

JSAPI快速上手说明

JSAPI快速上手说明

Demo怎么能快速跑起来

WPS加载项的运行原理

WPS加载项技术架构

OA助手——WPS加载项的一种应用

WPS加载项正常运行时WPS客户端的必要配置

WPS加载项（本Demo）代码结构说明

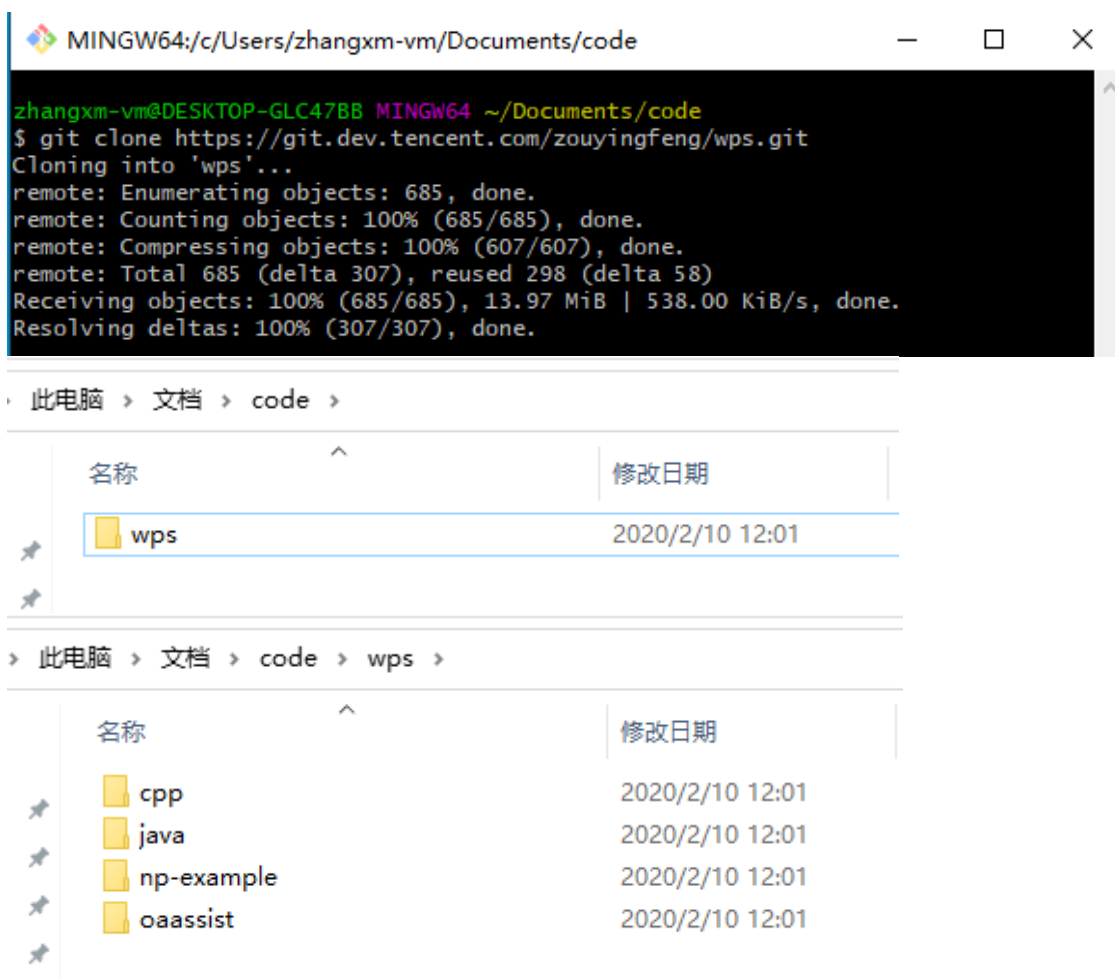
服务端代码结构说明

WPS加载项代码结构说明

Demo怎么能快速跑起来

1. 下载[git](#)和[nodejs](#)，执行默认安装即可；
2. 获取代码到本地，新建一个文件夹，并在此文件夹下通过git客户端执行clone命令，如下图所示：

