**文档中台**

**使用说明书**

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订人** | **修改日期** | **修改说明** |
| V5.1.4 | 高成柳 | 2021-03-05 | 创建文档 |
| V5.1.5 | 高成柳 | 2021-05-11 | 增加Demo使用说明 |
| V5.1.6 | 高健 | 2021-06-29 | 增加在线预览和业务唯一标识说明 |
| V5.1.700 | 赵一帆 | 2021-10-28 | 更新javaSDK、JS SDK |
| V5.1.700 | 高健 | 2021-11-5 | 增加3.3.1.2证书回调说明 |
| V5.1.700 | 高健 | 2021-11-11 | 增加企业内建应用说明；  调整aksk获取方法；  增加preview js sdk、weboffice sdk说明； |
| V5.1.700 | 高健 | 2021-12-20 | 增加bugfix\_release\_yundoc\_v5.1.7\_20211125版本带出的OPENSDK说明5.1.3节 |

目录

[1. 使用概述 1](#_Toc1505499003)

[1.1. 产品概述 1](#_Toc7771354)

[1.2. 使用环境 1](#_Toc1197168168)

[1.3. 服务使用 1](#_Toc597413046)

[2. 服务功能 3](#_Toc174602134)

[2.1. 服务架构 3](#_Toc496046965)

[2.1.1. 产品架构 3](#_Toc1550372831)

[2.1.2. 技术架构 4](#_Toc739485316)

[2.2. 功能说明 4](#_Toc281573872)

[2.2.1. 应用管理 4](#_Toc1224788232)

[2.2.2. 在线编辑 6](#_Toc807878381)

[2.2.3. 在线预览 8](#_Toc172791147)

[2.2.4. 格式处理 9](#_Toc2093618293)

[2.2.5. 应用文档 9](#_Toc1859598303)

[3. 对接说明 10](#_Toc1328840473)

[3.1. 服务授权 10](#_Toc2049538759)

[3.2. 业务唯一标识 10](#_Toc1430835527)

[3.3. 进行接入 11](#_Toc1088137339)

[3.3.1. 可视化应用管理 11](#_Toc571736908)

[3.3.2. 平台对接应用管理 16](#_Toc452053692)

[3.3.3. 应用能力实现 21](#_Toc1956022773)

[3.4. 能力接入 21](#_Toc347989039)

[3.4.1. 在线编辑 21](#_Toc835451035)

[3.4.2. 在线预览 22](#_Toc1172699414)

[3.4.3. 格式处理 22](#_Toc1072528341)

[4. 接口签名 23](#_Toc1857270658)

[4.1. 签名描述 23](#_Toc1364217973)

[5. SDK&Demo 24](#_Toc1897865063)

[5.1. SDK使用说明 24](#_Toc1556177154)

[5.1.1. JAVA SDK 24](#_Toc1829828833)

[5.1.2. Go SDK 25](#_Toc1743231581)

[5.1.3. Preview JS SDK 26](#_Toc914192509)

[5.1.4. Weboffice JS SDK 31](#_Toc1837600187)

[5.2. DEMO使用说明 31](#_Toc792916101)

# 使用概述

## 产品概述

WPS文档中台是WPS云办公产品中对第三方提供围绕文档处理、文档管理，帮助企业业务服务实现在线协同编辑，提升企业办公效率的赋能平台。WPS文档中台提炼WPS产品体系中对文档和协同办公的处理能力，通过标准Oauth2.0认证授权体系将能力进行授权封装，将能力进行原子化输出，为企业提供多元化的办公新体验。

## 使用环境

为保证最佳体验效果，建议使用谷歌浏览器（推荐Chrome 80以上内核版本）或基于谷歌内核的浏览器（如：QQ浏览器、EDGE，火狐、360安全浏览器V10（V10.4.1003.2，Chromium86））。若使用其他浏览器（如IE11、Firefox等），部分功能可能会有效果缺失。

## 服务使用

WPS文档中台是一个面向开发者的产品，应用管理、授权管理、应用配置可通过开放平台进行可视化的管理；同时也支持通过接口来完成应用的管理、授权的管理、以及接口的授权，通过接口管理应用需在创建第三方应用之前通过WPS实施同学获取授权AKSK，在调用接口时带上AKSK完成授权签名即可完成应用的创建以及应用相关的配置信息设置。

