

AppEmit 中间件 说明书

程序名称	AppEmit.exe	
程序版本	v0.8.3	
说明书版本	v0.8.3	
网址	http://www.appemit.com	
Email	appemit@appemit.com	
修改时间	2020 年 7 月 6 日	
修订	v0.8 支持 js 调用 office 插件 v0.7 增加 office 插件 v0.6 增加 office dso 插件 v0.5 增加浮动窗口 v0.4 增加注册功能 v0.3 增加重启 v0.2 增加 web v0.1 flash	

1 概述

AppEmit 是应用程序（尤其是浏览器）与本地程序间互相通信的易扩展的轻量级中间件。

AppEmit is an extensible lightweight middleware for communication between applications (especially browsers) and local programs.

主要采用了 HTML5 国际标准的 Web Socket 进行通话，默认为异步，JSON 格式传递参数。

➤ 主要功能：

- 1) 在所有浏览器播放含有 flash 的网页或 Flash 文件，包括 swf 交互动画、flv 影视等
- 2) 在所有浏览器打开、操作本地文件，比如阅读 PDF；创建、阅读、编辑 Office 文件，且支持 JavaScript 代码操作
- 3) 开发本地硬件 DLL 驱动模块的封装插件，实现在网页中操作控制本地的读卡器、打印机、扫描仪、高拍仪、U 盾等各种硬件设备
- 4) 各个应用程序之间通信，比如聊天
- 5) 在 Chrome 里嵌入 IE 内核网页，保护源码，可以不修改原有的 ActiveX 读取 html，同时支持开源内核 wke 和 blink

➤ 解决问题

- 1) 国际市场份额 68% 以上的 chrome 浏览器（数据来源 Netmarketshare；国内 25% 以上）在 2020 年 12 月后不再支持 flash，而微软的 edge 也不支持 ActiveX。
- 2) 客户习惯使用浏览器来处理各种业务。
- 3) 游戏商、银行、医院、电力、硬件等企业客户使用 dll、ActiveX、flash 等文件的场景需要。

➤ 相关链接

程序名称 AppEmit.exe

网址 <http://www.appemit.com>

测试地址 <http://www.appemit.com/demo/index.html>

Github <https://github.com/appemit/appemit>

Email 联系 appemit(at)appemit.com

推荐内容分发下载地址：

<https://cdn.jsdelivr.net/gh/appemit/appemit/dist/AppEmit.zip>

github 下载地址

<https://raw.githubusercontent.com/appemit/appemit/master/dist/AppEmit.zip>

1.1 使用条件

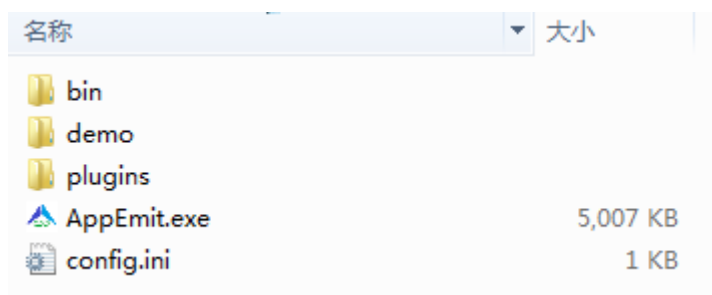
Windows 系统，支持 XP 以上。

软件/插件	功能/方式	依赖条件	说明
AppEmit		Windows 系统 XP 以上	32 位，可以运行在 x64、x86 上
flash	调用本地 flash ActiveX 控件，打开 swf 文件	本地安装 flash ActiveX 控件	
	打开网络中含有 swf 的网页 打开网络中的文件 swf	自动安装插件	或者下载完全插件版本可局域网使用。后续同。
IE	打开网络中含有 swf 的网页 采用不同内核 IE\webkit 打	-	

	开网络或者本地网页		
PDF	阅读网络或者本地 PDF 文件	-	
Office	打开、编辑 Office 文档	本地安装 Office	
媒体文件	打开网络中的文件 视频格式：mp4、flv、m3u8、rtmp 视频编码：H.264 音频编码：AAC、MP3 音频格式：MP3 等文件	-	

1.2 用法

下载免安装程序 AppEmit（不含插件小于 6M），运行 AppEmit.exe 即可。设置了开机自启动，应避免被杀毒软件关闭。



同时只能开启一个 AppEmit.exe 进程。

- 直接运行，如果本机已经运行了 AppEmit.exe，则不做处理。
- 如果本机已经有程序 AppEmit.exe 运行，右键以管理员方式运行，则关闭老进程，开启新的进程。

1.3 技术实现

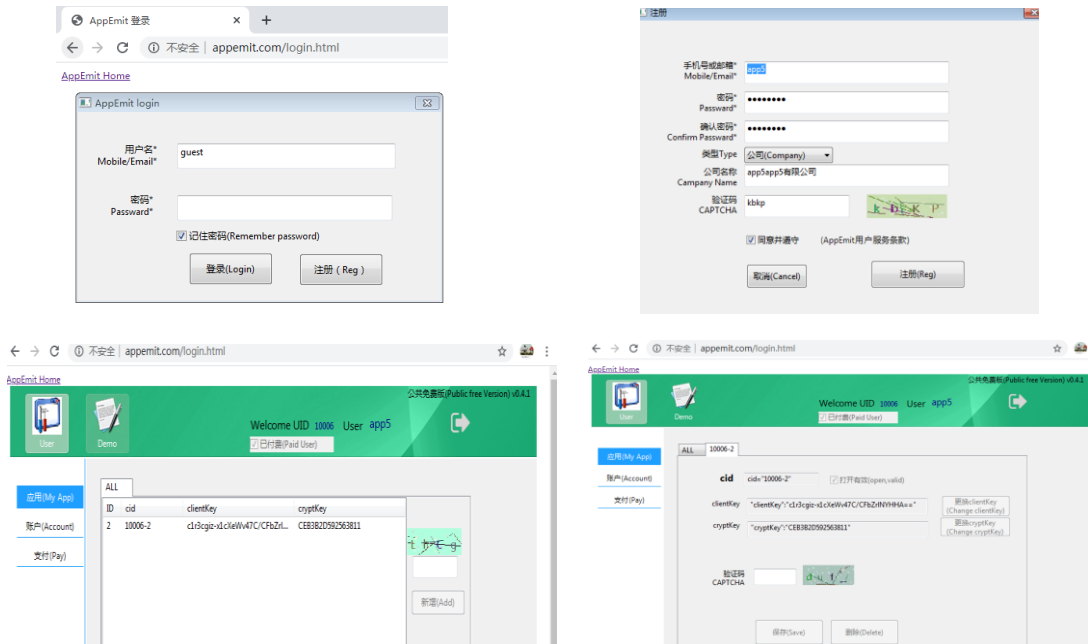
1.3.1 技术依赖

采用开源或公共控件，安全可靠。

- Web Socket 采用开源控件 HP Socket (<https://github.com/ldcsaa/HP-Socket>)，支持 ssl。
Dll 文件开放了 C 接口，可以在此基础二次开发控件，主要调用
 - HP Socket4C_U.dll
 - HP Socket4C-SSL_U.dll
- IE 内核调用 flash Active 控件
- 非 IE 内核调用 flash NPAPI 控件
- webkit 内核采用开源软件 wke 控件
- blink 内核采用开源软件 miniblink 控件

1.3.2 步骤

1. 网页注册后获得设置 cid, clientKey，获得连接授权。或者使用临时账户 cid=00000-1 测试。



一个账户 cid 目前最多 15 个, 设置 cid 后不要任意修改, 更新数据有 1-2 小时延迟。

2. 连接 Appemit 服务

`initAppEmit("ws://localhost:80/appemit?cid=00000-1&sid=1&flag=1")`

3. 设置 clientKey 授权, (clientKey 为私有, 发布后需要保密混淆加密 js) 初始化数据以及授权等

```
var init_AE={
    "clientKey":"temp-00000000000",
    "browser":ThisBrowser,
    "wsUrl":wsUrl,
    // "sid":"1",
    "gid":["1,2"],
}
```

`EmitReq_PaOP(init_AE);`

4. 发送命令

`startAppEmit('{"emit":"hardWare","Obj":"pc"}')`

1.3.3 demo

在 demo 下主要是 html 的举例,

- 包括获取 pc 信息, 实现通话的 index.html
- 以及播放 flash 的 AppEmbed.html

1.4 联系

邮件: appemit@appemit.com

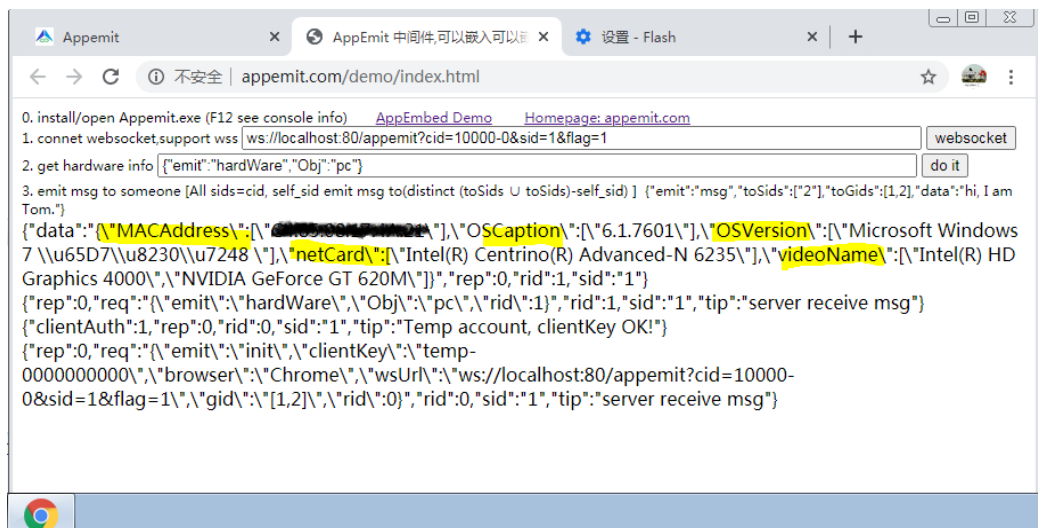
2 插件场景用法

2.1 获取客户端信息

使用浏览器打开 demo 下的 index.html。授权连接后, 发送获取 PC 信息命令。

`initAppEmit("ws://localhost:80/appemit?cid=00000-1&sid=1&flag=1")`

`startAppEmit('{"emit":"hardWare","Obj":"pc"}')`



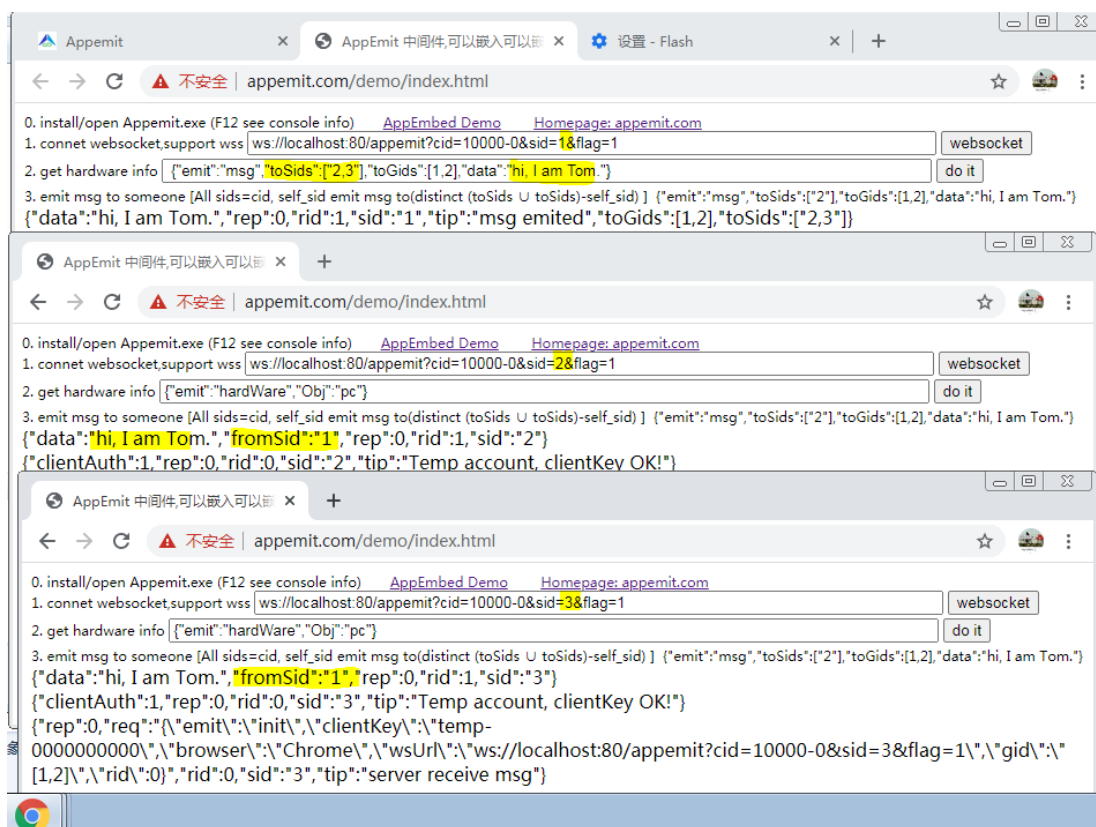
2.2 不同客户端通信

打开 demo 下的 index.html, 模拟不同 sid 打开浏览器。

连接 Appemit 授权后, 在 sid=1 下发送命令。

```
{"emit": "msg", "toSids": [2, 3], "toGids": [1, 2], "data": "hi, I am Tom."}
```

在客户 cid 全集下, 通过唯一的 sid 对话, 可以一对一, 或者一对多通话。



图为 1 对 2 和 3 通话。

另外还可以设置不同群 gid, 一个 sid 可以加入不同的 gid。

发送消息时, 在 cid 全集下, 所有的 toSids 和 toGids 取对应的 sid 交集剔除, 并排除自身。

2.3 插件部署

在发布时, 也可以把此 AppEmit 插件目录文件和 AppEmit.exe 一起打包后发布, 不用单独分发。

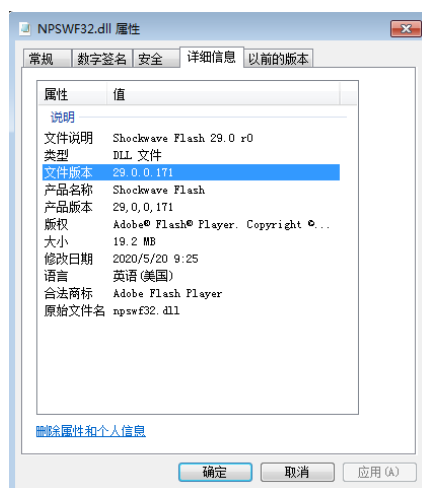
名称	路径	作用
----	----	----

blink	/plugins/lib/web/blink/.dll/node.dll	浏览器内核
	/plugins/lib/web/blink/.dll/wkex.dll	
flash_NP	/plugins/NPSWF32.dll"	NP 版本 flash
	/plugins/plugin.vch	
aliplayer	/plugins/common/aliplayer	Aliplayer 播放器支持 视频格式: mp4、flv、m3u8、rtmp 视频编码: H.264 音频编码: AAC、MP3 音频格式: MP3
PDF	/plugins/common/pdfjs	浏览 PDF 文件

