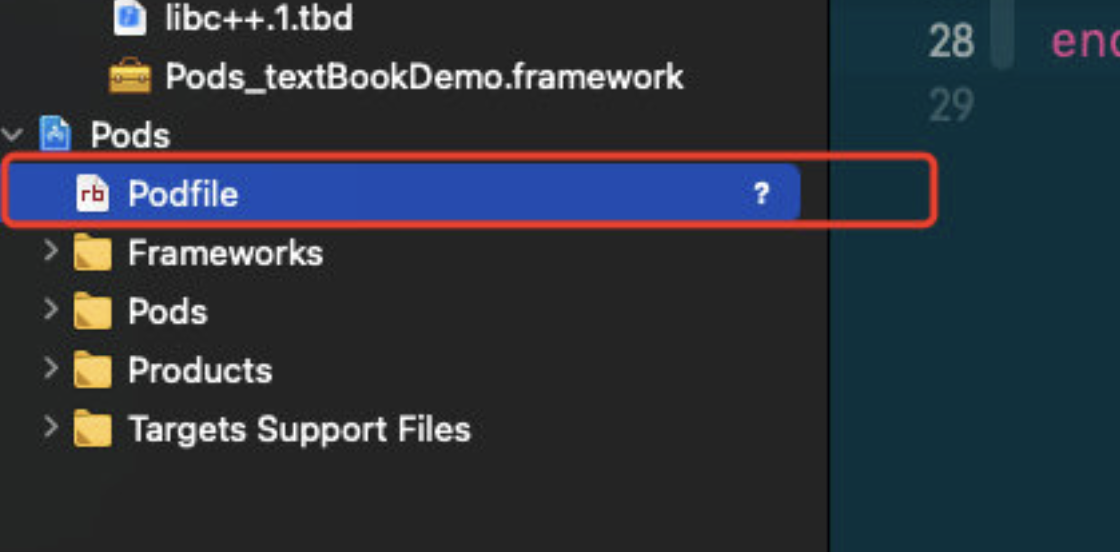


2 项目中配置增加 framework



3 配置pods

参考demo ，在 Podfile 中增加



Podfile

pod 'AFNetworking'

pod 'Masonry'

pod 'SDWebImage'

#SDK

pod 'PEPNetworking' #, :git =>

'https://github.com/PEPDigitalPublishing/PEPNetworking.git'

pod 'PEPBigData' #, :git =>

'https://github.com/PEPDigitalPublishing/PEPBigData.git'

pod 'ZSSRichTextEditor'#, :git =>

'https://github.com/nnhubbard/ZSSRichTextEditor.git'

pod 'YYModel'

#pod 'YYCategories'

pod 'SSZipArchive'

pod 'SAMKeychain'

pod 'FMDB'

pod 'MBProgressHUD'

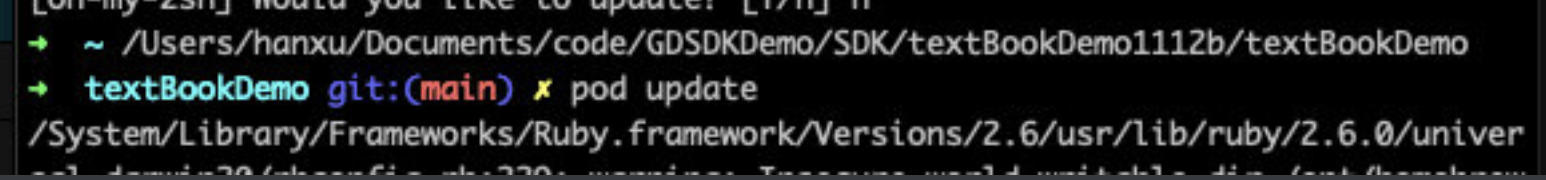
pod 'MJRefresh'

#pod 'PEPiFlyMSC', :git =>

'https://github.com/PEPDigitalPublishing/PEPiFlyMSC.git'

pod 'ChameleonFramework'

配置完成，使用命令行 pod update



4 初始化

参考 demo中 的 ViewController.m

#import <PDFReaderSDKTextBook/PDFReaderSDKTextBook.h>

#import <Masonry/Masonry.h>

#import <AFNetworking.h>

#import <PDFReaderSDKTextBook/PEPHttpUtil.h>

#import <PDFReaderSDKTextBook/PEPURLPath.h>

获取注册码:

NSDictionary \*param1 =@{@"user\_id": UserID,

@"client\_id": SDKClientAppID,

};

[PEPHttpUtil getWithPth:kAPI\_UserAuth2 params:param1

progress:^(NSProgress \*downLoadProgress) {

} success:^(id JSON) {

[MBProgressHUD hideHUD];

if ([JSON[@"statusCode"] isEqualToString:@"500110"]) {

NSMutableDictionary \*dict = [NSMutableDictionary

dictionaryWithDictionary:JSON[@"dataMap"]];

self.authorize\_code = dict[@"authorize\_code"];

[MBProgressHUD showNormalText:@"已获取授权码"]; }else{

[MBProgressHUD showNormalText:@"用户名或密码错误"]; }

} failure:^(NSError \*error) {

[MBProgressHUD showNormalText:@"请求失败"];

}];

进入教材

- (void)setPEPTextbookConfig

{

NSMutableDictionary \*dic = [NSMutableDictionary dictionary];

if (self.authorize\_code.length == 0) { [MBProgressHUD showNormalText:@"请获取授权码"]; return;

}

dic = (NSMutableDictionary \*)@{@"ticket":@"ticket",

@"code":self.authorize\_code,

};

[self loginWithCodeDic:dic];

}

- (void)loginWithCodeDic:(NSDictionary \*)param

{

PEPReaderTextBookManager \*manager = [PEPReaderTextBookManager

PEPShareManagerWithAPPID:SDKClientAppID];

NSString \*baseurl = kPEPBaseURL;

[manager setBaseURL:baseurl];

[manager PEPLoginWithAuthCode:param andCallBack:^(id resopnse) {

PEPTextbookViewController \*pepVC = [[PEPTextbookViewController

alloc]init];

@"user\_id":UserID,

@"client\_id":SDKClientAppID,

@"device\_type":@"2",

pepVC.hidesBottomBarWhenPushed = YES;

self.tabBarController.tabBar.hidden = YES;

[self.navigationController pushViewController:pepVC

animated:YES];

}]; }

其他功能⻅ PEPReaderTextBookManager 类

/\*\*

PEPReaderTextBookManager 实例化

@return PEPReaderTextBookManager 单例对象

\*/

+ (instancetype)PEPShareManagerWithAPPID:(NSString \*)APPID;

/\*\* 设置SDK的BaseUrl

@param URLString kBaseUrl

\*/

- (void)setBaseURL:(NSString \*)URLString ;

/\*\*

登录

\*/

//- (void)getUserAuthWithParam:(NSDictionary \*)dic;

- (void)PEPLoginWithAuthCode:(NSDictionary \*)codeDictionary

andCallBack:(void (^) (id resopnse))callBack;

/// 是否已登录状态

- (BOOL)PEPIsOnline;

/// 激活教材

/// @param activateString 激活码

- (void)PEPActivateBooksWithCode:(NSString \*)activateString andBookId: (NSString \*)bookId;

/\*\*

退出垂直平台 清除用户信息

\*/

- (void)PEPQuitCZPTuserAccount;

/\*\*

查询教材列表

@param dic 请求参数{rkxd:学段,zxxkc:学科,nj:年级,name:教材名称(支持模糊搜 索),pageno:分⻚⻚码(int 默认1),pagesize:分⻚大小(int 默认10),detail:是否返回 详情(String类型,true/false;默认false不返回)} 参数皆为非必需

@param finished 成功/失败完成后回调

\*/

- (void)queryTextbookListWithParam:(NSDictionary \*)dic withBlock:

(CompleteBlock)finished;

/\*\*

查询教材详情

@param dic 请求参数{id:教材ID(String,必需),chapter:是否返回章节(String类 型,true/false;默认false不返回)}

@param finished 成功/失败完成后回调

\*/

- (void)queryTextbookDetailWithParam:(NSDictionary \*)dic withBlcok:

(CompleteBlock)finished;

/\*\*

获取常量接口

\*/

- (void)queryStaticData;

/\*\*

未登录状态下查询设备列表

\*/

- (void)noLoginQueryDeviceList;