参考资料：

文档中台-开放平台接口说明

文档中台-demo-demo使用说明

# 服务功能

文档中台将提供的接口分为四个部分的能力模块（后续版本会逐渐丰富能力集合）：在线编辑、在线预览、格式处理、应用管理。

## 服务架构

### 产品架构

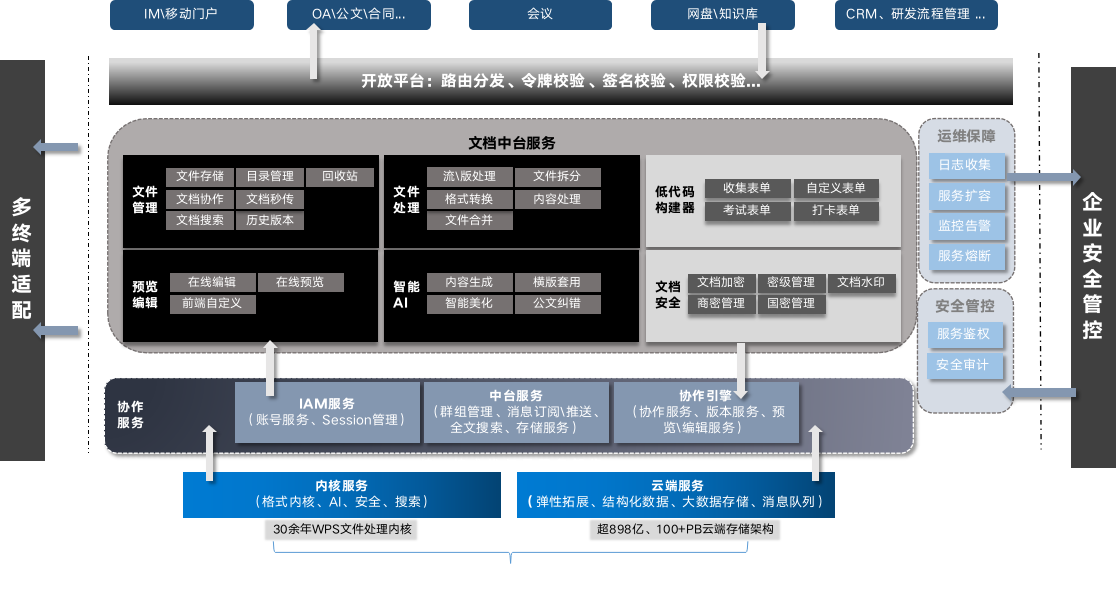


图 1 产品架构图

### 技术架构

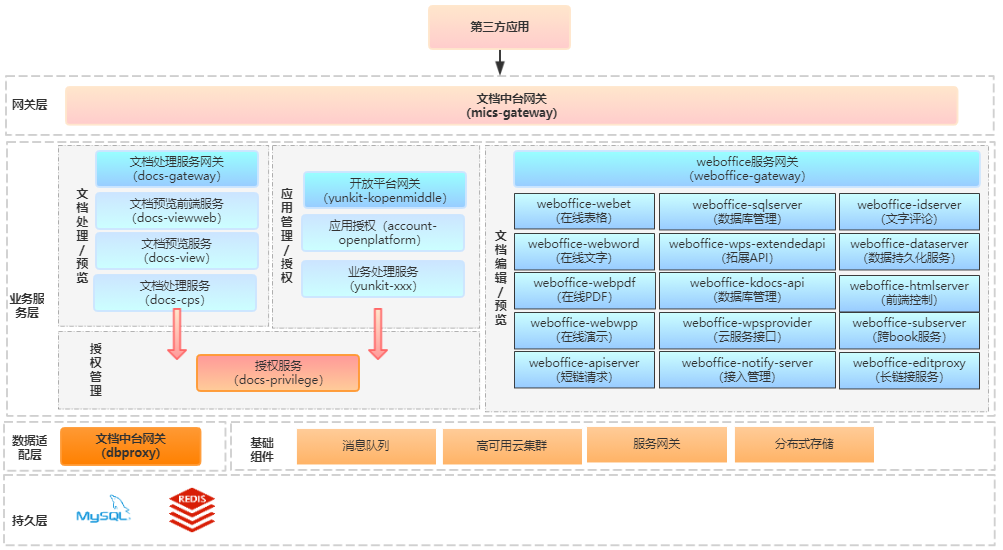


图 2 技术架构图

## 功能说明

### 应用管理

应用管理能力是文档中台为对接第三方应用管理平台开放的能力，适用场景：

|  |  |  |
| --- | --- | --- |
| **对接方角色** | **对接场景** | **使用说明** |
| 企业IT信息中心 | 对接应用运营中心，根据流程申请完成能力注册与管理 | 信息部门将文档中台做全集团采购的能力中台，通过企业自建的流程管理系统，将文档中台的应用申请、上架、密钥下放通过IT流程进行管理。 |
| 第三方运营商 | 将文档中台进行封装，付费售卖 | 运营商将文档中台的能力完成套餐定义，通过商业化订单完成应用能力的授权与套餐定义，针对不同套餐定义不同的售卖价格，完成订单后下发密钥。 |

在调用在线编辑、在线预览和格式处理三个模块前，调用方需要通过此能力开放出的应用创建、应用配置信息设置（如应用白名单、回调实现地址等）、获取应用列表和应用详情，完成对应用的管理实现。

### 在线编辑

第三方应用提供获取文件的接口，以及文件相关的信息，通过文档中台服务中的在线处理服务打开在线文件。应用可实现以下接口并接收相关推送后，可实现版本管理，文件编辑权限管控、历史版本预览、打开（或模板）文件等。



图 3 在线编辑支持格式

接口能力说明：

|  |  |  |
| --- | --- | --- |
| **开放接口** | **接口路径** | **接口描述** |
| 获取在线编辑链接 | /weboffice/v2/url | 获取文件在线编辑地址 |
| 用户会话的身份校验 | 前端sdk | 前端验证操作用户的权限 |
| 获取文件元数据 | /v1/3rd/file/info | 从第三方获取文件信息，包括：创建用户、水印、权限、文件类型及大小 |
| 获取用户信息 | /v1/3rd/user/info | 从第三方获取用户信息，包括：头像、用户名、用户id |
| 通知此文件目前有哪些人正在协作 | /v1/3rd/file/online | 通知第三方当前文档上用户的变动信息 |
| 上传文件新版本 | /v1/3rd/file/save | 往第三方推送文档变更信息，当文档保存时推送 |
| 获取特定版本的文件信息 | /v1/3rd/file/version/:version | 从第三方获取指定版本的文件信息，用于实现历史版本预览 |
| 文件重命名 | /v1/3rd/file/rename | 推送文档新的名称，第三方实现文件重命名，当文档被修改名称后推送 |
| 获取所有历史版本文件信息 | /v1/3rd/file/history | 从第三方获取历史版本信息，用于展示文档历史版本纪录 |
| 新建文件 | /v1/3rd/file/new | 从第三方获取模板文件后，跳转打开对应的文档进行编辑 |
| 回调通知 | /v1/3rd/onnotify | 当文档被打开或关闭时推送相关信息到第三方 |