2.4 Flash

两种方法，主要四种形式实现场景

- 1、使用客户端本地安装的 Flash Player ActiveX 控件，要是客户端没有，需要自行下载。下载地址: <http://www.adobe.com/go/getflashplayer>
- 2、使用 Appemit 程序自带的插件 plugins/NPSWF32.dll



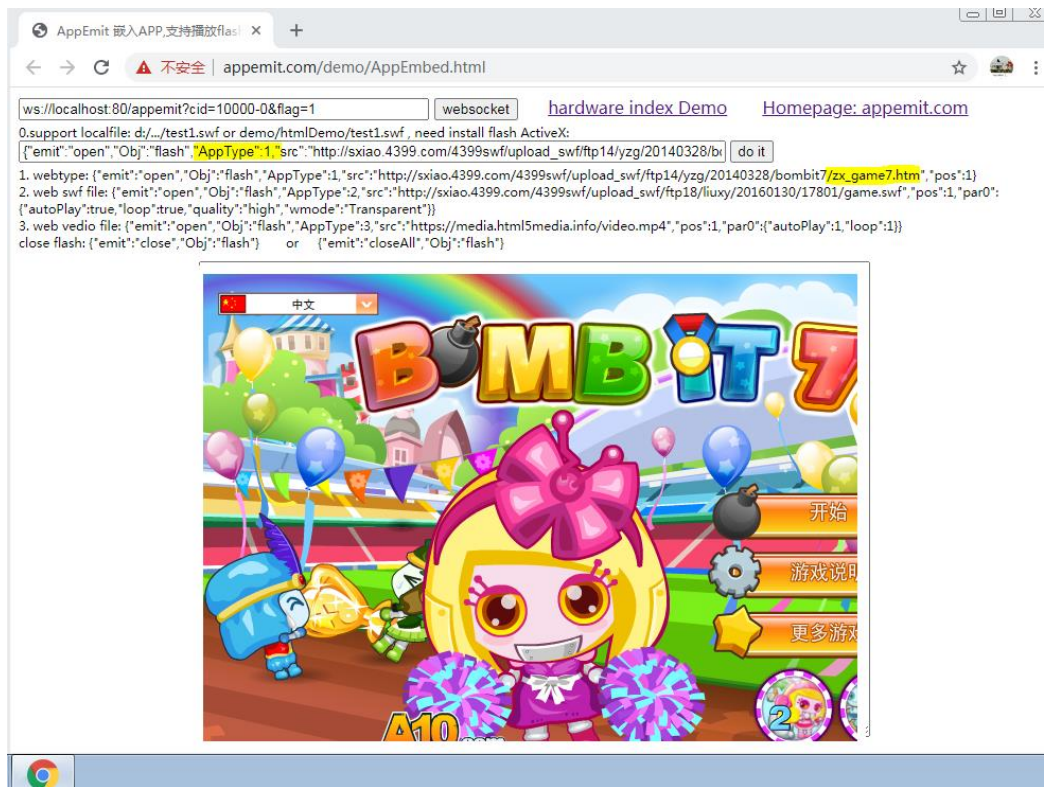
2.4.1 NPAPI-嵌入 web

能打开常用网页，目前的插件不支持 html5 的媒体特性。如有需要，可以使用 node 或者 electron 插件。

使用 webkit 内核使用 Appemit 程序自带的插件 NPSWF32.dll,能打开嵌有 flash 的网页，不默认具有右键菜单。

连接授权后，发送命令 "AppType":1 的形式。

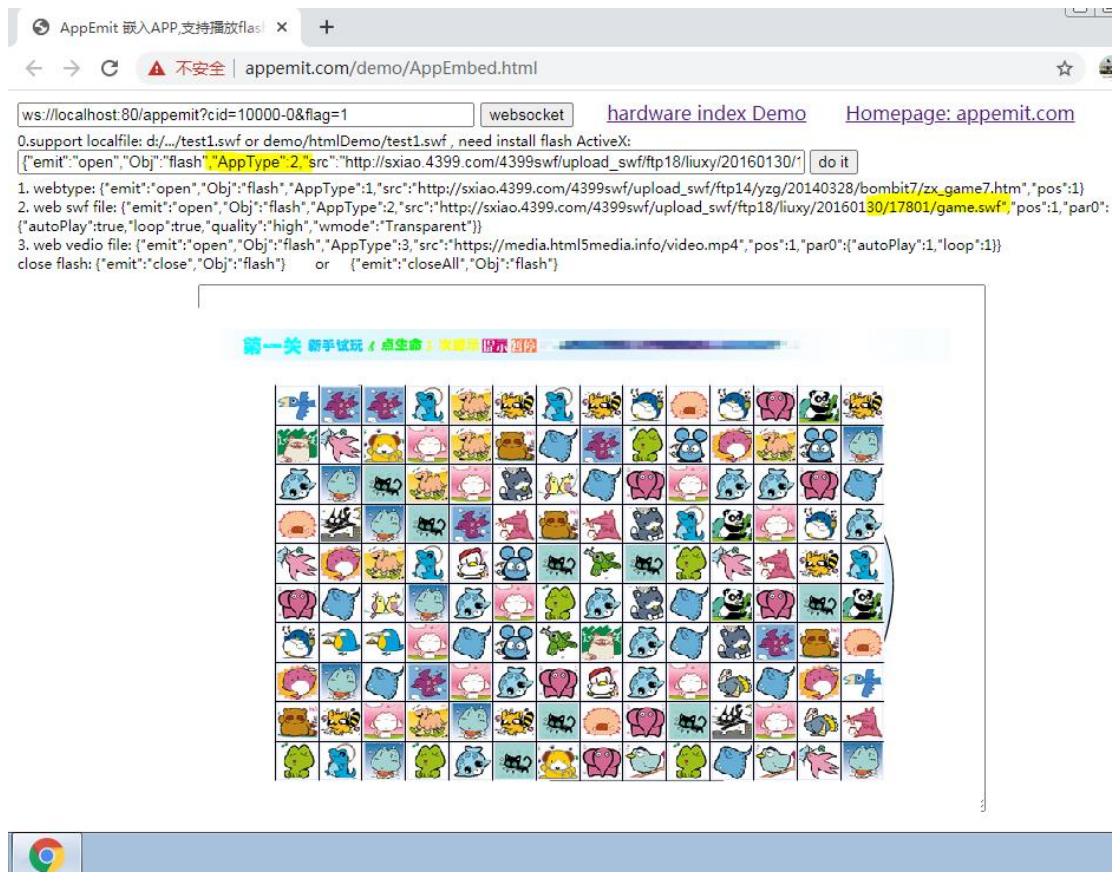
```
{"emit":"open","Obj":"flash","AppType":1,"src":"http://sxiao.4399.com/4399swf/upload_swf/ftp14/yzg/20140328/bombit7/zx_game7.htm","pos":1}
```

2. 4. 2 NPAPI-网络 flash 文件

使用 Appemit 程序自带的插件 NPSWF32.dll， 打开网络 flash 文件。
连接授权后， 发送命令 "AppType":2 的形式。

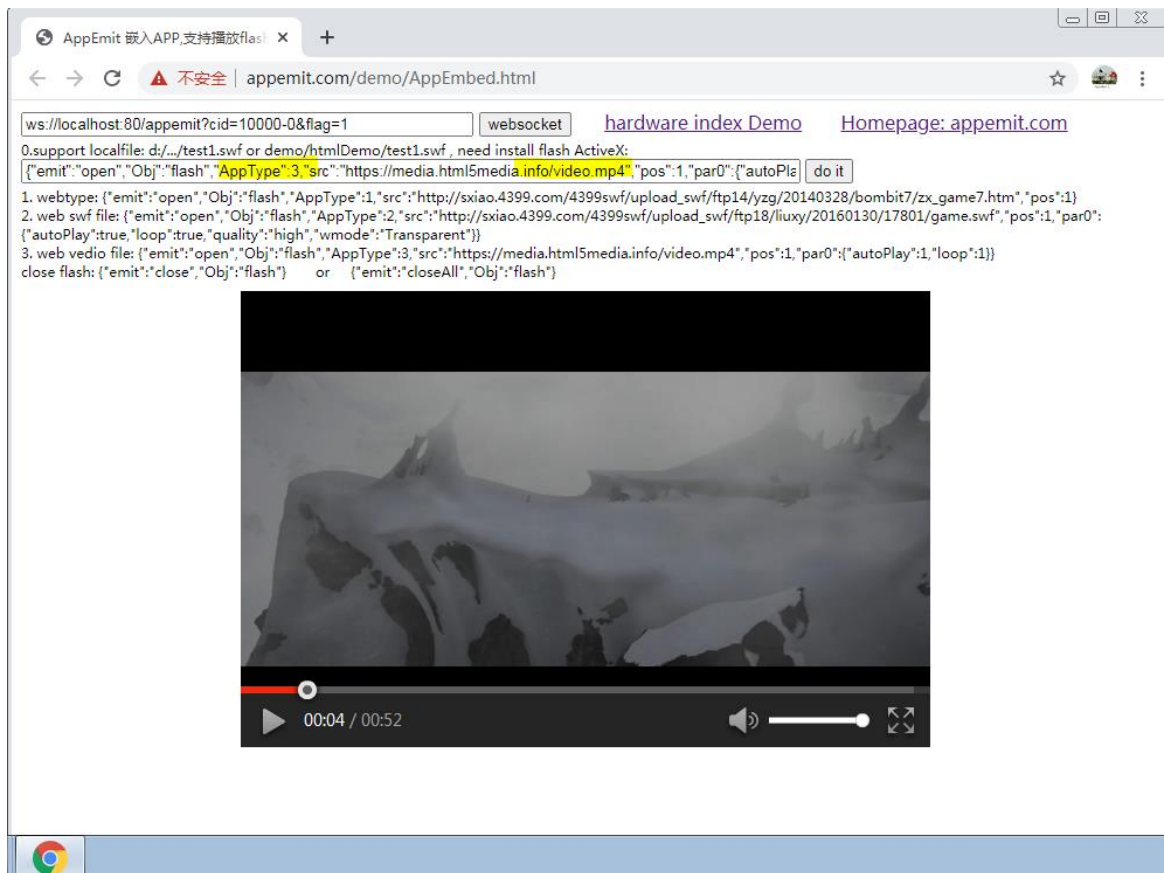
{ "emit": "open", "Obj": "flash", "AppType": 2, "src": "http://sxiao.4399.com/4399swf/upload_swf/ftp18/liuxy/20160130/17801/game.s
wf", "pos": 1, "par0": { "autoPlay": true, "loop": true, "quality": "high", "wmode": "Transparent" } }



2.4.3 NPAPI-网络媒体文件

使用 Appemit 程序自带的插件 NPSWF32.dll，打开网络媒体文件，包括 flv, mp4 等。连接授权后，发送命令“AppType”:3 的形式。

```
{"emit":"open","Obj":"flash","AppType":3,"src":"https://media.html5media.info/video.mp4","pos":1,"par0":{"autoplay":1,"loop":1}}
```



2.4.4 ActiveX 形式

2.4.4.1 打开网络 flash 文件

打开 demo 下的 AppEmbed.html, 连接授权后，发送使用 ActiveX (“AppType”:4) 打开网络 flash 文件命令，参数如下。

```
{"emit":"open","Obj":"flash","AppType":4,"src":"http://img1.yo4399.com/swf/0/0ff035e0e96584c07df65ab3636f72.swf","pos":1,"par0":{"autoplay":1,"toolbar":0,"rightMenu":0,"hitCaption":0,"hideStop":0,"loop":1,"volumeMute":0,"flashVars":"a=0&b=0&c=SetInSrc"}}
```