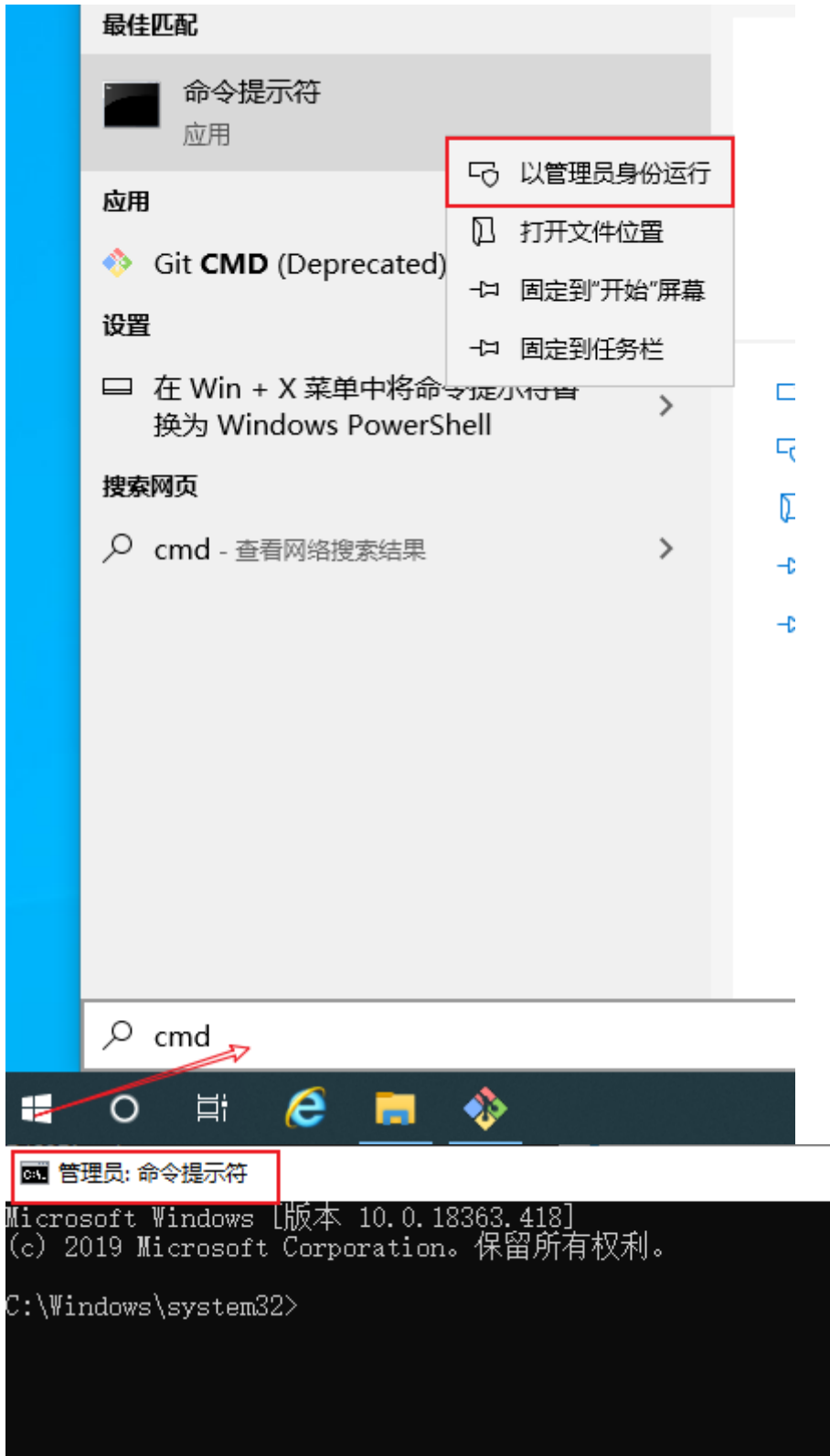
```
1 | git clone https://git.dev.tencent.com/zouyingfeng/wps.git
```



3. JSAPI的Demo所在文件夹是 `oaassist`，进入后的代码结构如下图所示：

名称	修改日期
EtOAAssist	2020/2/10 12:01
server	2020/2/10 12:01
WppOAAssist	2020/2/10 12:01
WpsOAAssist	2020/2/10 12:01
demo.html	2020/2/10 12:01
README.md	2020/2/10 12:01