典型场景：

* 为具有企业自建云盘的产品提供流式文档在线协同编辑的能力。
* 为企业IM或门户系统提供协同编辑的能力，解决相互传递发送导致版本混乱无法准确表达的问题。

### 在线预览

第三方应用实现获取文件相关的信息，通过文档中台服务中的在线预览服务打开在线文件，应用可自定义文档预览水印样式。

在线预览实现分为：四大组件格式预览服务及图片、压缩包（含压缩包），皆使用WPS内核转换后，通过前端（svg、cavas）渲染技术实现不同类型的文档的预览。

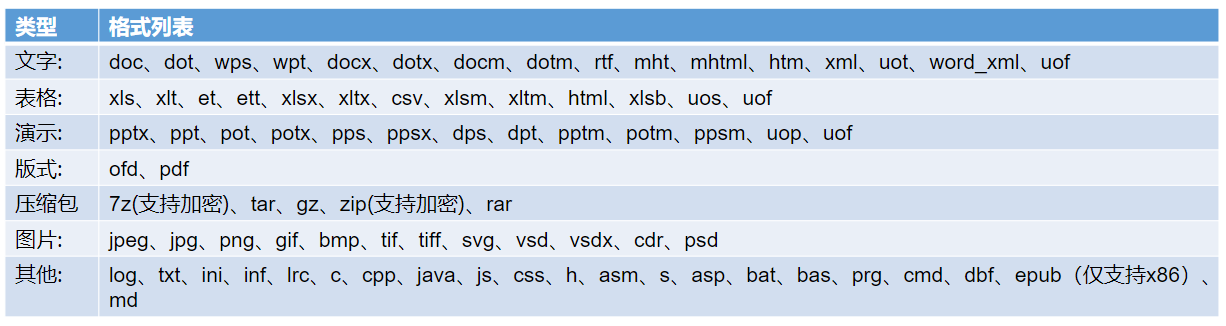


图 4 在线预览支持格式

典型场景：

* 业务系统去插件化、解决因浏览器版本或和容性问题，无法通过SDK调起客户端进行预览，从而转换到浏览器中进行文档的在线预览。
* 在移动门户上对业务系统中的文档附件进行在线预览。

### 格式处理

格式处理模块包括：格式转换、文件查询、内容操作、文档合并、文档拆分、图片处理、智能公文、多标签替换。

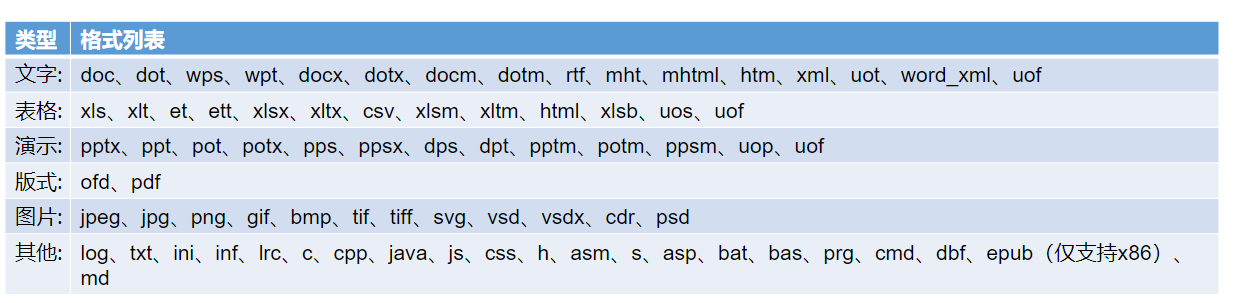


图 5 格式转换支持格式

* 支持将对文档添加红头、拆分、合并、智能公文、水印、清稿、图片处理；
* 支持将以上格式转换为OFD、PDF、图片；
* 支持通过网络地址获取文件；

典型场景：

* 解决企业在线上完成公文或合同内容的拟稿，并在业务流程完成后，套上公文红头，并将文档转换为可公开发布的版式文件进行发布和归档。
* 处理公文&合同文档，将文档正文与附件进行合并或拆分。

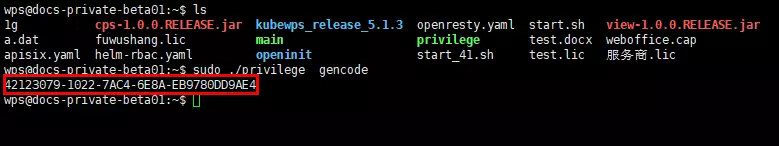
### 应用文档

[应用文档使用说明书](https://kdocs.cn/l/ceCmYhBIhWQK)

# 对接说明

## 服务授权

执行工具获取企业唯一码（[文档中台与云文档服务通用工具](https://www.kdocs.cn/l/cj86pYUh2QcH)），若存在多个机器，需要在每个节点上执行，获取后将多台机器授权码通过英文分号完成拼接。将授权相关信息周知到客户成功部李聪（或各大区组长）申请许可证试用授权，正式授权请找对应商务部门。