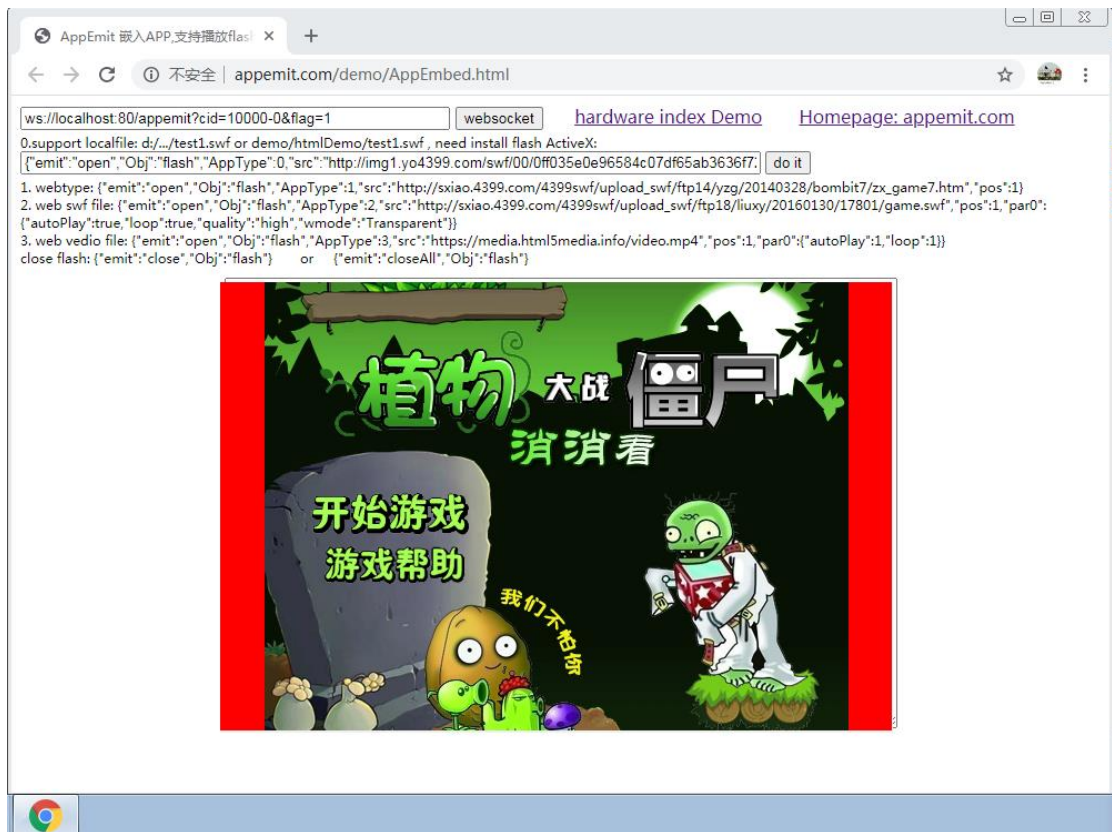
注意事项：

在客户端需要下载安装 flash player ActiveX。

路径是 / 或许 \\

flashVars 可以设置在 src 中

AppType 正数为嵌入 APP，负数为浮动窗口，后同。

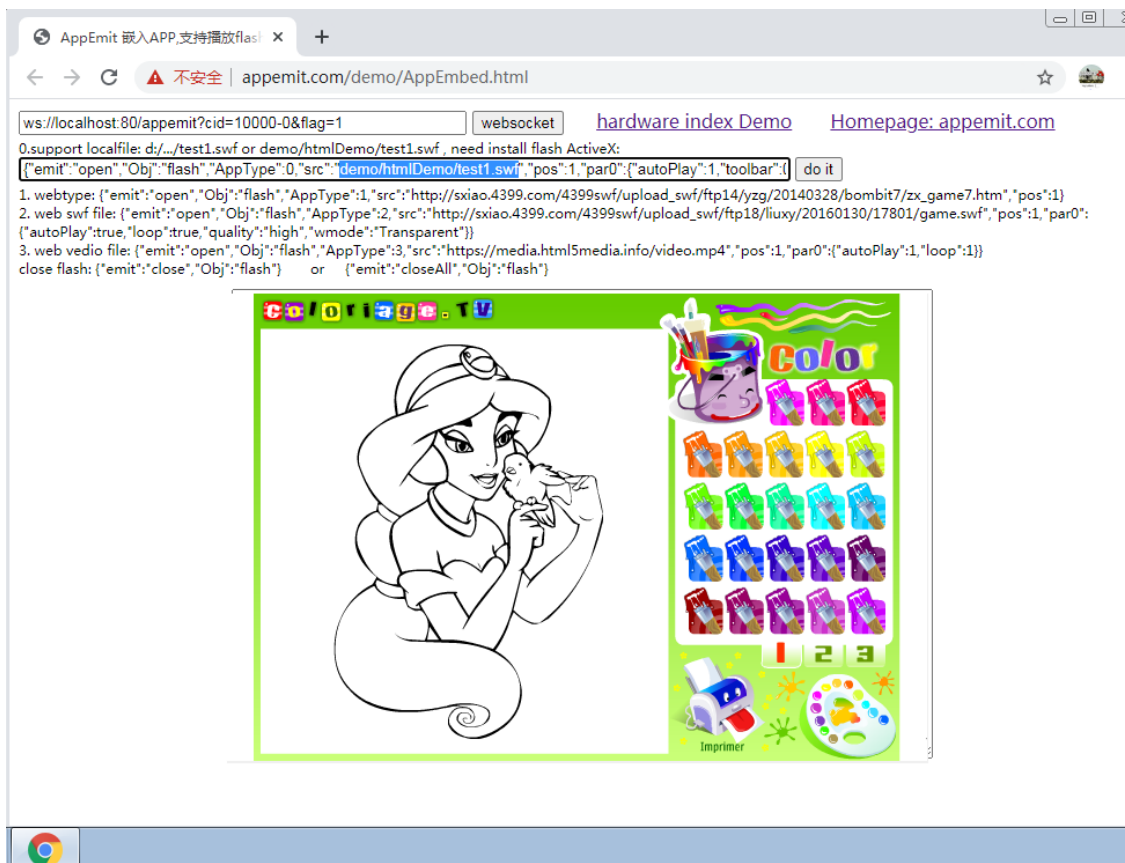


刷新即可关闭 flash

2. 4. 4. 2打开本地 flash 文件

可以是绝对或者相对路径，相对于 AppEmit.exe 的路径：“demo/htmlDemo/test1.swf”。

```
{ "emit": "open", "Obj": "flash", "AppType": 4, "src": "demo/htmlDemo/test1.swf", "pos": 1, "par0": { "autoplay": 1, "toolbar": 0, "rightMenu": 0, "hitCaption": 0, "hideStop": 0, "loop": 1, "volumeMute": 0, "flashVars": "a=0&b=0&c=SetInSrc" } }
```



2.4.5 关闭

1. 刷新即可关闭 flash
2. `{"emit":"close","Obj":"flash"}`
3. `{"emit":"closeAll","Obj":"flash"}`

2.5 Web

具有右键菜单。

2.5.1 IE 内核

"AppType":1 使用 IE 内核打开网页

```
{"emit":"open","Obj":"web","AppType":1,"pos":1,"par":{"htmlStr":null,"HttpServer_startUrl":null,"URL":"http://www.appemit.com"},"par0":{"header":null,"noScriptErr":true,"UIFLAG":null,"DLCTL":null,"userAgent":null,"crossDomain":true}}
```

设置 htmlStr 可以直接打开 html 源码。

设置 HttpServer_startUrl, 可以打开本地的 html 文件。

设置 URL 打开网页。三者优先级依次下降。

2.5.2 Webkit 内核

"AppType":2 使用 webkit 开源内核打开网页

```
{"emit":"open","Obj":"web","AppType":2,"pos":1,"par":{"htmlStr":null,"HttpServer_startUrl":null,"URL":"http://www.appemit.com"},"par0":{"header":null,"userAgent":null,"crossDomain":true,"download":1}}
```

设置 htmlStr 可以直接打开 html 源码。

设置 HttpServer_startUrl, 可以打开本地的 html 文件。

设置 URL 打开网页。三者优先级依次下降。

2.5.3 blink 内核

"AppType":3 使用开源 blink 内核(webkit 加强版)打开网页, 可以浏览网页, 支持前进 后退 刷新 复制 粘贴, 支持下载。

```
{"emit":"open","Obj":"web","AppType":3,"pos":1,"par":{"htmlStr":null,"HttpServer_startUrl":null,"URL":"http://www.appemit.com"},"par0":{"header":null,"userAgent":null,"crossDomain":true,"download":1}}
```

设置 htmlStr 可以直接打开 html 源码。

设置 HttpServer_startUrl, 可以打开本地的 html 文件。

设置 URL 打开网页。三者优先级依次下降。

2.6 office

2.6.1 PDF

"AppType":4 使用开源 blink 内核打开阅读 PDF 文件, 支持本地和网络 PDF。

```
{"emit":"open","Obj":"web","AppType":4,"pos":1,"par":{"URL":"/demo/AppEmit_help.pdf"},"par0":{"rightMenu":1}}}
```

设置 URL 打开 PDF 文件。

2.6.2 Office

使用微软 com 技术创建、打开阅读、编辑 Office，包括 word\excel\PPT 文件，支持本地和网络文件。

支持在 javascript 中添加类似 VBA 代码动态控制, 同时可以在浏览器网页里执行命令。

2.6.3 Office DsoFramer （不稳定）

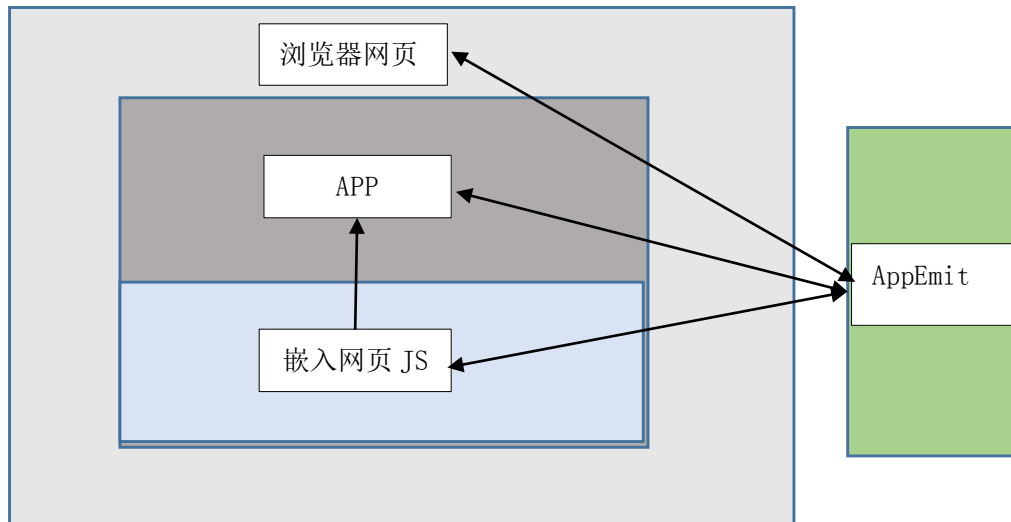
使用微软开源控件 DsoFramer 创建、打开阅读、编辑 Office，包括 word\excel\PPT 文件，支持本地和网络文件。

```
{ "emit": "open",
  "Obj": "office",
  "AppType": 1,
  "src": "/demo/htmlDemo/file/xl.xls",
  "pos": 1,
  "par0": { "Caption": true,
            "ProtectDoc": 0,
            "Toolbars": true,
            "btnFile": true,
            "MNU_NEWBLANK": 0,
            "MNU_NEW": 0,
            "MNU_OPEN": 0,
            "MNU_CLOSE": 0,
            "MNU_SAVE": 0,
            "MNU_SAVEAS": 0,
            "MNU_PGSETUP": 0,
            "MNU_PRINT": 0,
            "MNU_PROPS": 0,
            "show_UpdateTool": true }
}
```

设置 src 打开 office 文件。

2.7 DLL\组件调用

可以调用标准的 COM 组件、Activex 以及 DLL，支持可视化 ocx，同时可以开发互动的 dll，能在浏览器中控制调用的组件或者程序。



2.7.1 可视化 com 组件

- Flash ocx 调用, 能在网页中执行命令, 控制 flash, 举例见 demo/htmlDemo/dll.html
`{"emit": "open", "Obj": "dll", "AppType": 1, "src": [], "pos": 1, "par": {"htmlStr": null, "HttpServer_startUrl": null, "URL": "/demo/htmlDemo/js/dllJs.html", "webJs": 1}}`

调用 C:/Windows/SysWOW64/Macromed/Flash/Flash32_32_0_0_403.ocx 控件, 需要自行安装 flash.

可以在 dllJs.html 进一步设置参数 AppType==1 情况, 和过程调用代码。

同时也能通过浏览器网页发布命令。

- 第三方控件 ReportX.ocx, 联系 AppEmit 检查安全性后配置在 /plugins/thirdparty 目录中, 直接调用
 报表控件 reportX 调用。

能在网页中执行命令, 控制 reportX, 举例见 demo/htmlDemo/dll.html

- `{"emit": "open", "Obj": "dll", "AppType": 2, "src": [], "pos": 1, "par": {"htmlStr": null, "HttpServer_startUrl": null, "URL": "/demo/htmlDemo/js/dllJs.html", "webJs": 1}, "par0": {"AppMethod": "msg Synchronization component need set: POST" }}`

调用 "/plugins/thirdparty/report/ReportX.ocx" 控件

可以在 dllJs.html 进一步设置参数 AppType==2 情况, 和过程调用代码。

同时也能通过浏览器网页发布命令。

2.7.2 第三方 Dll 调用

- dll 调用, 第三方控件, 联系 AppEmit 检查安全性后配置在 /plugins/thirdparty 目录中, 直接调用, 能在网页中执行命令, 举例见 demo/htmlDemo/dll.html
`{"emit": "open", "Obj": "dll", "AppType": 4, "src": [], "pos": 1, "par": {"htmlStr": null, "HttpServer_startUrl": null, "URL": "/demo/htmlDemo/js/dllJs.html", "webJs": 1, "AppShow": false}, "par0": {"AppMethod": "msg Synchronization component need set: POST" }}`

可以在 dllJs.html 进一步设置参数 AppType==4 情况, 和过程调用代码, 演示调用 add 方法。

同时也能通过浏览器网页发布命令。

2.8 DLL 开发

自有开发的私人 DLL 或者组件，需要申请 Pid 后开发 dll，安装在/plugins/private/目录下，需要双方验证成功后才能调用

1. 参考 demo/htmlDemo/dll/dll_demo1.c, 提供了接口规范。 使用最新版本 TCC 开发的 dll。

调用 dll_demo1.dll，能在嵌入网页中执行命令，控制 dll_demo1, 举例见 demo/htmlDemo/dll.html

```
{"emit": "open", "Obj": "dll", "AppType": 3, "src": [], "pos": 1, "par": {"htmlStr": null, "HttpServer_startUrl": null, "URL": "/demo/htmlDemo/js/dllJs.html", "webJs": 1, "AppShow": false}, "par0": {"AppMethod": "msg Synchronization component need set: POST" }}
```

调用/plugins/private/50FCF891-1B93-4AE5-8A66-AB26A3C03378/dll_demo1.dll

可以在 dllJs.html 进一步设置参数 AppType==3 情况，和过程调用代码。

同时也能通过浏览器网页发布命令 runCmd。

3 DLL 开发

3.1.1 开发

自有开发的私人 DLL 或者组件，需要申请 Pid （pid>=guid=clsId）可使用 GUID 工具自行生成后开发 dll，安装在/plugins/private/目录下，需要双方验证成功后才能调用

参考 demo/htmlDemo/dll/dll_demo1.c, 提供了接口规范。 使用最新版本 TCC 开发的 dll。

主要是 5 个函数接口, App 开头的函数和变量名称不要改变

AppDll_init	dll 初始化，用户输入参数 认证
AppDll_loaded	dll 启动后执行
AppDll_destroy	dll 退出前执行
AppDll_RevMsg	互动时 接收 消息
AppDll_SendMsg	互动时 反馈 消息

模板如下, 使用了 cJSON 开源库。

```
#include <windows.h>
// #include <string.h>
#include "cJSON.h"  //需要下载开源库 cJSON
#include "cJSON.c"

// 开发可以互动信息的动态链接库 Dll
// http://www.appemit.com
////生成 DLL 必须使用最新版 tcc 扩展库才能支持 UTF8, UTF16 字符串
/*
```

入口函数, 该函数可以有也可以没有。

入口函数会自动加锁以保证线性调用, 要避免在 DllMain 内调用下列函数:

- 1、调用 LoadLibrary 或其他可能加载 DLL 的 API 函数(CreateProcess 等)
 - 2、可能再次触发 DllMain 的函数, 例如 CreateThread, ExitThread
 - 3、GetModuleFileName, GetModuleHandle 等其他可能触发系统锁的 API 函数
- 总之在 DllMain 最好不要调用 API 函数.

```

*/

/*
    主要是 5 个函数接口, App 开头的函数和变量名称不要改变
    AppDll_init      dll 初始化, 用户输入参数 认证
    AppDll_loaded    dll 启动后执行
    AppDll_destroy   dll 退出前执行
    AppDll_RevMsg     互动时 接收 消息
    AppDll_SendMsg    互动时 反馈 消息
*/
char *mycid = "00000-1"; //申请的公司产品 cid , 需要修改
char *clsId = "50FCF891-1B93-4AE5-8A66-AB26A3C03378"; // pid>=guid=clsId
可使用 GUID 工具自行生成
int dllThid; //本进程 ID
// sid; //websocket 里面设置的, 一般为用户 ID 或者 sessionId, 唯
一

// rid ; //第 rid 次调用
int AppAuth=0; //

//测试
char* joinStr(char *s1, char *s2, char *s3)
{
    char *result = malloc(strlen(s1)+strlen(s2)+strlen(s3)+1); //+1 for the
zero-terminator
    if (result == NULL) exit (1);
    strcpy(result, s1);
    strcat(result, s2);
    strcat(result, s3);
    return result;
}

int __stdcall DllMain(void * hinstDLL, unsigned long fdwReason, void *
lpvReserved) {

    if (fdwReason == 1/*DLL_PROCESS_ATTACH*/) {

    }
    return 1;
}

//__declspec(dllexport) 声明导出函数
__declspec(dllexport) int AppDll_RevMsg(HWND hwnd, char *ids, char *msg)
{

    AppDll_SendMsg(hwnd, ids, msg);
    return 0;

}

```



```

__declspec(dllexport) int AppDll_SendMsg(HWND hwnd, char *ids, char *msg)
{
    //检查是否有 clsId
    cJSON * ids_json= cJSON_Parse(ids);
    if (!ids_json) {ids_json=cJSON_CreateObject();}
    if (!cJSON_GetObjectItem(ids_json,"clsId")) {
        cJSON_AddStringToObject(ids_json, "clsId", clsId); //必需字段
    }
    ids=cJSON_PrintUnformatted(ids_json);
    cJSON_Delete(ids_json);

    struct {char * ids;char * msg; } callBackMsg = {
        .ids=ids,
        .msg =msg, // 只会反馈 msg 里面 data 字段到
websocket

    };

    SendMessage(
        hwnd, 0xACCE ,
        "AppOnMsg({string ids;string msg;})", //要调用的窗体函数名( 结构体
原型声明 ); 结构体原型声明应使用 API 语法
        &callBackMsg //将前面定义的结构体作为调用参数
    );

    return 0;

    /*
    0xACCE= WM_THREAD_CALLBACK 使所有回调安全的转发到 UI 线程。
    WM_THREAD_CALLBACK 可以跨线程跨语言并且不需要创建回调线程, 适用任何普
通 winform 对象。

    */
}

__declspec(dllexport) int AppDll_init(HWND hwnd, char *ids, char *msg)
{
    //判断获得 AppEmit 提供 ids(格式 json)里面 cid sid pid AuthKey 等数据
    //和 websocket 的自行 web 提供的 Json 里面的 data, 判断验证来源是否正确
    //
ids="{"cid\":"00000-
1\","sid\":"f1s\","rid\":2333,"AuthKey\":"000\","clsId\":"50FCF891-1B93-
4AE5-8A66-AB26A3C03378\"}";

    //具体数据和判断需要修改
    cJSON * ids_json= cJSON_Parse(ids);
    // char * ids3= cJSON_PrintUnformatted(ids_json); //如果解析报错尝试
使用 cJSON
    // cJSON * ids_json2= cJSON_Parse(ids3);

```

```

        if (!cJSON_GetObjectItem(ids_json, "clsId"))
        || !cJSON_GetObjectItem(ids_json, "cid")
        || !cJSON_GetObjectItem(ids_json, "AuthKey")) { cJSON_Delete(ids_json); return -1;}

        dllThid = GetCurrentThreadId();
        cJSON_AddNumberToObject(ids_json, "dllThid", dllThid); //必需
        cJSON_AddStringToObject(ids_json, "more", NULL); //备用字段
        //必须反馈验证
        if
        (strcmp(cJSON_GetObjectItem(ids_json, "cid")->valuelstring, mycid)==0           &&
        strcmp(cJSON_GetObjectItem(ids_json, "AuthKey")->valuelstring, "000")==0       &&
        strcmp(cJSON_GetObjectItem(ids_json, "clsId")->valuelstring, clsId)==0) {

                //msg 若反馈必须有  "\clsId\ " \AppAuth\". AppAuth 为1 clsId 一致
                才继续 // 支持\ " 或者 '

                msg = joinStr("{ \"data\": { \"code\":200, \"cid\": \"00000-1\", \"sid\": \"123\", \"rid\": -1, \"rec\": \"\", msg, \"AppStep\": \"init\" } }");
                cJSON_AddNumberToObject(ids_json, "AppAuth", 1);

                char *ids2=cJSON_PrintUnformatted(ids_json);
                //必须反馈验证

                AppDll_SendMsg(hwnd, ids2, msg);

        } else {
                cJSON_AddNumberToObject(ids_json, "AppAuth", 0);
                char *ids2=cJSON_PrintUnformatted(ids_json);
                // 反馈验证
                AppDll_SendMsg(hwnd, ids2, NULL);
        }

        cJSON_Delete(ids_json);

        return 0;
    }
    __declspec(dllexport) int AppDll_loaded(HWND hwnd, char *ids, char *msg)
    {
        //处理业务 //msg 若反馈必须有 data
        // msg = joinStr("{ \"data\": { \"code\":200, \"cid\": \"00000-1\", \"sid\": \"123\", \"rid\": -1, \"rec\": \"\", msg, \"AppStep\": \"loaded\" } }");

        //若反馈则
        AppDll_SendMsg(hwnd, ids, msg); //如果 msg 中没有 data, 则不反馈到浏览
器中

```

```

        return 0;
    }
    __declspec(dllexport) int AppDll_destroy(HWND hwnd, char *ids, char *msg)
    {
        //处理业务 msg 若反馈必须有 data
        //msg = joinStr("{\"data\":{\"code\":200,\"cid\":\"00000-1\", \"sid\":\"123\", \"rid\":-1, \"rec\":\", msg, \"\", \"AppStep\":\"destroy\"}");

        //若反馈则
        AppDll_SendMsg(hwnd, ids, msg);

        return 0;
    }
    //测试
    __declspec(dllexport) int Add(int a, int b )
    {
        return a+b;
    }
    __declspec(dllexport) int AppDll_init(HWND hwnd, char *ids, char *Json)
    {
        //判断获得 AppEmit 提供 ids(格式 json)里面 cid sid pid Auth 等数据
        //和 websocket 的自行 web 提供的 Json 里面的 data, 判断验证来源是否正确

        //msg 若反馈必须有 \"AppStep\", \"clsId\" \"AppAuth\"。AppAuth 为 true
        clsId 一致 才继续
        // 支持\" 或者 ' 如果有 data, 只发送 data 字段到浏览器 websock 接收
        char *msg = joinStr(joinStr("{\"data\":{\"code\":200,\"cid\":\"00000-1\", \"sid\":\"123\", \"rid\":-1, \"rec\":\", Json, \"\", \"AppStep\":\"init\", \"AppStep\":\"init\", \"AppAuth\":true, \"clsId\":\"\", clsId, \"\"}");
        dllThid = GetCurrentThreadId();
        //必须反馈验证
        AppDll_OnMsg(hwnd, ids, msg);

        return 0;
    }
    __declspec(dllexport) int AppDll_loaded(HWND hwnd, char *ids, char *Json)
    {
        //处理业务 //msg 若反馈必须有 \"clsId\". 如果有 data, 只发送 data 到浏览器 websock 接收
        char *msg = joinStr(joinStr("{\"data\":{\"code\":200,\"cid\":\"00000-1\", \"sid\":\"123\", \"rid\":-1, \"rec\":\", Json, \"\", \"AppStep\":\"loaded\", \"AppStep\":\"loaded\", \"clsId\":\"\", clsId, \"\"}");

        //若反馈则 OnMsg
        AppDll_OnMsg(hwnd, ids, msg);
    }

```

```

        return 0;
    }
    __declspec(dllexport) int AppDll_destroy(HWND hwnd, char *ids, char *Json)
    {
        //处理业务 msg 若反馈必须有  \“clsId\” .如果有 data, 只发送 data 到浏览器
        websock 接收
        char *msg
        joinStr(joinStr("{\“data\“:{\“code\“:200,\“cid\“:\“00000-
1\“, \“sid\“:\“123\“, \“rid\“:-
1, \“rec\“:\“, Json, \“, \“AppStep\“:\“destroy\“, \“AppStep\“:\“destroy\“, \“clsId\“:\“
”), clsId, \“\“}”);

        //若反馈则 OnMsg
        AppDll_OnMsg(hwnd, ids, msg);

        return 0;
    }
    //测试
    __declspec(dllexport) int Add(int a, int b )
    {
        return a+b;
    }
}

```

3.1.2 调用

1. 生成 dll_demo1.dll 后, 安装在/plugins/private/Pid/下面
 /plugins/private/50FCF891-1B93-4AE5-8A66-AB26A3C03378/dll_demo1.dll”
 就能在能在网页中执行命令, 调用互动 dll_demo1, 举例见 demo/htmlDemo/dll.html
 {"emit": "open", "Obj": "dll", "AppType": 3, "src": [], "pos": 0, "par": {"htmlStr": nul
 1, "HttpServer_startUrl": null, "URL": "/demo/htmlDemo/js/dllJs.html", "webJs": 1, "App
 Show": false}, "par0": {"AppMethod": "msg Synchronization component need set: POST" }}

可以在 dllJs.html 进一步设置参数 AppType==3 情况, 在 dllJs.html 中也可以设置
 src 参数和过程, 调用代码。

同时也能通过浏览器网页发布命令 runCmd, 执行 AppDll_RevMsg 方法。

```

        codeStr= txt2code(function() {/*
        AppJsObject.dll_demo1.AppDll_RevMsg(AppDll_hwnd, '', '\“data\“:send from
JS", "info":3}') );
        * /});
        var Req={"emit": "runCmd", "Obj": "dll", "codeStr":codeStr }
        EmitReq(Req);
    
```

3.1.2.1 嵌入网页 JS 控制

➤ AppJs_init=function(AppType)

如果在浏览器网页中没有设置 src, 可以在这里设置。

```

src:[ {
        // webJs0:0,
        //只
        能设置为 0 关闭本 js 的控制 ; 或者注释 为默认打开控制
    
```

```
createType:"createDll" //
dll 调用
, objName:"dll_demo1"
, dllFile:"/plugins/private/50FCF891-1B93-4AE5-8A66-AB26A3C03378/dll_demo1.dll" // 必须 \\ / 放在 /plugins/private/clsId/文件夹下面
, clsId : "50FCF891-1B93-4AE5-8A66-AB26A3C03378" //guid 发送给 dll 信息
, iid:null
, data:'{"msg":"init from JS","info":1}'
//发送给 dll 数据

} //暂时支持一个控件
]
, AppShow:false //整体不可见 必须设置, 默认可见
```

- AppJs_loaded=function(AppType) {
//系统启动 APP ,产生 AppJsObject. shockwave 对象后执行
- AppJsObject.dll_demo1.AppDll_loaded(AppDll_hwnd, "", {"msg":"loaded from JS","info":2}');

➤ AppJs_destroy=function() { //AppJsObject.shockwave 销毁前执行
AppJsObject.dll_demo1.AppDll_destroy(AppDll_hwnd, "", {"msg":"destroy from JS","info":3}');
➤ AppJs_closed=function(status) { //AppJsObject.shockwave 关闭后执行。已经没有 AppJsObject.shockwave 对象了

3. 1. 2. 2浏览器网页执行代码命令控制

AppJsObject.dll_demo1
Var Req={"emit":"runCmd","Obj":"dll","codeStr": codeStr}
EmitReq(Req);
通过前面 dllJs.html 定义的 AppJsObject.dll_demo1, 可以直接控制 dll_demo1。当然也可以不需要 dllJs.html 定义, 能在浏览器网页中 websocket 发送使用"emit":"runCmd"命令发送 AppJs_init、AppJs_loaded, 代码也可以。

4 参数

4. 1 连接

ws://localhost:80/appemit?cid=00000-1&sid=1&flag=1

名称	设置	含义	说明
协议	ws	SSL 为 wss	
网址	Localhost 127. 0. 0. 1		
Port	[80, 8617, 8618, 8619, 9780]	数值 或者 数组 , 默认 80 。 ws_force=9780	依次尝试打开端口。若都失败, 则尝试关闭占用 ws_force 端口的进程, 并

			打开 ws_force 端口。 都可以在 config.in 修改
	[443, 5124, 5125, 5126, 43100]	ssl 默认 443。wss_force=43100	同上。
path	appemit	必需	
para	cid	必需。00000-1 为免费账号。	全集。
	sid	字符串，通常情况可选。唯一 session 或者用户名 ID	测试后最好在 js 中实现隐藏。如果需要调用私有 APP，则必须有，否则无法互相通话。
	flag	可选。默认 0，非调试。 1 调试	

4.2 初始化数据

```
{"emit": "init", ...}
```

```
var init_AE={
    "emit": "init",
    "clientKey": "temp-00000000000",
    "browser": ThisBrowser,
    "wsUrl": wsUrl,
    "sid": "123456",
    "gid": "[1, 2]"
}
```

名称	设置	含义	说明
emit	init	必需。初始化请求。	
clientKey	temp-00000000000	必需，客户端，与 cid 对应。	保密，js 应该混淆加密。
browser	ThisBrowser	必需。使用浏览器。默认	
wsUrl	wsUrl	必需。默认	可以在 config.in 修改
sid	字母数字下划线短线	必需。用户或者 session，唯一才可以正常通话。	生产环境，同一设置于此。
gid	数组	非必需。群	一个 sid 可有不同 gid
interval_Pong	整数	非必需。毫秒，保持心跳。	Firefox 最好设置，3000。

4.3 命令

4.3.1 硬件

4.3.1.1 硬件信息

```
{"emit": "hardWare", "Obj": "pc", "rid": 1}
```

名称	设置	含义	说明
emit	hardWare	必需。通信请求。	
Obj	pc	必需。目标对象。	
rid	整数	必需。表示具体请求标号。	每次发送命令自动加 1。后续省略描述。

4.3.1.2 USB Hid 通信接口信息

`{"emit": "hid"}`

名称	设置	含义	说明
emit	hid	必需。获得 hid 通信接口信息。	

4.3.1.3 热拔事件

`{"emit": "deviceNotification", "driveType": 2}`

名称	设置	含义	说明
emit	deviceNotification	必需。监控热拔请求。	
driveType	数值或者数组 2 , [1, 2], -1	-1 关闭监控 1 DRIVE_NO_ROOT_DIR 说明无效的 2 DRIVE_REMOVABLE 可移动磁盘 3 DRIVE_FIXED 固定磁盘 4 DRIVE_REMOTE 网络磁盘 5 DRIVE_CDROM 光驱 6 DRIVE_RAMDISK 为 RAM 磁盘	