4. 以管理员的身份启动命令行工具，操作步骤如下图：



切换到JSAPI的Demo文件夹的 `server` 目录下，执行如下命令（文档例子中的目录是：

C:\Users\zhangxm-vm\Documents\code\wps\oaassist\server :

```
1 | cd C:\Users\zhangxm-vm\Documents\code\wps\oaassist\server
```

```
管理员: 命令提示符
Microsoft Windows [版本 10.0.18363.418]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Windows\system32>cd C:\Users\zhangxm-vm\Documents\code\wps\oaassist\server

C:\Users\zhangxm-vm\Documents\code\wps\oaassist\server>dir
驱动器 C 中的卷没有标签。
卷的序列号是 5E94-9455

C:\Users\zhangxm-vm\Documents\code\wps\oaassist\server 的目录
2020/02/10 12:01 <DIR>      .
2020/02/10 12:01 <DIR>      ..
2020/02/10 12:01          41 .gitignore
2020/02/10 12:01        466 package.json
2020/02/10 12:01       6,711 StartupServer.js
2020/02/10 12:01 <DIR>      wwwroot
                3 个文件          7,218 字节
                3 个目录 43,473,178,624 可用字节

C:\Users\zhangxm-vm\Documents\code\wps\oaassist\server>_
```

5. 执行 npm install 命令，初始化Demo的模拟服务端：

```
管理员: 命令提示符
Microsoft Windows [版本 10.0.18363.418]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Windows\system32>cd c:\Users\zhangxm-vm\Documents\code\wps\oaassist\server

c:\Users\zhangxm-vm\Documents\code\wps\oaassist\server>npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN oaassist_demo@1.0.0 No repository field.

added 64 packages from 43 contributors and audited 143 packages in 6.886s
found 0 vulnerabilities

c:\Users\zhangxm-vm\Documents\code\wps\oaassist\server>
```

6. 执行 node StartupServer.js 启动服务端：

```
管理员: 命令提示符 - node StartupServer.js
Microsoft Windows [版本 10.0.18363.418]
(c) 2019 Microsoft Corporation. 保留所有权利。

C:\Windows\system32>cd c:\Users\zhangxm-vm\Documents\code\wps\oaassist\server

c:\Users\zhangxm-vm\Documents\code\wps\oaassist\server>npm install
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN oaassist_demo@1.0.0 No repository field.

added 64 packages from 43 contributors and audited 143 packages in 6.886s
found 0 vulnerabilities

c:\Users\zhangxm-vm\Documents\code\wps\oaassist\server>node StartupServer.js
2020年2月10日 14:6:50 启动本地web服务(http://127.0.0.1:3888)成功!
```

7. 打开 oaassist 目录下的 demo.html 文件，开始基础环境检测并体验Demo：

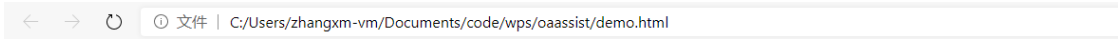
打开 demo.html 文件后，默认开始检测服务端运行情况，之后检测本地WPS客户端的安装情况，如下图则表示服务端启动正常，本地未安装WPS客户端。

OA助手演示Demo



请联系项目经理，获取WPS2019企业版。

安装之后再次运行出现如下结果，表示Demo的演示环境构建完成。



OA助手演示Demo

8. 点击开始体验后，跳转到的界面即是模拟的业务系统操作页面，此页面的功能，可以模拟通过前端页面调起本地WPS客户端，在WPS客户端完成文档编辑后，通过WPS加载项将数据与服务端同步的使用场景。