工具下载地址：https://www.kdocs.cn/l/cjsAbhYMBQoN

## 业务唯一标识

在线编辑、在线预览、内容处理本次升级到V2接口，三个能力的接口参数都需要添加业务唯一标识。通过记录业务唯一标识，可实现对文档中台能力的活跃统计。

业务唯一标识是业务方区分业务的唯一id，比如OA流程ID、合同ID等。由应用自定义，64个字符以内，仅支持字母、数字、下划线且保证在同一应用内不重复。

## 进行接入

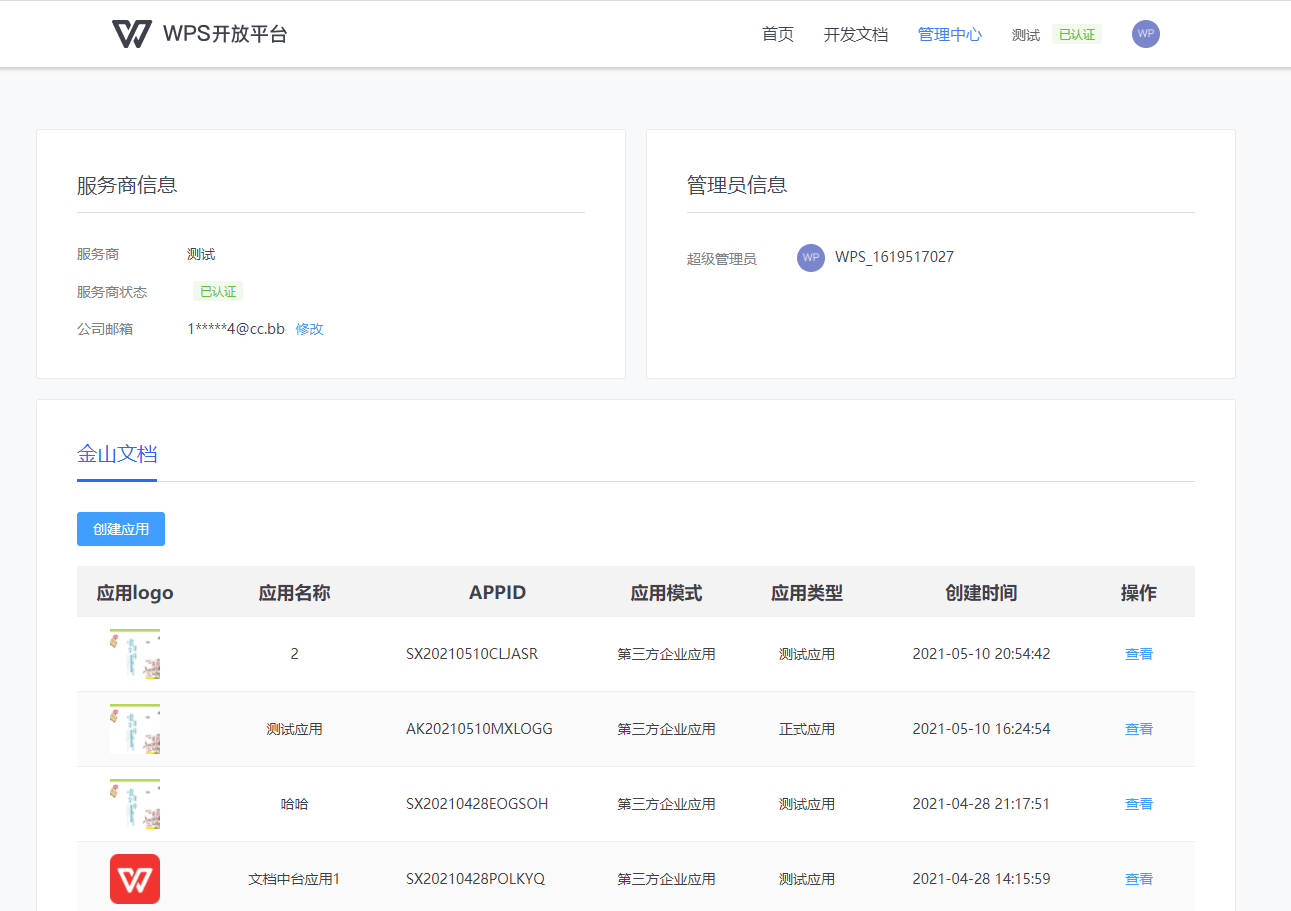
### 可视化应用管理

访问文档中台可视化服务：开放平台，http(https)://{domain}/open。使用wpsadmin/Wps@123456完成登录，可通过可视化界面完成应用的创建和管理、开发对接。

登录到管理后台http(https)://{domain}/openplatform，可对在可视化平台创建应用的能力申请进行审核。

#### 应用管理

点击页面左上角的【创建应用】，输入应用相关必填信息，点击保存即可：



**文档中台应用类型为第三方企业应用和企业内建应用。**

|  |  |
| --- | --- |
| **名称** | **说明** |
| 第三方企业应用 | 由平台服务商或已认证的企业服务商通过开放平台创建的应用，此类应用可以在被企业管理员授权之后，对应WPS+企业用户使用 |
| 企业内建应用 | 由企业服务商通过开放平台创建的应用，此类应用仅能被应用创建的WPS+企业内的企业用户使用 |