4.3.2 通话

`{"emit": "msg", "toSids": ["2"], "toGids": [1, 2], "data": "hi, I' am Tom."}`

名称	设置	含义	说明
emit	msg	必需。通信事件请求。	
toSids	必需要有一个	非必需。可以是数组。	
toGids		非必需。可以是数组。	
data		必需。	

4.3.3 APP 事件

参数格式如下

名称	设置	含义	说明
emit	open	必需。打开控件 APP 通信事件请求。	
Obj		必需。 flash 默认 word 后续支持 excel 后续支持 CAD 后续支持	
AppType	± 1 ± 2 ± 3 ± 4	正数 表示嵌入 负数 浮动窗口 在 Obj 不同时，AppType 的含义不一样。	当 AppType 负数浮动窗口时，默认 APP 为置顶。 取消置顶 <code>{"emit": "setPar", "Obj": "flash", "topMost": false}</code>
par0			注意不同 APP 可能不同

header		头部	
userAgent		代理	只写属性，不可读。
crossDomain	bool	默认 true True false	是否跨域
rightMenu	右键菜单	-1 不处理，控件原有状态 0/null 禁用菜单 1 自定义简单菜单	
show_Update Tool	显示下载插件等待窗口	默认 true false 不显示	需要下载更新包显示。可以直接在 https://github.com/ap-pemit/appemit/plugins 下载全集的插件

4.3.3.1 "AppType":1 打开 flash 网页 webkit 内核

```
{"emit":"open","Obj":"flash","AppType":1,"src":"http://sxiao.4399.com/4399swf/upload_swf/ftp14/yzg/20140328/bombit7/zx_game7.htm","pos":1,"par0":{"header":null,"userAgent":null,"crossDomain":true}}
```

名称	设置	含义	说明
emit	open	必需。打开控件 APP 通信事件请求。	
Obj	flash	必需。	
AppType	1	必需。 0 ActiveX 1 web 2 web flash 文件 3 web 媒体文件	若为负数-1, 则是浮动窗口
src		必需。	AppType 为 0 时支持本地文件，可以是相对或者绝对路径。
pos	{"left":372,"top":203,"width":606,"height":406}	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器，自动识别的位置需要优化
Par0		可选。	
header		头部	
userAgent		代理	只写属性，不可读。
crossDomain	bool	默认 true True false	是否跨域
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	

4.3.3.2 "AppType":2 打开网络 flash

```
{"emit":"open","Obj":"flash","AppType":2,"src":"http://sxiao.4399.com/4399swf/upload_swf/ftp18/liuxy/20160130/17801/game.swf","pos":1,"par0":{"autoplay":true,"loop":true,"quality":"high","wmode":"Transparent","header":null,"userAgent":null,"crossDomain":true}}
```

名称	设置	含义	说明
emit	open	必需。打开控件 APP 通信事件请求。	
Obj	flash	必需。	
AppType	2	必需。 0 ActiveX 1 web 2 web flash 文件 3 web 媒体文件	使用 webkit 内核打开, 没有默认的右键菜单。 若为负数-2, 则是浮动窗口
src		必需。	AppType 为 0 时支持本地文件, 可以是相对或者绝对路径。
pos	{"left":372,"top":203,"width":606,"height":406}	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器, 自动识别的位置需要优化
par0	autoplay	可选。默认 true	参考 flash 官方默认参数。
	loop	可选。默认 true	参考 flash 官方默认参数。
	quality	可选。默认 high	参考 flash 官方默认参数。
	wmode	可选。默认 Transparent	参考 flash 官方默认参数。
header		头部	
userAgent		代理	只写属性, 不可读。
crossDomain	bool	默认 true True false	是否跨域
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	

4.3.3.3 "AppType":3 打开 flash 等媒体文件

```
{"emit":"open","Obj":"flash","AppType":3,"src":"https://cdn.jsdelivr.net/gh/appemit/appemit/docs/video/x1.flv","pos":1,"par0":{"autoplay":1,"loop":1,"header":null,"userAgent":null,"crossDomain":true}}
```

名称	设置	含义	说明
emit	open	必需。打开控件 APP 通信事件请求。	
Obj	flash	必需。	
AppType	3	必需。 0 ActiveX 1 web 2 web flash 文件	若为负数-3, 则是浮动窗口

		3 web 媒体文件	
src		必需。	AppType 为 0 时支持本地文件，可以是相对或者绝对路径。
pos	{"left":372,"top":203,"width":606,"height":406}	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器，自动识别的位置需要优化
Par0	autoPlay	可选。默认 1	参考 https://player.alicdn.com/aliplayer/setting/setting.html
	loop	可选。默认 1	
header		头部	
userAgent		代理	只写属性，不可读。
crossDomain	bool	默认 true Ture false	是否跨域
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	

4.3.3.4 "AppType":4 打开 flash 文件

```
{"emit":"open","Obj":"flash","AppType":4,"src":"http://img1.yo4399.com/swf/00/0ff035e0e96584c07df65ab3636f72.swf","pos":1,"par0":{"autoplay":1,"toolbar":0,"rightMenu":0,"hitCaption":0,"hideStop":0,"loop":1,"volumeMute":0,"flashVars":{"a=0&b=0&c=SetInSrc"}}
```

名称	设置	含义	说明
emit	open	必需。打开控件 APP 通信事件请求。	
Obj	flash	必需。 flash 默认 word 后续支持 excel 后续支持 CAD 后续支持	
AppType	4	必需。 4 ActiveX 1 web 2 web flash 文件 3 web 媒体文件	设为 4 时，如何本地没有安装 ActiveX 则默认使用方式 2 打开 若为负数-4, 则是浮动窗口
src		必需。	AppType 为 0 时支持本地文件，可以是相对或者绝对路径。
pos	{"left":372,"top":203,"width":	必需。 1 默认使用代码自动识别的位置。	

	606,"height":406}		
par0	autoPlay	可选。 0 不自动播放 1 自动播放，默认	
	toolbar	可选。 0 没有控制条，默认 1 有控制条	
	rightMenu	可选。 0 1	
	hitCaption	可选。 0 左按鼠标不能拖动，默认 1 左按鼠标能拖动	
	hideStop	可选。 0 隐藏不见时不停止播放，默认 1 隐藏不见时停止播放	
	loop	可选。 0 不自动循环播放， 1 循环播放，默认	
	volumeMute	可选。 0 不静音，默认 1 静音	
	flashVars	可选。	可以设置在 src 里面
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	

4.3.3.5 IE 内核打开网页

```
{ "emit": "open", "Obj": "web", "AppType": 1, "pos": 1, "par": { "htmlStr": null, "HttpServer_startUrl": null, "URL": "http://www.appemit.com" }, "par0": { "header": null, "noScriptErr": true, "UIFLAG": null, "DLCTL": null, "userAgent": null, "crossDomain": true, "rightMenu": null } }
```

名称	设置	含义	说明
emit	open	必需。打开网页事件请求。	
Obj	web	必需。	
AppType	1	必需。 1 IE 内核 2 webkit 内核 3 blink 内核	有默认的右键菜单。 若为负数-1, 则是浮动窗口
pos	{ "left": 372, "top": 203, "width": 606, "height": 406 }	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器，自动识别的位置需要优化

par		必需。优先级别依次下降	三个参数必须有一个不是空。
htmlStr		Html 代码	
HttpServer_startUrl		以服务器形式打开本地 html 文件路径，可以是绝对或者相对路径。 /为分隔符。	
URL	http://www.appemitt.com 或者 /demo/html Demo/html. html	支持网页地址或者本地 html 文件路径。	
Par0		可选。	
noScriptErr	bool	默认 true True false	
UIFLAG		_UIFLAG_DIALOG=@0x1/*_UIFLAG_DIALOG*/ _UIFLAG_DISABLE_HELP_MENU=@0x2/*_UIFLAG_DISABLE_HELP_MENU*/ _UIFLAG_NO3DBORDER=@0x4/*_UIFLAG_NO3DBORDER*/ _UIFLAG_SCROLL_NO=@0x8/*_UIFLAG_SCROLL_NO*/ _UIFLAG_DISABLE_SCRIPT_INACTIVE=@0x10/*_UIFLAG_DISABLE_SCRIPT_INACTIVE*/ _UIFLAG_OPENNEWWIN=@0x20/*_UIFLAG_OPENNEWWIN*/ _UIFLAG_DISABLE_OFFSCREEN=@0x40/*_UIFLAG_DISABLE_OFFSCREEN*/ _UIFLAG_FLAT_SCROLLBAR=@0x80/*_UIFLAG_FLAT_SCROLLBAR*/ _UIFLAG_DIV_BLOCKDEFAULT=@0x100/*_UIFLAG_DIV_BLOCKDEFAULT*/ _UIFLAG_ACTIVATE_CLIENTHIT_ONLY=@0x200/*_UIFLAG_ACTIVATE_CLIENTHIT_ONLY*/ _UIFLAG_OVERRIDEBEHAVIORFACTORY=@0x400/*_UIFLAG_OVERRIDEBEHAVIORFACTORY*/ _UIFLAG_CODEPAGELINKEDFONTS=@0x800/*_UIFLAG_CODEPAGELINKEDFONTS*/ _UIFLAG_URL_ENCODING_DISABLE_UTF8=@0x1000/*_UIFLAG_URL_ENCODING_DISABLE_UTF8*/ _UIFLAG_URL_ENCODING_ENABLE_UTF8=@0x2000/*_UIFLAG_URL_ENCODING_ENABLE_UTF8*/	可以使用一个或多个 _UIFLAG_ 前缀的常量自定义外观, 多个常量之间用位或操作符() 连接。 LOG 禁止选中文本(用于 web ui) _UIFLAG_SCROLL_NO 禁用滚动条 _UIFLAG_NO3DBORDER 禁用所有窗口 3D 边框 _UIFLAG_NO3DOUTERBORDER 禁用顶层窗口 3D 边框 _UIFLAG_DISABLE_HELP_MENU 在菜单中移除帮助菜单 _UIFLAG_DISABLE_SCRIPT_INACTIVE 窗口激活以前不运行网页脚本 _UIFLAG_OPENNEWWIN 在新窗口打开链接 _UIFLAG_FLAT_SCROLLBAR 显示平面滚动条 _UIFLAG_ACTIVATE_CLIENTHIT_ONLY 仅在用户点击客户区时激活(非客户区指滚动条等位置) _UIFLAG_URL_ENCODING_DISABLE_UTF8 禁用 UTF8 发送 URL _UIFLAG_URL_ENCODING_ENABLE_UTF8 使用 UTF8 发送 URL _UIFLAG_ENABLE_FORMS_AUTOCOMplete 允许表单自动

		NG_ENABLE_UTF8*/ _UIFLAG_ENABLE_FORMS_AUTOCOMPL ETE=@0x4000/*_UIFLAG_ENABLE_FO RMS_AUTOCOMplete*/ _UIFLAG_ENABLE_INPLACE_NAVIGAT ION=@0x10000/*_UIFLAG_ENABLE_I NPLACE_NAVIGATION*/ _UIFLAG_IME_ENABLE_RECONVERSIO N=@0x20000/*_UIFLAG_IME_ENABLE _RECONVERSION*/ _UIFLAG_THEME=@0x40000/*_UIFLA G_THEME*/ _UIFLAG_NOTHEME=@0x80000/*_UIF LAG_NOTHEME*/ _UIFLAG_NOPICS=@0x100000/*_UIF LAG_NOPICS*/ _UIFLAG_NO3DOUTERBORDER=@0x200 000/*_UIFLAG_NO3DOUTERBORDER*/ _UIFLAG_DISABLE_EDIT_NS_FIXUP= @0x400000/*_UIFLAG_DISABLE_EDI T_NS_FIXUP*/ _UIFLAG_LOCAL_MACHINE_ACCESS_C HECK=@0x800000/*_UIFLAG_LOCAL_ MACHINE_ACCESS_CHECK*/ _UIFLAG_DISABLE_UNTRUSTEDPROTO COL=@0x1000000/*_UIFLAG_DISABL E_UNTRUSTEDPROTOCOL*/ _UIFLAG_HOST_NAVIGATES=@0x2000 000/*_UIFLAG_HOST_NAVIGATES*/ _UIFLAG_ENABLE_REDIRECT_NOTIFI CATION=@0x4000000/*_UIFLAG_ENA BLE_REDIRECT_NOTIFICATION*/ _UIFLAG_USE_WINDOWLESS_SELECTC ONTROL=@0x8000000/*_UIFLAG_USE _WINDOWLESS_SELECTCONTROL*/ _UIFLAG_USE_WINDOWED_SELECTCON TROL=@0x10000000/*_UIFLAG_USE_ WINDOWED_SELECTCONTROL*/ _UIFLAG_ENABLE_ACTIVEX_INACTIV ATE_MODE=@0x20000000/*_UIFLAG_ ENABLE_ACTIVEX_INACTIVATE_MODE */ _UIFLAG_DPI_AWARE=@0x40000000/ *_UIFLAG_DPI_AWARE*/	完成 _UIFLAG_ENABLE_INPLACE_N AVIGATION 在点击邮件等 链接时, 打开相关应用程 序, 而不是新窗口 _UIFLAG_NOTHEME 使用主题 _UIFLAG_THEME 禁用主题 _UIFLAG_NOPICS 禁用内容 分级 _UIFLAG_DIV_BLOCKDEFAULT 编辑模式回车输入 div _UIFLAG_DISABLE_EDIT_NS_ FIXUP 编辑模式禁用名字 空间修正 _UIFLAG_LOCAL_MACHINE_AC CESS_CHECK 防止远程网页 导航到本地计算机 _UIFLAG_DISABLE_UNTRUSTE DPROTOCOL 禁止非信任协 议, 包含 ms-its, ms-itss, its, mk:@msitstore
DLCTL		_DLCTL_DLIMAGES=@0x10/*_DLCTL_ DLIMAGES*/ _DLCTL_VIDEOS=@0x20/*_DLCTL_VI DEOS*/	DLCTL_前缀的常量以控制下 载行为, 多个常量之间用位或 操作符() 连接。 _DLCTL_DLIMAGES 允许从服