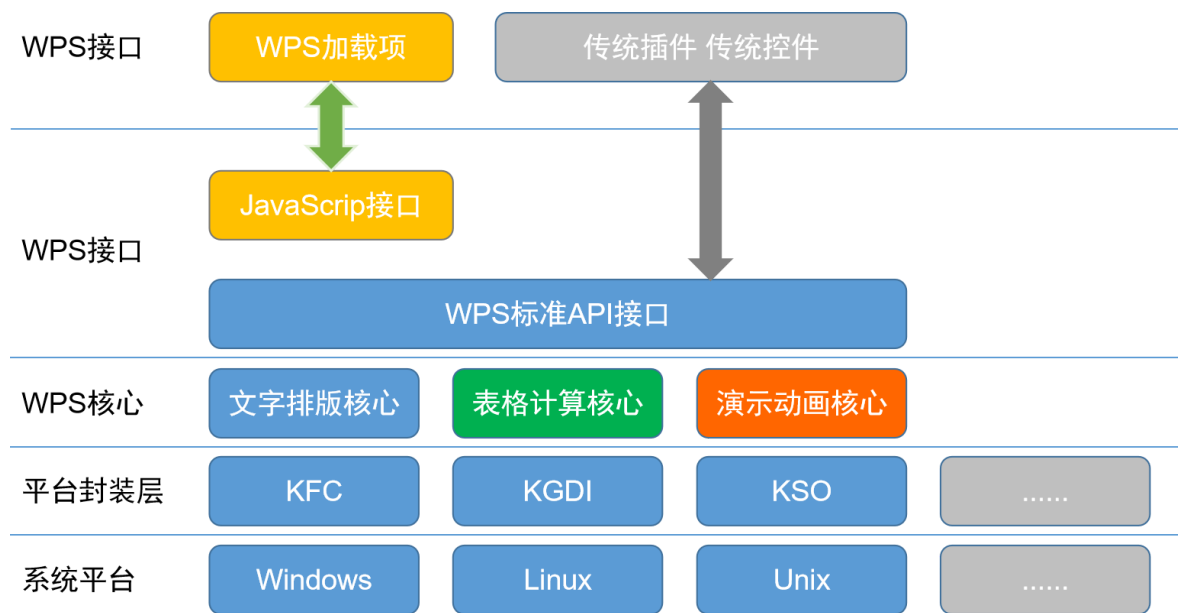


WPS加载项的运行原理

WPS加载项技术架构

- WPS加载项通过WPS JavaScript接口使用WPS标准API接口
- WPS JavaScript接口是基于WPS标准API接口，可有效降低开发人员学习成本