在应用列表点击【查看】，跳转到应用详情。可查看appid和appkey。并配置应用授权相关回调设置：授权回调地址，以及ip白名单等。

**注：应用相关的配置项每5分钟生效一次。**

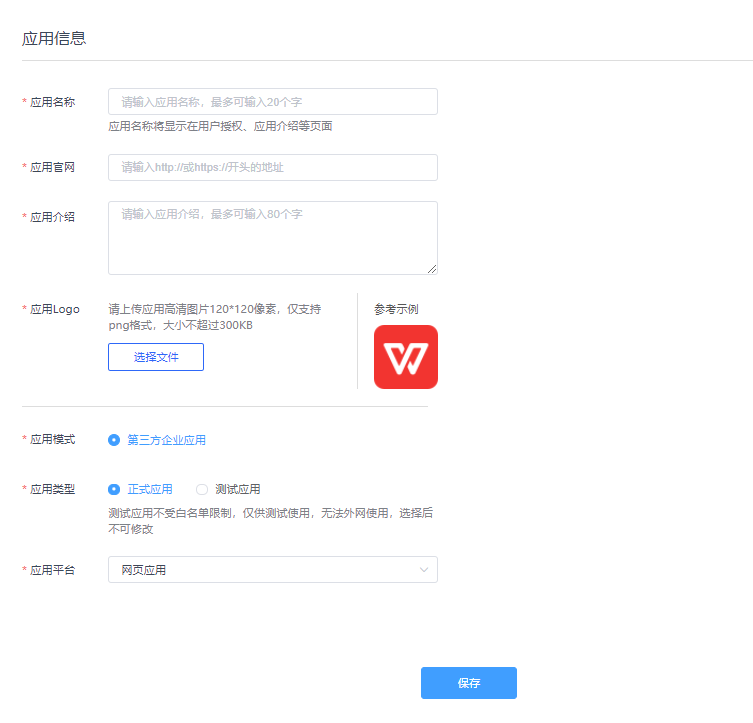


图 6 应用信息配置页

#### 证书回调

证书回调是通过回调形式在证书到期前90天每天9:00向应用方推送当前证书信息，防止证书到期后能力无法使用。

注意：该配置为必填项目，如果不填写无法申请新权限！



#### 应用授权

在应用进行接入能力权限申请后，需管理员在管理系统后台对申请的能力进行授权。



图 7 能力申请

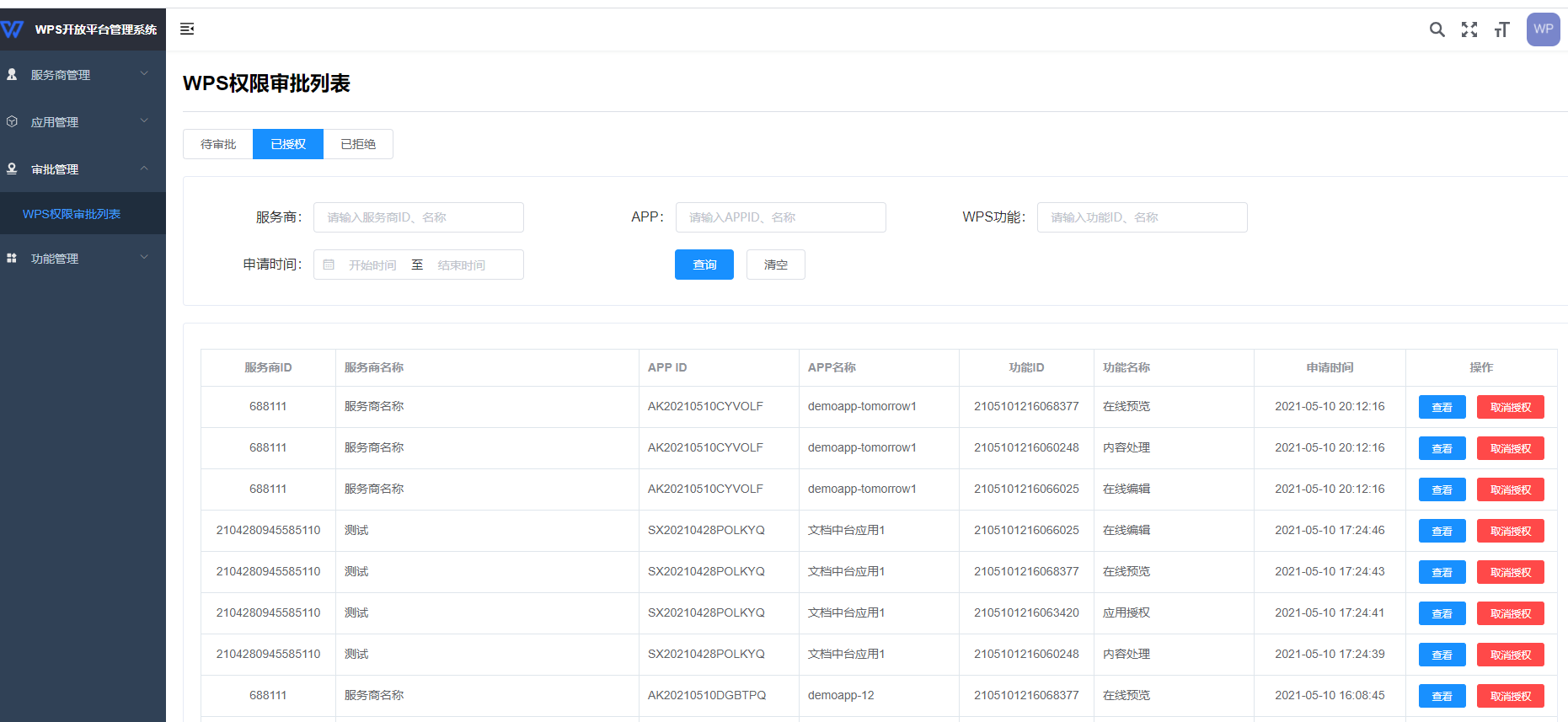


图 8 能力授权

* 二次授权步骤：

进容器，执行命令重新导入证书，例如：

curl -XPOST "http://docs-privilege/v1/licence/import" -T test.lic

返回如下命令则导入成功：

{"code":20000,"msg":"","servertime":"2021-05-28 12:00:14"}

#### 开发文档

开放平台提供应用授权，对针对不同能力申请的调用时，根据场景要求对不同接入应用进行对应的授权处理。



图 9 开发文档

### 平台对接应用管理

#### 获取密钥

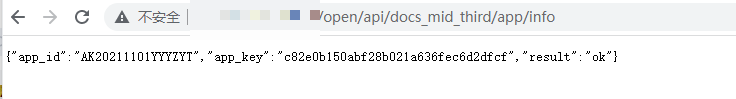
企业开发者在进行对接前，需要获取企业管理员提供的文档中台的服务密钥，开发者获取密钥后才能进行应用创建与应用相关管理接口的调用，获取appid和appkey之后使用wps-3 header签名完成应用创建。下图详细描述管理员密钥的获取步骤：

1. 登录开放平台管理页面：http(https)://{domain}/open
2. 进入页面后调用以下接口，返回管理员aksk：

http(https)://{domain}/open/api/docs\_mid\_third/app/info

返回如下信息：

{"app\_id":"AK20211101YYYZYT","app\_key":"c82e0b150abf28b021a636fec6d2dfcf","result":"ok"}