	<p>_DLCTL_BGSOUNDS=@0x40/*_DLCTL_BGSOUNDS*/</p> <p>_DLCTL_NO_SCRIPTS=@0x80/*_DLCTL_NO_SCRIPTS*/</p> <p>_DLCTL_NO_JAVA=@0x100/*_DLCTL_NO_JAVA*/</p> <p>_DLCTL_NO_RUNACTIVEXCTLS=@0x200/*_DLCTL_NO_RUNACTIVEXCTLS*/</p> <p>_DLCTL_NO_DLACTIVEXCTLS=@0x400/*_DLCTL_NO_DLACTIVEXCTLS*/</p> <p>_DLCTL_DOWNLOADONLY=@0x800/*_DLCTL_DOWNLOADONLY*/</p> <p>_DLCTL_NO_FRAMEDOWNLOAD=@0x1000/*_DLCTL_NO_FRAMEDOWNLOAD*/</p> <p>_DLCTL_RESYNCHRONIZE=@0x2000/*_DLCTL_RESYNCHRONIZE*/</p> <p>_DLCTL_PRAGMA_NO_CACHE=@0x4000/*_DLCTL_PRAGMA_NO_CACHE*/</p> <p>_DLCTL_NO_BEHAVIORS=@0x8000/*_DLCTL_NO_BEHAVIORS*/</p> <p>_DLCTL_NO_METACHARSET=@0x10000/*_DLCTL_NO_METACHARSET*/</p> <p>_DLCTL_URL_ENCODING_DISABLE_UTF8=@0x20000/*_DLCTL_URL_ENCODING_DISABLE_UTF8*/</p> <p>_DLCTL_URL_ENCODING_ENABLE_UTF8=@0x40000/*_DLCTL_URL_ENCODING_ENABLE_UTF8*/</p> <p>_DLCTL_NOFRAMES=@0x80000/*_DLCTL_NOFRAMES*/</p> <p>_DLCTL_FORCEOFFLINE=@0x1000000/*_DLCTL_FORCEOFFLINE*/</p> <p>_DLCTL_NO_CLIENTPULL=@0x2000000/*_DLCTL_NO_CLIENTPULL*/</p> <p>_DLCTL_SILENT=@0x40000000/*_DLCTL_SILENT*/</p> <p>_DLCTL_OFFLINEIFNOTCONNECTED=@0x80000000/*_DLCTL_OFFLINEIFNOTCONNECTED*/</p> <p>_DLCTL_OFFLINE=@0x80000000/*_DLCTL_OFFLINE*/</p>	<p>务器下载图片, 如果指定了第三个参数, 未指定此标志, 则网页不下载任何图片.</p> <p>_DLCTL_VIDEOS 允许从服务器下载视频片断, 如果指定了第三个参数, 未指定此标志, 则网页不下载任何视频片断.</p> <p>_DLCTL_BGSOUNDS 允许播放文档指定的背景声音</p> <p>_DLCTL_NO_SCRIPTS web 窗体不执行任何页面脚本 (指 javascript 等)</p> <p>_DLCTL_NO_JAVA web 窗体不执行任何 Java applet</p> <p>_DLCTL_NO_RUNACTIVEXCTLS web 窗体不执行文档中的任何 ActiveX 控件;</p> <p>_DLCTL_NO_DLACTIVEXCTLS web 窗体不下载文档中的任何 ActiveX 控件;</p> <p>_DLCTL_DOWNLOADONLY web 窗体下载网页, 但不显示</p> <p>_DLCTL_NO_FRAMEDOWNLOAD web 窗体对包含框架的页面进行语法分析但不下载任何帧, 同时忽略框架,</p> <p>_DLCTL_RESYNCHRONIZE web 窗体忽略缓存中的数据并向服务器请求更新</p> <p>_DLCTL_PRAGMA_NO_CACHE 迫使请求发送给服务器并忽略代理 (这里一般指服务端缓存), 即使代理指明数据是最新的也是如此.</p> <p>_DLCTL_NO_METACHARSET 隐藏文档中的 META 元素指示的字符集;</p> <p>_DLCTL_URL_ENCODING_DISABLE_UTF8 禁止 UTF-8 编码</p> <p>_DLCTL_URL_ENCODING_ENABLE_UTF8 允许 UTF-8 编码</p> <p>_DLCTL_NOFRAMES 禁止框架</p> <p>_DLCTL_FORCEOFFLINE web 窗体工作在脱机方式</p> <p>_DLCTL_NO_CLIENTPULL web 窗体不执行任何客</p>
--	---	---

			户端的 pull 操作 _DLCTL_SILENT 组件对话框、脚本错误对话框静默模式 _DLCTL_OFFLINEIFNOTCONNECTED 如果未连接互联网，浏览器组件将以脱机方式工作
header		头部	
userAgent		代理	默认 null
crossDomain	bool	默认 true True false	
rightMenu	右键菜单	-1 不处理，控件原有状态 0/null 禁用菜单 1 自定义简单菜单	各个含义不同，请注意。 -1 不处理。默认有详细菜单。通常需要使用 flash 右键使用此值。对于 flash 右键功能请测试。
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	

4.3.3.6 webkit 内核打开网页

```
{
  "emit": "open",
  "Obj": "web",
  "AppType": 2,
  "pos": 1,
  "par": {
    "htmlStr": null,
    "HttpServer_startUrl": null,
    "URL": "http://www.appemit.com"
  },
  "par0": {
    "header": null,
    "userAgent": null,
    "crossDomain": true,
    "rightMenu": 1,
    "download": 1
  }
}
```

名称	设置	含义	说明
emit	open	必需。打开网页事件请求。	
Obj	web	必需。	
AppType	2	必需。 1 IE 内核 2 webkit 内核 3 blink 内核	有默认的右键菜单。 若为负数-2, 则是浮动窗口
pos	{ "left": 37 2, "top": 20 3, "width": 606, "height": 406 }	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器，自动识别的位置需要优化
par		必需。优先级别依次下降	三个参数必须有一个不是空。
htmlStr		Html 代码	
HttpServer_startUrl		以服务器形式打开本地 html 文件路径，可以是绝对或者相对路径。 /为分隔符。	
URL	http://www.appemit.com	支持网页地址或者本地 html 文件路径。	

	om 或者 /demo/html Demo/html. html		
Par0		可选。	
header		头部	
userAgent		代理	只写属性，不可读。
crossDomain	bool	默认 true True false	是否跨域
rightMenu	右键菜单	-1 不处理，控件原有状态 0/null 禁用菜单 1 自定义简单菜单	各个含义不同，请注意。 -1 不处理。默认没有菜单。通常需要使用 flash 右键使用此值。 对于 flash 右键功能请测试。
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	
download	1	默认 true true false	支持连接下载功能

4.3.3.7 blink 内核打开网页

```
{ "emit": "open", "Obj": "web", "AppType": 3, "pos": 1, "par": { "htmlStr": null, "HttpServer_startUrl": null, "URL": "http://www.appemit.com" }, "par0": { "header": null, "userAgent": null, "crossDomain": true, "rightMenu": 1, "download": 1 } }
```

名称	设置	含义	说明
emit	open	必需。打开网页事件请求。	
Obj	web	必需。	
AppType	3	必需。 1 IE 内核 2 webkit 内核 3 blink 内核	有默认的右键菜单。 若为负数-3, 则是浮动窗口
pos	{ "left": 372, "top": 203, "width": 606, "height": 406 }	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器，自动识别的位置需要优化
par		必需。优先级别依次下降	三个参数必须有一个不是空。
htmlStr		Html 代码	
HttpServer_startUrl		以服务器形式打开本地 html 文件路径，可以是绝对或者相对路径。 /为分隔符。	
URL	http://www.appemit.com	支持网页地址或者本地 html 文件路径。	

	om 或者 /demo/html Demo/html. html		
Par0		可选。	
header		头部	
userAgent		代理	可读写。默认 Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3729.169 Safari/537.36
crossDomain	bool	默认 true true false	
rightMenu	右键菜单	-1 不处理, 控件原有状态 0/null 禁用菜单 1 自定义简单菜单	各个含义不同, 请注意。 -1 不处理。默认有菜单。 通常需要使用 flash 右键 使用此值。 对于 flash 右键功能请测 试。
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	
download	1	默认 true true false	支持连接下载功能

4.3.3.8 打开 PDF 文件

打开阅读 PDF 文件, 支持本地和网络 PDF

```
{
  "emit": "open",
  "Obj": "web",
  "AppType": 4,
  "pos": 1,
  "par": {
    "URL": "https://cdn.jsdelivr.net/gh/appemit/appemit@master/docs/demo/AppEmit_help.pdf",
    "par0": {
      "rightMenu": 1
    }
  }
}
```

名称	设置	含义	说明
emit	open	必需。打开 PDF 事件请求。	
Obj	web	必需。	
AppType	4	必需。 1 IE 内核 2 webkit 内核 3 blink 内核 4 PDF 文件	有默认的右键菜单。 若为负数-4, 则是浮动窗 口
pos	{ "left": 372, "top": 203, "width": 606, "height": 406 }	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器, 自动识 别的位置需要优化

par		必需。	
URL	https://cdn.jsdelivr.net/gh/appeemit/master/docs/demo/AppEmit_help.pdf 或者 /demo/AppEmit_help.pdf	支持网页 pdf 地址, 或本地文件绝对或相对路径。	
Par0		可选。	
rightMenu	右键菜单	-1 不处理, 控件原有状态 0/null 禁用菜单 1 自定义简单菜单	各个含义不同, 请注意。
rightMenu	右键菜单	-1 不处理, 控件原有状态 0/null 禁用菜单 1 自定义简单菜单	各个含义不同, 请注意。 -1 不处理。默认有菜单。通常需要使用 flash 右键使用此值。 对于 flash 右键功能请测试。
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	

4.3.3.9 Office

创建、打开阅读、编辑 Office 文件, 支持本地和网络文件, 包括 word\excel\PPT 文件
更多 VBA 官方参考<https://docs.microsoft.com/zh-cn/office/vba/api/overview/word>
或者录制宏, 编辑参考代码。

```
{ "emit": "open", "Obj": "office", "AppType": 1, "src": "/demo/htmlDemo/file/d1.docx", "pos": 1, "par0": { "Titlebar": 0, "ProtectDoc": 0, "Menubar": 0, "Toolbars": 0, "btnFile": 0, "HttpPost": { "URL": "http://eu.httpbin.org/post", "valid": 0, "fileName": "s1.docx" }, "MNU_NEWBLANK": 0, "MNU_NEW": 0, "MNU_OPEN": 0, "MNU_CLOSE": 0, "MNU_SAVE": 0, "MNU_SAVEAS": 0, "MNU_PGSETUP": 0, "MNU_PRINT": 0, "MNU_PROPS": 0, "show_UpdateTool": true } }
```

名称	设置	含义	说明
emit	open	必需。打开 office 事件请求。	
Obj	office	必需。	
AppType	1	必需。 1 word 2 excel 3 powerpoint	有默认的右键菜单。 若为负数-1, 则是浮动窗口
src	字符串; path 为 [path, openPar]	path 为必需, 字符串网址或者本地相对或者绝对路径 , 或者数组(防止网页 src 里面没有 文件名称或者后缀名称需要修改,	Word 系列文件 Excel 系列文件 PPT 系列文件 不同应用的 openPar 不

		则使用 fileType)。 openPar: 具体不同。 "/demo/htmlDemo/file/x1.xls" ["demo/htmlDemo/file/x1.xls" , openPar] [[http://.../demo/htmlDemo/fil e/x1.xls, "x2.xlsx"], openPar]	同, 详细见后
pos	{ "left":37 2, "top":20 3, "width": 606, "heigh t":406 }	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器, 自动识别的位置需要优化
Par0		可选。	
progId	[]	默认为 MS office ["Word.Application", "Kwps. Appl ication", "WPS. Application"] ["Excel. Application", "Ket. Appl ication", "ET. Application"] ["PowerPoint. Application", "WPP . Application"]	采用不同自动化服务器 打开文档、表格、幻灯片, 按顺序依次尝试。
openPar			不同应用不一样, 详细见 后
Titlebar	0	默认 0 不显示标题栏 1 显示标题栏 0 不显示	
Menubar	1	默认 1 显示菜单栏 1 显示标题栏 0 不显示	
Statusbar	1	默认 1 显示状态栏 1 显示状态栏 0 不显示	
Protect	1		不同应用不一样, 详细见 后
Toolbars	1	默认 1 , 显示工具栏 1 显示 0 不显示	需要手动点击
btnFile	1	工具栏的文件 File 按钮 1 /true 默认显示 0	Toolbars 为真时有效
ViewType	1		不同应用不一样, 详细见 后
HttpPost	{ "URL": "ur l", "valid": 1, "method": " POST" "username"	URL 上传的地址, 必需 method 上传的方法 POST\GET, 默 认 POST cookie	如 果 上 传 , URL 必 需, valid=1 设为有效。

	<pre> :"name" "username" :"*" "cookie":null </pre>		
header		头部	以下参数用于下载上传使用
userAgent		代理	默认 null
cookies			
proxy			
proxyBypass			
flags			
crossDomain	bool	默认 true True false	
MNU_UPLOAD	0	默认 0，工具栏上传命令菜单 1 显示上传提交 0 不显示	Toolbars、btnFile、HttpPost["URL"] 同时为真时有效
MNU_NEWBLANK	0	默认 0，工具栏 FILE 命令菜单 1 显示新建空白 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_NEW	0	默认 0，工具栏 FILE 命令菜单 1 显示新建 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_OPEN	0	默认 0，工具栏 FILE 命令菜单 1 显示打开 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_CLOSE	0	默认 0，工具栏 FILE 命令菜单 1 显示关闭 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_SAVE	0	默认 0，工具栏 FILE 命令菜单 1 显示保存 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_SAVEAS	0	默认 0，工具栏 FILE 命令菜单 1 显示另存 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_PGSETUP	0	默认 0，工具栏 FILE 命令菜单 1 显示页面设置 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_PRINT	0	默认 0，工具栏 FILE 命令菜单 1 显示打印 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_PROPS	0	默认 0，工具栏 FILE 命令菜单 1 显示属性 0 不显示	Toolbars、btnFile 同时为真时有效
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口	