- WPS标准API接口全面兼容Office接口

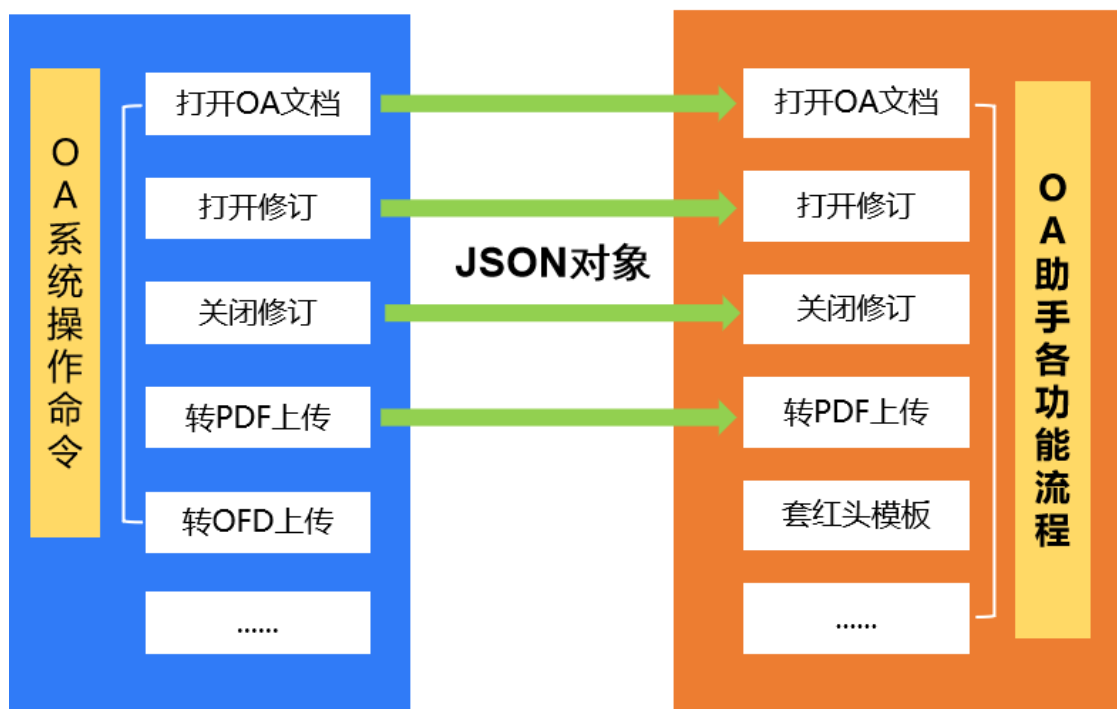


OA助手——WPS加载项的一种应用

- OA助手是OA系统与WPS客户端进行协同的桥梁
- 本Demo提供的样例代码中，实现了OA系统中与Office客户端交互的常用业务逻辑处理流程，开发者可基于此自由扩展
- OA助手不需要依赖浏览器控件，具有跨平台、跨浏览器、可扩展性强、系统健壮和部署升级便利的特点
- OA助手实现的OA系统网页和WPS客户端交互原理是利用自定义URL协议，如下图：

OA系统网页

WPS客户端



- 所有浏览器都支持自定义URL协议
- OA系统与OA助手通过自定义URL协议交互
- OA系统页面与WPS客户端交换JSON数据
- JSON对象的内容由OA系统与OA助手协商

- WPS客户端根据JSON数据加载OA助手相应功能

WPS加载项正常运行时WPS客户端的必要配置

1. oem.ini文件

- 这是WPS客户端的配置文件，所在位置在WPS客户端安装目录的 office6\cfgs 下，以Windows客户端举例，WPS客户端安装在 C:\Program Files (x86)\kingsoft\WPS office\11.8.2.8748，则 oem.ini 的路径即是 C:\Program Files (x86)\kingsoft\WPS office\11.8.2.8748\office6\cfgs\oem.ini
- 此文件对WPS加载项运行必须设置的配置项如下：

```
1 [Support]
2 JsApiPlugin = true
3 JsApiShowWebDebugger = true
4
5 [Server]
6 JSPluginsServer = http://127.0.0.1:3888/jsplugins.xml
```

- JsApiPlugin，设置为 true 表示启用WPS加载项
- JsApiShowWebDebugger，设置为 true 表示在WPS加载项的工具栏上显示 打开JS调试器 的按钮，如下图。推荐在开发阶段此配置设置为true，正式生产环境中设置为false



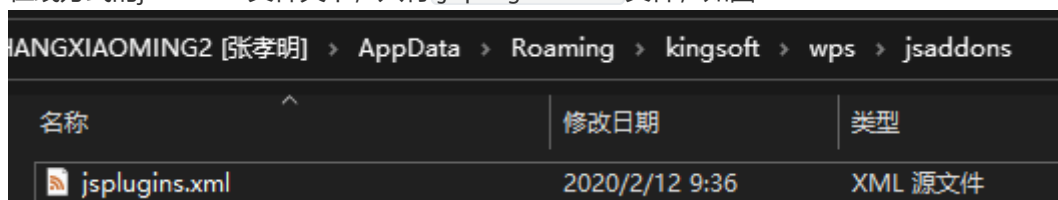
- JSPluginsServer，OA系统通过自定义浏览器协议调起WPS客户端时，WPS客户端会通过此配置项获取WPS加载项的配置文件

2. jsplugins.xml文件

- 这是WPS加载项在客户端的配置文件，以Windows环境距离，位置在 %appdata%\kingsoft\wps\jsaddons 下
- WPS加载项运行有两种方式，在线和离线。在线方式表示自定义功能区、任务窗格和Web对话框的数据文件存储在服务端，WPS客户端启动后通过网络请求获取；离线方式则表示这些数据会在第一次加载时将数据拉取到本地，之后每一次启动将对比本地和xml文件中定义版本号，不一致将进行更新。
- 在线方式配置

```
1 <jsplugins>
2   <jspluginonline name="EtOAAssist" type="et"
3     url="http://127.0.0.1:3888/plugin/et"/>
4   <jspluginonline name="wpsOAAssist" type="wps"
5     url="http://127.0.0.1:3888/plugin/wps"/>
6   <jspluginonline name="wppOAAssist" type="wpp"
7     url="http://127.0.0.1:3888/plugin/wpp"/>
8 </jsplugins>
```