通过3.2获取管理员aksk后，在调用3.3提供的应用管理能力创建应用时需通过管理员aksk进行接口签名。如下图：

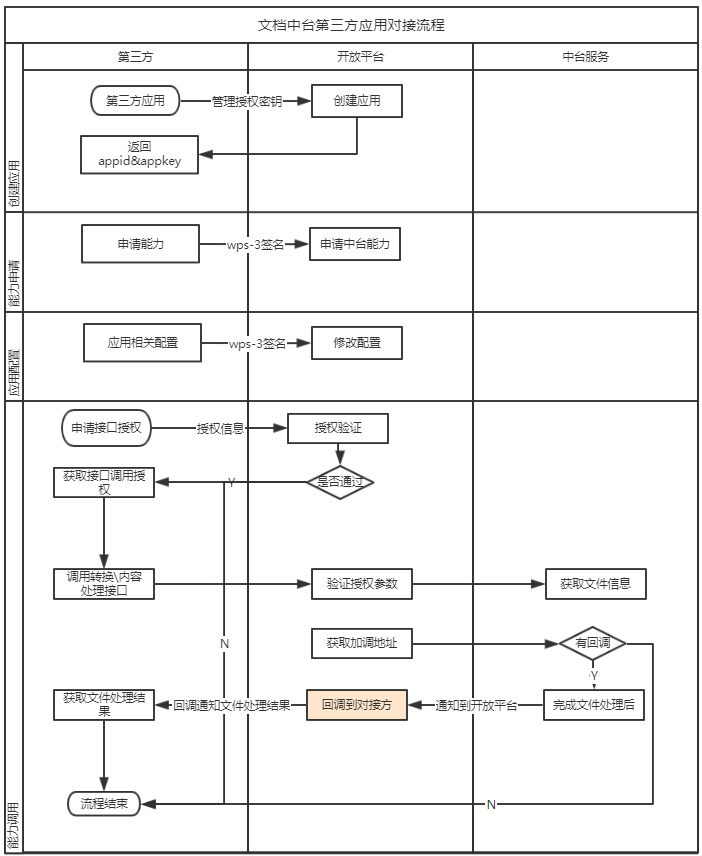


图 10 对接流程

文档中台v5.1.4版本未提供可视化界面进行应用相关的创建设置功能，但开放了对应的应用管理接口，第三方可根据对接平台的使用需求自由定义使用交互。下面主要介绍较为核心的使用场景：应用创建、应用配置设置、获取授权信息、应用授权申请等。

#### 应用注册

通过3.2获取后的appid和appkey，带上密钥调用应用创建接口（/admin/v1/app）。

#### 应用授权

获取应用所能申请到的能力列表（/admin/v1/function/list），完成能力申请（/admin/v1/app/{app\_id:[\w=+]+}/function/{function\_id:\d+}）。

#### 应用参数配置

获取为所申请的能力设置必要的配置参数，特别注意接口必填参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **能力模块** | **配置参数** | **接口** | **必须实现** |
| 在线编辑 | 在线编辑文件信息、权限、版本管理等接口配置 | /admin/v1/app/{app\_id:[\w=+]+}/online\_edit | 部分必须 |
| 在线预览 | 在线编辑文件信息接口配置 | /admin/v1/app/{app\_id:[\w=+]+}/online\_preview | 必须 |
| 文件处理 | 文件处理结果回调地址 | /admin/v1/app/{app\_id:[\w=+]+}/format\_handle | 非必须 |

其中，在线编辑中的接口配置需必填回调域名、获取文件无数据、获取用户信息、上传文件新版本、获取指定版本信息、获取所有版本信息。

### 应用能力实现

完成以上关于应用相关信息设置后，可进行接口开发。

## 能力接入

获取应用对应的appid和appkey后，带上应用签名可进行能力调用。

### 在线编辑

使用线编辑能力时，获取在线编辑地址，在浏览器端打开文档时，文档中台服务会发起几个回调接口：获取文件元数据、获取用户信息、上传文件新版本、获取指定版本信息、获取所有版本信息。对接服务实现以上接口的数据返回才可打开文件进行文档在线和协同编辑。

同时，在实现其它接口如：上下线通知等，文档中台服务会在文档发生对应场景的变化时通知到第三方应用。相关应用可通过回调推送的数据，完成文件的历史版本管理、用户在线情况统计等。

### 在线预览

使用线编辑能力时，获取在线预览地址，在浏览器端打开文档时，文档中台服务会发起几个回调接口：获取文件元数据。对接服务实现接口的数据返回才可打开文件，进行文档在线预览。

在线预览同时提供前端Preview JS SDK，用于业务方通过iframe标签嵌入在线预览页面是能通过此SDK进行Token鉴权和调用打印功能。

在线预览支持程序类文档、图片、PDF\OFD、四大组件等格式。

### 格式处理

格式处理编辑能力支持同步和异步获取文件处理结果，使用异步获取文档处理结果时，需要配置对应的回调地址。

# 接口签名

文档中台所有接口的签名都只需要对uri进行签名，即接口对应的path路径。服务的域名地址不包含在签名范围内。

## 签名描述

接口鉴权需要wps3签名的接口,header中需要携带以下字段：

| **参数** | **参数**  **类型** | **是否**  **必须** | **说明** |
| --- | --- | --- | --- |
| Content-Type | string | required | 目前固定为: application/json |
| Content-Md5 | string | required | HTTP Body中数据的md5值十六进制表达方式 |
| Date | string | required | 取当前时间, 格式: Wed, 23 Jan 2013 06:43:08 GMT |
| X-Auth | string | required | 签名值 |

X-Auth计算方法参考签名如下：

X-Auth："WPS-3:" + AppId + ":" + sha1(AppKey + Content-Md5 + URI + Content-Type + Date)；

Content-Md5: HTTP Body中数据的md5值十六进制表达方式，如果参数不带在body，则算空字符串""的md5值；

URI：不带域名，包含uri和query参数，如"/api\_url?app\_id=aaaa"；

标准sha1算法： 结果为一个40个字符的字符串，16进制小写表示的string。

# SDK&Demo

## SDK使用说明

对接方可根据java sdk完成能力接入，也可以直接对接openapi。

### JAVA SDK

#### 结构

wps-openapi-javasdk.zip压缩包解压后，结构如下

*压缩包*

*--- lib/*

*--- conf/*

*--- apidocs/*

*--- README.MD*

*--- wps-openapi-sdk-xxxx.jar*

* lib中是依赖包，如果是maven项目可以看jar中的pom.xml
* conf中是可供参考的配置文件配置
* apidocs下是javadoc文档
* README.MD记录了详细的使用方法
* wps-openapi-sdk-xxxx.jar提供了访问WPS文档中台api的功能

下载地址：

<https://o.wpseco.cn/kdocs/l/coGfFi6HGSFJ> wps-openapi-javasdk.zip

#### 使用方法

具体使用方法请查看README.MD文档

### Go SDK

#### 结构

*压缩包*

*|--- opensdk/*

*|--- comm*

*|--- cpssdk*

*|--- opensdk*

*|--- go.mod*

*|--- README.MD*

#### 压缩包下载地址

下载方法：文档中台-demo-下载代码

#### 使用方法

具体使用方法请查看README.MD文档

### OPEN SDK

文档中台前端sdk（后文简称OpenSDK）是文档中台向网页开发者提供的网页开发工具包，是整合了WebOffice SDK和原预览SDK而成的，网页开发者可以通过OpenSDK实现页面的预览和编辑、监听事件、获取时间节点的能力，同时可以直接使用高级API来操作文档。