		0 不显示	
--	--	-------	--

4.3.3.9.1 Word 其它参数

更多 <https://docs.microsoft.com/zh-cn/office/vba/api/overview/word>

名称	设置	含义	说明																		
Par0		可选。																			
src	[path, 0, 1, 0]	ConfirmConversions=false, _ ReadOnly=false, AddToRecentFiles=false, PasswordDocument="", _ PasswordTemplate="", Revert=false, WritePasswordDocument="", _ WritePasswordTemplate="", Format=wdOpenFormatAuto, XMLTransform=""	按顺序																		
ViewType	1	wdMasterView 5 主控视图。 wdNormalView 1 普通视图。 wdOutlineView 2 大纲视图。 wdPrintPreview 4 打印预览视图。 wdPrintView 3 页面视图。 wdReadingView 7 阅读视图。 wdWebview 6 Web 视图。																			
Protect	[Type,NoReset,Password,UseIRM, EnforceStyleLock]	Type 必需 wdAllowOnlyComments=1 只允许将批注添加到文档中。 wdAllowOnlyFormFields=2 只允许通过窗体域将内容添加到文档中。 wdAllowOnlyReading=3 允许对文档进行只读访问。 wdAllowOnlyRevisions=0 只允许对现有内容进行修订。 wdNoProtection=-1 不对文档应用保护。 <table><tr><td>名称</td><td>必需/可选</td><td>说明</td></tr><tr><td>Type</td><td>必需</td><td>wdAllowOnlyComments=1 只允许将批注添加到文档中。 wdAllowOnlyFormFields=2 只允许通过窗体域将内容添加到文档中。 wdAllowOnlyReading=3 允许对文档进行只读访问。 wdAllowOnlyRevisions=0 只允许对现有内容进行修订。 wdNoProtection=-1 不对文档应用保护。</td></tr><tr><td>NoReset</td><td>可选</td><td>如果为 False , 则将窗体域重置为其默认值;如果为 True , 则在文档受保护时保留当前的窗体字段值。 如果 _Type_ 不是 wdAllowOnlyFormFields, 则 NoReset 将被忽略。</td></tr><tr><td>Password</td><td>可选</td><td>如果提供, 则密码能够编辑文档, 或者更改或删除保护。</td></tr><tr><td>UseIRM</td><td>可选</td><td>指定在保护文档不受更改时是否使用信息权限管理 (IRM)。</td></tr><tr><td>EnforceStyleLock</td><td>可选</td><td>指定是否对受保护的文档强制进行格式设置限制。</td></tr></table>	名称	必需/可选	说明	Type	必需	wdAllowOnlyComments=1 只允许将批注添加到文档中。 wdAllowOnlyFormFields=2 只允许通过窗体域将内容添加到文档中。 wdAllowOnlyReading=3 允许对文档进行只读访问。 wdAllowOnlyRevisions=0 只允许对现有内容进行修订。 wdNoProtection=-1 不对文档应用保护。	NoReset	可选	如果为 False , 则将窗体域重置为其默认值;如果为 True , 则在文档受保护时保留当前的窗体字段值。 如果 _Type_ 不是 wdAllowOnlyFormFields, 则 NoReset 将被忽略。	Password	可选	如果提供, 则密码能够编辑文档, 或者更改或删除保护。	UseIRM	可选	指定在保护文档不受更改时是否使用信息权限管理 (IRM)。	EnforceStyleLock	可选	指定是否对受保护的文档强制进行格式设置限制。	更改为自行在 officeJs 中控制。后面同 对非保护的文档保护。 网络文件, 不设置则默认为保护文档, [3,null,"random"], 产生随机密码, 密码不可获取, 请慎重处理。 不保护设置为[-1] 对本地文件不默认处理
名称	必需/可选	说明																			
Type	必需	wdAllowOnlyComments=1 只允许将批注添加到文档中。 wdAllowOnlyFormFields=2 只允许通过窗体域将内容添加到文档中。 wdAllowOnlyReading=3 允许对文档进行只读访问。 wdAllowOnlyRevisions=0 只允许对现有内容进行修订。 wdNoProtection=-1 不对文档应用保护。																			
NoReset	可选	如果为 False , 则将窗体域重置为其默认值;如果为 True , 则在文档受保护时保留当前的窗体字段值。 如果 _Type_ 不是 wdAllowOnlyFormFields, 则 NoReset 将被忽略。																			
Password	可选	如果提供, 则密码能够编辑文档, 或者更改或删除保护。																			
UseIRM	可选	指定在保护文档不受更改时是否使用信息权限管理 (IRM)。																			
EnforceStyleLock	可选	指定是否对受保护的文档强制进行格式设置限制。																			
Unprotect	pwd	文档解锁密码	对保护的文档 (type!= -1) 输入 解锁 密码 Unprotect 不保护。																		

4.3.3.9.2 Excel 其它参数

更多 <https://docs.microsoft.com/zh-cn/office/vba/api/overview/excel>

名称	设置	含义	说明
Par0		可选。	
src	[path, openPar]	(UpdateLinks、ReadOnly、Format、Password、WriteResPassword、IgnoreReadOnlyRecommended、原产地、定界符、可编辑、通知、转换器、AddToMru、Local、CorruptLoad)	["/demo/htmlDemo/file/xl.xlsx",null,1] [[https://cdn.jsdelivr.net/gh/appemit/appemit@master/docs/demo/file/xl.tmp,"xl1.xlsx"]] https://docs.microsoft.com/zh-cn/office/vba/api/excel.workbooks.open
ViewType	1	xlNormalView 1 正常。 xlPageBreakPreview 双面 分页预览。 xlPageLayoutView 第三章 页面视图	
Protect	数组[]	(Password、DrawingObjects、内容、场景、UserInterfaceOnly、AllowFormattingCells、AllowFormattingColumns、AllowFormattingRows、AllowInsertingColumns、AllowInsertingRows、AllowInsertingHyperlinks、AllowDeletingColumns、AllowDeletingRows、AllowSorting、AllowFiltering、AllowUsingPivotTables)	对所有已经已存工作表保护 注意 AllowFiltering 参数 https://docs.microsoft.com/zh-cn/office/vba/api/excel.worksheet.protect
Unprotect	pwd	文档解锁密码	输入解锁密码 Unprotect 不保护。

Activate_ID	1, 数值或者字符串	工作表的序号(左边第一为1), 或者名称. 默认打开的第几个工作表, 或者为该名称的工作表	暂未启用
-------------	------------	---	------

4.3.3.9.3 PowerPoint 其它参数

更多 <https://docs.microsoft.com/zh-cn/office/vba/api/overview/powerpoint>

名称	设置	含义	说明
Par0		可选。	
src	[path, openPar]	(ReadOnly、Untitled 无标题、 WithWindow)	https://docs.microsoft.com/zh-cn/office/vba/api/powerpoint.presentations.open
ViewType	1	ppViewHandoutMaster ' 讲义母版 4 ppViewMasterThumbnails ' 缩略图母版 12 ppViewNormal ' 普通视图 9 ppViewNotesMaster ' 备注母版 5 ppViewNotesPage ' 备注页 3 ppViewOutline ' 大纲视图 6 ppViewPrintPreview ' 打印预览 10 ppViewSlide ' 幻灯片视图 1 ppViewSlideMaster ' 幻灯片母版 2 ppViewSlideSorter ' 幻灯片浏览 7 ppViewThumbnails ' 缩略图视图 11 ppViewTitleMaster ' 标题母版 8	ppt ppViewType 类型

4.3.3.9.4 更多示例

1. 打开本地 txt, 只读, 不显示标题栏, 显示工具栏, 不上传

```
{ "emit": "open", "Obj": "office", "AppType": 1, "src": "/demo/htmlDemo/file/t1.txt",
"pos": 1, "par0": { "openPar": [null, true], "Titlebar": 0, "ProtectDoc": 0, "Menubar": 0,
"Toolbars": 0, "btnFile": 0, "HttpPost": { "URL": "http://eu.httpbin.org/post", "valid": 0,
"fileName": "s1.docx"}, "MNU_NEWBLANK": 0, "MNU_NEW": 0, "MNU_OPEN": 0, "MNU_CLOSE": 0,
"MNU_SAVE": 0, "MNU_SAVEAS": 0, "MNU_PGSETUP": 0, "MNU_PRINT": 0, "MNU_PROPS": 0, "show_UpdateTool": true }}
```

2. 打开网络 excel, 重命名, 不显示标题栏, 显示工具栏, 不上传

```
{ "emit": "open", "Obj": "office", "AppType": 2, "src": ["https://cdn.jsdelivr.net/g
h/appemit/appemit@master/docs/demo/file/x1.tmp", "a1.xlsx"], "pos": 1, "par0": { "Titlebar": 0, "Menubar": 1, "Toolbars": 1, "btnFile": 0, "HttpPost": { "URL": "http://eu.httpbin.org/post", "valid": 0 }, "MNU_NEWBLANK": 0, "MNU_NEW": 0, "MNU_OPEN": 0, "MNU_CLOSE": 0, "MNU_SAVE": 0, "MNU_SAVEAS": 0, "MNU_PGSETUP": 0, "MNU_PRINT": 0, "MNU_PROPS": 0, "show_UpdateTool": true }}
```

3. 打开网络 ppt, 重命名, 幻灯片视图, 不显示标题栏, 显示工具栏, 不上传

```
{ "emit": "open", "Obj": "office", "AppType": 3, "src": ["https://cdn.jsdelivr.net/g
```

h/appemit/appemit@master/docs/demo/file/pl.tmp", "pl.pptx"], "pos":1, "par0":{"View Type":1, "Titlebar":0, "Menubar":1, "Toolbars":1, "btnFile":0, "HttpPost":{"URL":"http://eu.httpbin.org/post", "valid":0}, "MNU_NEWBLANK":0, "MNU_NEW":0, "MNU_OPEN":0, "MNU_CLOSE":0, "MNU_SAVE":0, "MNU_SAVEAS":0, "MNU_PGSETUP":0, "MNU_PRINT":0, "MNU_PROPS":0, "show_UpdateTool":true}}

4.3.3.9.5 疑难

- 1. 要确保单独使用 word 打开，文件格式不报错，否则打不开。
- 2. 比如文档格式不对，需要重新另存打开，否则会出现等未知报错。
- 3. 选择保存时如果时间过长，会出现服务器正在运行中，需要在任务管理器关闭对应的 word\excel\ppt 的进程，重新启动。
- 4. Office 自动化初始化

Office 服务器 CLSID 项
Access.Application {73A4C9C1-D68D-11D0-98BF-00A0C90DC8D9}
Excel.Application {00024500-0000-0000-C000-0000000000046}
FrontPage.Application {04DF1015-7007-11D1-83BC-006097ABE675}
Outlook.Application {0006F03A-0000-0000-C000-0000000000046}
Application.Application {91493441-5A91-11CF-8700-00AA0060263B}
Word.Application {000209FF-0000-0000-C000-0000000000046}
在运行中
D:\PROGRA~1\MICROS~1\Office15\WINWORD.EXE /RegServer
检查 HKEY_CLASSES_ROOT\CLSID CLSID 项的 LocalServer32
D:\PROGRA~1\MICROS~1\Office15\EXCEL.EXE /automation
D:\PROGRA~1\MICROS~1\Office15\WINWORD.EXE /Automation
D:\PROGRA~1\MICROS~1\Office15\POWERPNT.EXE /AUTOMATION

4.3.3.10 Office javascript

4.3.3.10.1 嵌入网页 JS 控制

在前章节的基础上，增加一个 javascript 文件或者 html 文件，在 officeJs 中添加代码控制 office, 支持 word excel ppt。更多参考 VBA 和宏录制。

更多 <https://docs.microsoft.com/zh-cn/office/vba/api/overview/word>

{"emit":"open", "Obj":"office", "AppType":1, "src":"/demo/htmlDemo/file/d1.docx", "pos":1, "par":{"htmlStr":null, "HttpServer_startUrl":null, "URL":"/demo/htmlDemo/js/officeJs.html", "webJs":1}, "par0":{"Titlebar":0, "Menubar":0, "Toolbars":0, "btnFile":0, "Statusbar":0 }}

在 websocket 中打开 d1.docx, 实际在 par 的文件/demo/htmlDemo/ officeJs.html 中的 js 修改打开 d2.doc, 并且增加输入文字功能。

名称	设置	含义	说明
emit	open	必需。打开 office 事件请求。	
Obj	office	必需。	
AppType	1	必需。 1 word 2 excel 3 powerpoint	有默认的右键菜单。 若为负数-1, 则是浮动窗口
src	字符串; path 为 [path, openPar]	path 为必需, 字符串网址或者本地相对或者绝对路径, 或者数组 (防止网页 src 里面没有文件名称或者后缀名称需	Word 系列文件 Excel 系列文件 PPT 系列文件 不同应用的 openPar 不同, 详细

		要修改，则使用 fileType)。 openPar: 具体不同。 "/demo/htmlDemo/file/x1.xls" ["/demo/htmlDemo/file/x1.xls", openPar] [[http://.../demo/htmlDemo/file/x1.xls, "x2.xlsx"], openPar]	见后
par			下面三个参数必须有一个。
htmlStr		Html 代码	
HttpServer_startUrl		以服务器形式打开本地 html 文件路径，可以是绝对或者相对路径。/为分隔符。	
URL	http://www.appemit.com.../officeJs.html 或者 /demo/htmlDemo/js/officeJs.html	支持网页地址或者本地 html 文件路径。	
webJs	1	启用 par 的设置的嵌入网页 JS 1 IE 2 webkit 3 blink 0 关闭	或者设置为 0，把 officeJs 的 src 参数设置在 src 里面。使用 runCmd 执行输入命令。

在 officeJs.html 中也可以设置 src 参数。

webJs0	0	0 关闭本 js 的控制，即使 webJs 大于 0 注释后启用。	系统默认执行了 4 个过程接口 AppJs_init AppJs_loaded AppJs_destroy AppJs_closed
--------	---	-----------------------------------	--

➤ AppJs_init=function(AppType)

如果在浏览器网页中没有设置 src，可以在这里设置。

webJs0, 0 关闭本 js 的控制

objName:"word" 为程序对象名称，可不设置，自动根据 AppType 判断。如果修改了为 word1，则后面的 AppJsObject.word.Selection，应该修改为 AppJsObject.word1.Selection

➤ AppJs_loaded=function(AppType) {

//系统启动 APP，产生 AppJsObject.objName 对象后执行

定义了全局变量 AppJsObject.word
Selection=AppJsObject.word.Selection;Application=AppJsObject.word.Applicati
on;’ //注意 结尾一定要有; 或者\r\n

- AppJs_destroy=function() { //AppJsObject.objName 销毁前执行
- AppJs_closed=function(status) { //AppJsObject.objName 关闭后执行。已经
没有 AppJsObject.objName 对象了

4.3.3.10.2 浏览器网页执行代码命令控制

Var Req={"emit":"runCmd","Obj":"office","codeStr":
"Selection.TypeParagraph();Selection.TypeText("控制插入一句话-runCmd"); } "
EmitReq(Req);
通过前面 officeJs.html 定义的 AppJsObject.objName，可以直接控制 office。当然也可以
不需要 officeJs.html 定义，能在浏览器网页中 websocket 发送使用"emit":"runCmd"命令发
送 AppJs_init、AppJs_loaded，代码也可以。
可以添加 js 文件或者 html，在 html 还能动态控制 office，更多功能请联系我们。