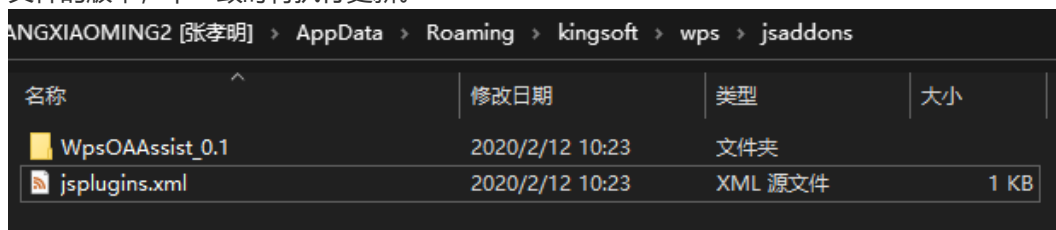
此配置WPS客户端每次启动时，通过 oem.ini 中 JSPluginsServer 配置项进行更新。
在线方式的jsaddons文件夹下，只有 jsplugins.xml 文件，如图：



- 离线方式配置

```
1 <jsplugins>
2   <jsplugin name="EtOAAssist" type="wps"
  url="http://127.0.0.1:3888/plugins/v0.1/EtOAAssist.7z" version="0.1"
  />
3   <jsplugin name="WpsOAAssist" type="wps"
  url="http://127.0.0.1:3888/plugins/v0.1/WpsOAAssist.7z"
  version="0.1" />
4   <jsplugin name="WppOAAssist" type="wps"
  url="http://127.0.0.1:3888/plugins/v0.1/WppOAAssist.7z"
  version="0.1" />
5 </jsplugins>
```

此配置WPS客户端每次启动时，通过 oem.ini 中 JSPluginsServer 配置项进行更新。
比在线配置多了 version 配置项，WPS客户端每次启动后，将对比此配置中 version 和本地文件的版本，不一致时将执行更新。



如上图，WPS加载项的离线文件夹是 WpsOAAssist_0.1，此规则为xml文件中，
name_version，如果破坏此规则，会造成WPS加载项不可正常加载。

WPS加载项（本Demo）代码结构说明

服务端代码结构说明

- 此服务端模拟了OA业务系统与Office的交互场景，采用的Node.js做得实现，开者亦可选择其他语言实现相应的功能。
- StartupServer.js: 此文件是Node服务端入口文件，实现的主要功能包括
 - 以下请求，是支持WPS加载项运行时的必须实现的：
 - /plugin/et、/plugin/wps、/plugin/wpp: 在线方式下，获取加载项内容
 - /Upload: 保存文件到服务端，需按照表单数据格式进行解析
 - /Download: 从服务端获取指定文件
 - 以下请求，非必须实现：
 - /WpsSetupTest: WPS客户端环境检测
- jsplugins.xml: 此文件是服务端的WPS加载项的配置文件，WPS客户端的 oemini 的 JSPluginsServer 需指向可获取此文件，每次WPS客户端启动都会获取此件来更新本地文件
- resource/wps.html(et.html,wpp.html): 此三个文件是OA助手应用场景实网页
- resource/js/wps.js(et.js,wpp.js): 此三个文件是OA助手应用场景实例网对应的js代码，**开发者可将此文件的代码引入到自己的项目代码**
 - WpsStartup.Startup 方法中，**插件名称(name)**必须与 jsplugins.xml 文件中的 name 保持一致，否则将无法正常加载WPS加载项
 - WpsStartup.Startup 方法中，**加载项方法入口(func)**必须与TODO:一致
- resource/js/wpsstartup.js: 此文件封装了前端网页启动WPS客户端的自定URL协议的公用方法，**开发者可将此文件引入到自己的项目工程中**

WPS加载项代码结构说明

以WPsOAAssit为例说明

- ribbon.xml：此文件是自定义选项卡的配置文件，按照[MSO标准](#)的规则添加tabs、tab、group、box、button这些常用的元素