/\*\*

查询常量表是否过期

\*/

//- (void)queryStaticFileModified;

#pragma mark - 新增单册书本下载接口 -

#pragma mark - 下载相关 -

//下载书 根据章节数组

- (void)PEPDownloadResourceWaithChapterArray:(NSArray \*)chapterArray bookID:(NSString \*)bookID;

//下载整本书书 根据id

- (void)PEPDownloadZipAndResourceWithBookID:(NSString \*)bookID; //获取下载状态 0-完成 1-已下载了部分(继续下载) 2-更新 3-取消(正在下载)4-删 除 5-下载

- (NSInteger)PEPGetBookDownloadStatusWithBookID:(NSString \*)bookID;

// 根据bookID删除教材

- (void)PEPDeleteBookWithBookID:(NSString \*)bookID;

// 更新

- (void)PEPUpdateBookWithBookID:(NSString \*)bookID;

// 暂停下载

- (void)PEPSuspendDownloadWithBookID:(NSString \*)bookID; // 继续下载

- (void)PEPContinueDownloadWithBookID:(NSString \*)bookID;

#pragma mark - 数据相关 -

// 获取我的教材书本信息(下载中和已下载)

- (NSArray \*)PEPGetMyTextBook;

// 打开书籍

- (void)PEPOpenBookWithBookID:(NSString \*)bookID;

// 查询教材详情

- (void)getPEPTextbookDetailWithBookID:(NSString \*)bookID withBlock: (CompleteBlock)completeBlock;

// 查询我的下载历史

- (void)getPEPTextbookDownHositryWithBlock: (CompleteBlock)completeBlock;

# SDK服务端相关接口说明

## 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 |  |

## 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 | 睿易对接项目client\_id |

Response Param List

| Name | Identifier | Type | Remark |
| --- | --- | --- | --- |
| 请求状态 | statusCode | number |  |
| 状态说明 | msg | string |  |
| 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

| Name | Identifier | Type | Remark |
| --- | --- | --- | --- |
| 教材id | id | string |  |
| 页码，双页用逗号隔开 | pageNo | string | 示例：单页2；双页2，3 |
| token | access\_token | string |  |

Response Param List

| Name | Identifier | Type | Remark |
| --- | --- | --- | --- |
| 500100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 |
|  | dataList|9 | array<object> |  |
| ---资源格式 | file\_format | string | @mock=$order('.jpg','.jpg','.mp3','.mp3','.mp3','.mp3','.mp3','.mp3','.mp3') |
| ---资源名称 | title | string | @mock=$order('2 花的学校','3 不懂就要问-2','yw\_3s\_008\_diandu01','yw\_3s\_008\_diandu02','yw\_3s\_008\_diandu03','yw\_3s\_008\_diandu04','yw\_3s\_008\_diandu05','yw\_3s\_008\_diandu06','yw\_3s\_008\_diandu07') |
| ---资源id | id | string | @mock=$order('12110013011810102001535339178976','12110013011810103001535339178986','12110013011810103001535348292870','12110013011810103001535348292871','12110013011810103001535348292872','12110013011810103001535348292873','12110013011810103001535348292874','12110013011810103001535348292875','12110013011810103001535348292876') |

## 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

| Name | Identifier | Type | Remark |
| --- | --- | --- | --- |
| 服务器授权token | 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） |
| 对接client\_id | client\_id | string | client\_id |

Response Param List

| Name | Identifier | Type | Remark |
| --- | --- | --- | --- |
| 请求状态 | statusCode | number |  |
| 状态描述 | msg | string |  |
|  | dataList | array<object> |  |
| ---副标题 | sub\_heading | string |  |
| ---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
2. 接口名称

获取统计指标数据

1. 请求类型

POST

1. 请求Url

/saas/out/auth/quota.json

1. 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 |  |  |  |

1. 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 |
| errno |  |  |  | 响应状态码 | String |  |
| errmsg |  |  |  | 响应状态信息 | String |  |
| data |  |  |  | 响应数据 | Object |  |
| id |  |  | 指标ID | String |  |
| name |  |  | 指标名称 | String |  |
| group |  |  | 指标分组信息 | Object |  |
| list |  | 分组数据列表 | List |  |

1. 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**  
                } ],

}

}

}

1. 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 |  |

# 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