适配版本：

|  |  |
| --- | --- |
| **平台** | **适用版本** |
| x86 | v5.1.7\_20211125 及以后版本 |
| xc | 暂不支持 |

#### 下载

在使用之前，请先下载最新版本的OpenSDK代码。

[open-jssdk-v1.0.0.umd.js-点击下载](https://kdocs.cn/l/sdP3fYLYGh8U)

[open-jssdk-v1.0.0.cjs.js-点击下载](https://kdocs.cn/l/sfCE4wW901Yb)

[open-jssdk-v1.0.0.es.js-点击下载](https://kdocs.cn/l/so3EkjIPX3U3)

index.d.ts-下载：https://plus.wps.cn/view/media/l/sbVckGzn8Qhs

#### 引用

OpenSDK提供了支持非模块化以及AMD、CommonJS、ES6多种模块化规范的包，下面以非模块化方式引用为例，讲解OpenSDK的快速上手。

\*注意：暂不支持在同一个页面中使用该"sdk"加载多个预览编辑链接，如果有此需求，请在该页面创建多个iframe，在每个iframe内部使用"sdk"加载一个预览编辑链接，以此实现window层的隔离

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8" />  <meta  name="viewport"  content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no"  />  <meta http-equiv="X-UA-Compatible" content="ie=edge" />  <title>OpenSDK</title>  </head>  <body>  <div  id="office"  style="display: block; margin-top: 15px; width: 100%; height: 600px"  ></div>  <script src="./open-jssdk-v1.0.0.umd.js"></script>  <script>  console.log("引入后可以开始使用 OpenSDK 了！");  console.log(OpenSDK);  </script>  </body>  </html> |

在这份代码中，我们在HTML代码中引用了OpenSDK：

|  |
| --- |
| <script src="./open-jssdk-v1.0.0.umd.js"></script> |

其它引用方式：

OpenSDK提供支持非模块化以及AMD、CommonJS、ES6多种模块化的包，以下是不同版本包对应的模块化规范

|  |  |  |
| --- | --- | --- |
| **文件名** | **模块化规范** | **说明** |
| open-jssdk-v1.0.0.umd.js | UMD | 兼容非模块化、AMD、CommonJS |
| open-jssdk-v1.0.0.cjs.js | CommonJS | CommonJS规范 |
| open-jssdk-v1.0.0.es.js | ES6 | ES6模块化规范 |

1、非模块化

|  |
| --- |
| <script src="open-jssdk-v1.0.0.umd.js"></script> |

2、CommonJS规范

|  |
| --- |
| let OpenSDK = require('./open-jssdk-v1.0.0.cjs.js') |

3、AMD规范

|  |
| --- |
| define(["./open-jssdk-v1.0.0.umd.js"],function(OpenSDK){  //do something...  }) |

4、ES6模块化规范

|  |
| --- |
| import OpenSDK from './open-jssdk-v1.0.0.es.js' |

#### 初始化

引用之后，就可以通过初始化配置，让OpenSDK在配置的挂载节点下创建一个iframe，并自动初始化相关数据和事件

挂载节点指OpenSDK插入iframe时挂载的节点

|  |
| --- |
| window.onload = async function () {  const jssdk = await OpenSDK.config({  url: "在线文档预览地址",  mount: document.querySelector("#office"),  setToken: {  token: "your token", //根据自身的业务需求，通过异步请求或者模板输出的方式，取得token  timeout: 30 \* 60 \* 1000,  },  });  }; |

由于 iframe 限制，所以需要给挂载的节点指定具体宽高

#### 鉴权设置

使用OpenSDK传递token进行鉴权与初始化操作基本一致，只是在“在线文档预览地址的url”参数中设置 \_w\_tokentype=1

|  |
| --- |
| window.onload = async function () {  const jssdk = await OpenSDK.config({  url: "在线文档预览地址" + "&\_w\_tokentype=1",  mount: document.querySelector("#office"),  setToken: {  token: "your token", //根据自身的业务需求，通过异步请求或者模板输出的方式，取得token  timeout: 30 \* 60 \* 1000,  },  //获取 token 函数；可选  refreshToken: () => {  //自身业务处理...  //可以返回Promise或者 return {token,timeout}  return {  token: "your token",  timeout: 30 \* 60 \* 1000,  };  },  });  }; |

#### 监听事件

事件埋点

|  |
| --- |
| jssdk.on("opendocStage", (e) => {  console.log(e);  }); |

#### SDK实例完毕

|  |
| --- |
| jssdk.ready().then(function () {  //do something  }); |

#### 销毁SDK实例

|  |
| --- |
| jssdk.destory(); |

#### 打印功能

调用页面的打印功能，调用打印之前先判断jssdk是否实例化完成

|  |
| --- |
| jssdk.print(); |

#### 获取WebOffice子类SDK实例

|  |
| --- |
| const webofficeSdk = jssdk.getWebofficeInstance(); |

##### WebOfficeSDK使用场景：

1. 所有的编辑页面
2. 预览页面：  
   文字（word）：doc、docx、wps等后缀文件;  
   表格（Excel）：xls、xlsx等后缀文件；  
   演示（PPT）：ppt、pptx等后缀文件；  
   PDF ：pdf等后缀文件；

##### WebOffice sdk使用示例代码

[WebOfficeSDK 文档地址](https://wwo.wps.cn/docs/front-end/" \t "_blank)：https://wwo.wps.cn/docs/front-end/

|  |
| --- |
| //获取wboffice实例  const webofficeSdk = jssdk.getWebofficeInstance();  await webofficeSdk.ready();  const app = webofficeSdk.Application;  //设置字体颜色  app.ActiveDocument.Content.Font.ColorIndex = 2; |

#### 版本号

本 jssdk 为文档中台面向网页开发者提供的整合所有格式服务的开发工具包。

jssdk 的版本遵循 X.Y.Z 格式，分别代表：

· X : 主版本号。当 API 的兼容性变化时，X 会变化

· Y : 次版本号。当增加功能时，Y 会变化

· Z : 修订号。当 BUG 修复时，Z 会变化

本 jssdk 与文档中台部署版本绑定使用，jssdk 版本向上兼容中台部署版本。

### Preview JS SDK

|  |  |
| --- | --- |
| **平台** | **适用版本** |
| x86 | v5.1.7\_2021028 及之前版本 |
| xc | v5.1.7 |

预览场景使用的JSSDK

注：（参见5.1.6版本文档中台使用说明）

release\_v5.1.6\_0630版本首次发布前端JS-SDK-v1.0.0

bugfix\_release\_open\_v5.1.6\_20210716版本更新前端JS-SDK-v1.1.0

#### 准备

JS SDK文件下载地址为<https://o.wpseco.cn/kdocs/l/cp3gh1FMsWlb>

preview-jssdk-v1.1.0.zip ，请自行下载后放入项目中作为资源加载。

#### Token鉴权

通过iframe标签嵌入在线预览页面时，可以通过 JS SDK传递 Token 完成鉴权。

设置 token 分下面几个步骤：

1. 在代码中引用js sdk依赖
2. 在页面中创建一个 iframe 标签用来加载预览链接（注意：不要手动设置这个 iframe 的 src 属性）
3. 通过初始化配置，js-sdk 会在挂载节点 iframe 中加载预览链接，并自动初始化相关数据和事件（Token 鉴权功能必须通过 JSSDK方式加载预览链接）
4. 在线预览地址必须在 url 参数中添加 \_w\_tokentype=1（注意：不需要token鉴权的预览链接不能添加该参数，会导致预览失败）
5. 通过 setToken 主动设置 token
6. 通过在 config 中配置 refreshToken 被动刷新 token（注意：由于 js-sdk 会提前 5 分钟调用该方法，所以建议设置的 timeout 为 10 分钟以上（10 \* 60 \* 1000 以上，单位为毫秒），避免刷新过快的问题）

示例代码：

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width,initial-scale=1.0,maximum-scale=1.0,user-scalable=no">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>JSSDK</title>  </head>  <body>  <iframe id="test" frameborder="0" width="100%" height="600px"></iframe>  <script src="./preview-jssdk-v1.1.0.umd.js"></script>  <script>  console.log('引入后可以开始使用 jssdk 了！');  console.log(PreviewSDK);  // 获取 token 函数  const refreshToken = () => {  // 自身业务处理...  // 可以返回 Promise 或者 return { token, timeout }  return Promise.resolve({  token: 'yourToken', // 必需：你需要设置的 toekn  timeout: 10 \* 60 \* 1000, // 必需：token 超时时间，以 10 分钟示例  });  };  const jssdk = PreviewSDK.config({  url: url, // 在线预览链接，需要添加 \_w\_tokentype=1 参数  mount: document.querySelector('#test'), // 挂载 iframe 节点  refreshToken: refreshToken // 配置超时获取 token 函数  })  // 设置 token  jssdk.setToken({  token: 'yourToken', // 根据自身的业务需求，通过异步请求或者模板输出的方式，取得 token  timeout: 10 \* 60 \* 1000, // token 超时时间，可配合 refreshToken 配置函数使用，在超时前自动调用 refreshToken 重新刷新 token  });  </script>  </body>  </html> |

#### **JS 打印接口**

用户使用 iframe 加载在线预览页面后，可以使用该 api 调用预览页面内置的打印功能。

通过 iframeWindow.postMessage('wpsPreviewPrint', '\*') 调用打印接口

示例代码：

|  |
| --- |
| <html>  ...  <iframe src="url" id="test"></iframe>  <script>  window.frames['test'].contentWindow.postMessage('wpsPreviewPrint', '\*')  </script>  ...  </html> |

#### SDK设置回调获取weboffice预览地址

用户使用文档中台预览时想使用weboffice的js sdk功能，但是weboffice页面是被嵌入在文档中台预览页面的，用户无法获取到weboffice预览链接，也就使用不了weboffice的sdk功能，所以现在preview sdk提供了获取weboffice预览链接的方法。

preview sdk提供了setOfficePreviewCallback(callbackFunction: (webofficeUrl: string, err: error) => void)方法，该方法的参数callbackFunction是获取weboffice预览地址的回调函数，通过该回调函数的参数 webofficeUrl，用户可以获取到weboffice预览地址并进行weboffice sdk初始化调用api。

注意：本api需要同时引入weboffice jssdk使用，weboffice jssdk具体api方法请看目录5.1.4

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <script src="./preview-jssdk-v1.1.0.umd.js"></script>  <script src="./web-office-sdk-v1.1.14.umd.js"></script>  </head>  <body>  <iframe src="" id="test" width="100%" height="600px"></iframe>  <div id="office"></div>  <script type="text/javascript">  const previewSdk = PreviewSDK.config({  url: url, // 文档中台预览地址  mount: document.querySelector('#test'),  })  previewSdk.setOfficePreviewCallback(function(officeUrl, err) {  if (!err) {  console.log('从回调函数获取到weboffice预览地址', officeUrl)  previewSdk.destroy()  loadWebOffice(officeUrl)  }  })  async function loadWebOffice(officeUrl) {  // api文档https://wwo.wps.cn/docs/front-end/introduction/  const instance = WebOfficeSDK.config({  url: officeUrl,  mount: document.querySelector('#office'),  })  await instance.ready();  const app = instance.Application;  // Slide 设置对象  const SlideShowSettings = await app.ActivePresentation.SlideShowSettings;  // 进入幻灯片播放模式  await SlideShowSettings.Run();  // 开启自动播放  await SlideShowSettings.StartAutoPlay();  }  </script>  </body>  </html> |

### Weboffice JS SDK

编辑场景使用的js sdk

应用细节：https://wwo.wps.cn/docs/update-log/jssdk/

注：原则上所有版本的Weboffice JS SDK都可以使用。具体API接口对JSSDK、WebOffice版本的要求以WebOffice知识库为准。

## DEMO使用说明

见文档中台demo中：《WPS文档中台Demo用户使用说明书》