4.3.3.11 Office DsoFramer

创建、打开阅读、编辑 Office 文件，支持本地和网络文件，包括 word\excel\PPT 文件

```
{ "emit": "open",  
  "Obj": "office",  
  "AppType": 8,  
  "src": "/demo/htmlDemo/file/xl.xls",  
  "pos": 1,  
  "par0": { "Caption": true,  
            "ProtectDoc": 0,  
            "Toolbars": true,  
            "btnFile": true,  
            "MNU_NEWBLANK": 0,  
            "MNU_NEW": 0,  
            "MNU_OPEN": 0,  
            "MNU_CLOSE": 0,  
            "MNU_SAVE": 0,  
            "MNU_SAVEAS": 0,  
            "MNU_PGSETUP": 0,  
            "MNU_PRINT": 0,  
            "MNU_PROPS": 0,  
            "show_UpdateTool": true }  
}
```

名称	设置	含义	说明
emit	open	必需。打开 office 事件请求。	
Obj	office	必需。	
AppType	8	必需。 8 DsoFramer	有默认的右键菜单。 若为负数-1, 则是浮动窗 口
src	字符串 src 网址或者本	src 为必需。	Word 系列文件 Excel 系列文件

	地相对或者绝对路径或者数组 [src, flag, ProgId , username, pwd , fileType]	flag:true/false ProgId: Word. Document Excel. Sheet PowerPoint. Show Visio.Drawing filename:temp.doc 等防止 src 里面没有文件名称或者后缀名称不对, 则使用 fileType; {"demo/htmlDemo/file/xl.xls", true, "Excel. Sheet", "name", "PWD"}	PPT 系列文件 fileType
pos	{"left":372, "top":203, "width":606, "height":406}	必需。 1 默认使用代码自动识别的位置。	对不同的浏览器, 自动识别的位置需要优化
Par0		可选。	
Caption	0	默认 false True 显示路径 string 显示具体 string 0/false/null	AppType 为负数时候显示标题栏
ProtectDoc	0	默认 0 , 保护文档 1 肯定, 解锁密码为 随机字符串, 不可获得。 String, 解锁密码为具体 String 0 可编辑文档	
Toolbars	1	默认 1 , 显示工具栏 1 显示 0 不显示	
btnFile	1	工具栏的文件 File 按钮 1 /true 默认显示 0	Toolbars 为真时有效
ShowView	1	wdNormalView = 1, wdOutlineView = 2, wdPrintView = 3, wdPrintPreview = 4, wdMasterView = 5, //这个是大纲 wdWebView = 6	
HttpPost	{"URL":"url", "method":"POST", "username":"name", "password":"password"}	URL 上传的地址, 必需 method 上传的方法 POST\GET, 默认 POST cookie	如果上传, URL 必需。

	: "*" "cookie": null}		
MNU_UPLOAD	0	默认 0，工具栏上传命令菜单 1 显示上传提交 0 不显示	Toolbars、btnFile、HttpPost["URL"] 同时为真时有效
MNU_NEWBLANK	0	默认 0，工具栏 FILE 命令菜单 1 显示新建空白 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_NEW	0	默认 0，工具栏 FILE 命令菜单 1 显示新建 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_OPEN	0	默认 0，工具栏 FILE 命令菜单 1 显示打开 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_CLOSE	0	默认 0，工具栏 FILE 命令菜单 1 显示关闭 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_SAVE	0	默认 0，工具栏 FILE 命令菜单 1 显示保存 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_SAVEAS	0	默认 0，工具栏 FILE 命令菜单 1 显示另存 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_PGSETUP	0	默认 0，工具栏 FILE 命令菜单 1 显示页面设置 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_PRINT	0	默认 0，工具栏 FILE 命令菜单 1 显示打印 0 不显示	Toolbars、btnFile 同时为真时有效
MNU_PROPS	0	默认 0，工具栏 FILE 命令菜单 1 显示属性 0 不显示	Toolbars、btnFile 同时为真时有效
show_UpdateTool	0	在配置文件设置默认值 0 1 显示更新文件下载窗口 0 不显示	

4.3.3.11.1 更多示例

4. 打开网络 docx，不显示工具栏，保护文档

```
{ "emit": "open", "Obj": "office", "AppType": 8, "src": ["https://cdn.jsdelivr.net/g/h/appemit/appemit@master/docs/demo/file/d1.tmp", false, "Word.Document", "userName", "PWD", "tmp.docx"], "pos": 1, "par0": { "Caption": true, "ProtectDoc": 0, "Toolbars": 0, "btnFile": true, "MNU_NEWBLANK": 0, "MNU_NEW": 0, "MNU_OPEN": 0, "MNU_CLOSE": 0, "MNU_SAVE": 0, "MNU_SAVEAS": 0, "MNU_PGSETUP": 0, "MNU_PRINT": 0, "MNU_PROPS": 0, "show_UpdateTool": true }}
```

5. 打开网络 xlsx，显示工具栏，显示文件按钮，上传文件(提交地址需要修改)

```
{ "emit": "open", "Obj": "office", "AppType": 8, "src": ["https://cdn.jsdelivr.net/g/h/appemit/appemit@master/docs/demo/file/x1.tmp", false, "Excel.Sheet", "userName", "
```

```
PWD", "a1.xlsx"], "pos":1, "par0":{"Caption":true, "ProtectDoc":0, "Toolbars":1, "btnFile":true, "HttpPost":{"URL":"http://eu.httpbin.org/post", "fileName":"s1.xlsx"}, "MNU_UPLOAD":1, "MNU_NEWBLANK":1, "MNU_NEW":0, "MNU_OPEN":0, "MNU_CLOSE":0, "MNU_SAVE":0, "MNU_SAVEAS":0, "MNU_PGSETUP":0, "MNU_PRINT":0, "MNU_PROPS":0, "show_UpdateTool":true}}
```

6. 打开本地 pptx, 显示工具栏, 不显示文件按钮

```
{"emit":"open", "Obj":"office", "AppType":8, "src":["/demo/htmlDemo/file/p1.pptx", false, "PowerPoint.Show", "name", "PWD"] , "pos":1, "par0":{"Caption":0, "ProtectDoc":1, "Toolbars":1, "btnFile":0, "MNU_NEWBLANK":0, "MNU_NEW":0, "MNU_OPEN":0, "MNU_CLOSE":0, "MNU_SAVE":0, "MNU_SAVEAS":0, "MNU_PGSETUP":0, "MNU_PRINT":0, "MNU_PROPS":0, "show_UpdateTool":true}}
```

4.3.3.11.2 其它参数

文档另存为

```
HRESULT SaveAs([in] VARIANT strFileName, [in] VARIANT dwFileFormat, [out,retval] long* pbool);
```

参数:

strFileName: 文件本地路径, 如 c:\\11.doc

dwFileFormat: 文件格式

Excel: Type enum XlFileFormat	Word: Type enum WdSaveFormat	PPT: enum PpSaveAsFileType
xlAddIn = 18 xlCSV = 6 xlCSVMac = 22 xlCSVMSDOS = 24 xlCSVWindows = 23 xlDBF2 = 7 xlDBF3 = 8 xlDBF4 = 11 xlDIF = 9 xlExcel2 = 16 xlExcel2FarEast = 27 xlExcel3 = 29 xlExcel4 = 33 xlExcel5 = 39 xlExcel7 = 39 xlExcel9795 = 43 xlExcel4Workbook = 35 xlIntlAddIn = 26 xlIntlMacro = 25 xlWorkbookNormal = -4143 xlSYLK = 2 xlTemplate = 17 xlCurrentPlatformText = -4158 xlTextMac = 19 xlTextMSDOS = 21 xlTextPrinter = 36	wdFormatDocument = 0 wdFormatTemplate = 1 wdFormatText = 2 wdFormatTextLineBreaks = 3 wdFormatDOSText = 4 wdFormatDOSTextLineBreaks = 5 wdFormatRTF = 6 wdFormatUnicodeText = 7 wdFormatEncodedText = 7 wdFormatHTML = 8	ppSaveAsPresentation = 1 ppSaveAsPowerPoint7 = 2 ppSaveAsPowerPoint4 = 3 ppSaveAsPowerPoint3 = 4 ppSaveAsTemplate = 5 ppSaveAsRTF = 6 ppSaveAsShow = 7 ppSaveAsAddIn = 8 ppSaveAsPowerPoint4FarEast = 10 ppSaveAsDefault = 11 ppSaveAsHTML = 12 ppSaveAsHTMLv3 = 13 ppSaveAsHTMLDual = 14 ppSaveAsMetaFile = 15 ppSaveAsGIF = 16 ppSaveAsJPG = 17 ppSaveAsPNG = 18 ppSaveAsBMP = 19

xlTextWindows = 20 xlWJ2WD1 = 14 xlWK1 = 5 xlWK1ALL = 31 xlWK1FMT = 30 xlWK3 = 15 xlWK4 = 38 xlWK3FM3 = 32 xlWKS = 4 xlWorks2FarEast = 28 xlWQ1 = 34 xlWJ3 = 40 xlWJ3FJ3 = 41 xlUnicodeText = 42 xlHtml = 44		
--	--	--

4.3.3.11.3 疑难

DsoFramer 是微软的开源控件，现在已经不更新，容易出现 bug。

比如文档格式不对，需要重新保存在打开，否则会出现两个窗口等未知报错。

选择保存时如果时间过长，会出现服务器正在运行中，需要在任务管理器关闭对应的 word\excel\ppt 的进程，重新启动。

4.3.3.12 第三方 DLL\组件调用

4.3.3.12.1 可视化 Flash ocx 调用

能在网页中执行命令，控制 flash, 举例见 demo/htmlDemo/dll.html

```
{"emit": "open", "Obj": "dll", "AppType": 1, "src": [], "pos": 1, "par": {"htmlStr": null, "HttpServer_startUrl": null, "URL": "/demo/htmlDemo/js/dllJs.html", "webJs": 1}}
```

调用 C:/Windows/SysWOW64/Macromed/Flash/Flash32_32_0_0_403.ocx 控件，需要自行安装 flash。

可以在 dllJs.html 进一步设置参数 AppType==1 情况，和过程调用代码。同时也能通过浏览器网页发布命令。

名称	设置	含义	说明
emit	open	必需。打开 dll 事件请求。	
Obj	dll	必需。	
AppType	1	必需。 根据自己的代码确定。	有默认的右键菜单。 若为负数-1, 则是浮动窗口
src	字符串数组	打开多个 DLL，只能一个可视化	
par			下面三个参数必须有一个。
htmlStr		Html 代码	
HttpServer_startUrl		以服务器形式打开本地 html 文件路径，可以是绝对或者相对路径。/为分隔符。	
URL	http://www.appemit.com.../dllJs.html 或者	支持网页地址或者本地 html 文件路径。	

	/demo/html Demo/js/ dllJs.html		
webJs	1	启用 par 的设置的嵌入网页 JS 1 IE 2 webkit 3 blink 0 关闭	或者设置为 0，把 dllJs 的 src 参数设置在 src 里面。 使用 runCmd 执行输入命令。
par0			
AppMethod	null	默认非必需。 POST。同步呼叫的控件，应该设置 为 POST，采取 websocket 相同的 异步通话	
在 dllJs.html 中也可以设置 src 参数。			
webJs0	0	0 关闭本 js 的控制，即使 webJs 大于 0 注释后启用。	系统默认执行了 4 个过程接口 AppJs_init AppJs_loaded AppJs_destroy AppJs_closed

4.3.3.12.1.1 嵌入网页 JS 控制

```

➤ AppJs_init=function(AppType)
如果在浏览器网页中没有设置 src，可以在这里设置。
webJs0, 0 关闭本 js 的控制
    createType:"createEmbed" // 可视化的控件
createEmbed(comCarrier,clsId,iid) , 只能有一个 , createObject(clsId,iid)
createInstance(clsId,itface) createUnknown(clsId,iid)
    ,comName:"flash"

//ocx

    ,objName:"shockwave"
    //,comCarrier:"custom" //
createEmbed 可视化 picture(图片控件设置)、static 、 winform 默认推荐 custom
//不同类型组件采取不要载体

    ,dllFile:"C:/Windows/SysWOW64/Macromed/Flash/Flash32_32_0_0_403.ocx"
// 必须 \\

    ,clsId : "D27CDB6E-AE6D-11CF-96B8-
444553540000" //guid

    ,iid:null
    ,itface:null
    ,embedObject:null

➤ AppJs_loaded=function(AppType) {
//系统启动 APP ,产生 AppJsObject. shockwave 对象后执行
AppJsObject.shockwave.Movie = "/demo/htmlDemo/file/test1.swf"

```

- AppJs_destroy=function() { //AppJsObject.shockwave 销毁前执行
- AppJs_closed=function(status) { //AppJsObject.shockwave 关闭后执行。已经没有了 AppJsObject.shockwave 对象了

4.3.3.12.1.2 浏览器网页执行代码命令控制

```

if (cmdId=="stop") {
    codeStr = txt2code(function() {/*
AppJsObject.shockwave.stop();
return {status="stop"}
*/});
    } else if(cmdId=="play") {
        codeStr = txt2code(function() {/*
AppJsObject.shockwave.play();
return {status="play"}
*/});
    }
}
Var Req={"emit":"runCmd","Obj":"dll","codeStr": codeStr}
EmitReq(Req);

```

通过前面 dllJs.html 定义的 AppJsObject.shockwave，可以直接控制 flash。当然也可以不需要 dllJs.html 定义，能在浏览器网页中 websocket 发送使用“emit”:“runCmd”命令发送 AppJs_init、AppJs_loaded，代码也可以。

4.3.3.12.2 报表控件 reportX 调用

能在网页中执行命令，控制 reportX，举例见 demo/htmlDemo/dll.html

```

{"emit":"open","Obj":"dll","AppType":2,"src":[],"pos":1,"par":{"htmlStr":null,"HttpServer_startUrl":null,"URL":"/demo/htmlDemo/js/dllJs.html","webJs":1},"par0":{"AppMethod":"msg Synchronization component need set: POST"}}

```

调用“/plugins/thirdparty/report/ReportX.ocx” 控件

可以在 dllJs.html 进一步设置参数 AppType==2 情况，和过程调用代码。

同时也能通过浏览器网页发布命令。

名称	设置	含义	说明
emit	open	必需。打开 dll 事件请求。	
Obj	dll	必需。	
AppType	2	必需。 根据自己的代码确定。	有默认的右键菜单。 若为负数-1，则是浮动窗口
src	字符串数组	打开多个 DLL，只能一个可视化	
par			下面三个参数必须有一个。
htmlStr		Html 代码	
HttpServer_startUrl		以服务器形式打开本地 html 文件路径，可以是绝对或者相对路径。/为分隔符。	

URL	http://www.appemit.com.../dllJs.html 或者 /demo/htmlDemo/js/dllJs.html	支持网页地址或者本地 html 文件路径。	
webJs	1	启用 par 的设置的嵌入网页 JS 1 IE 2 webkit 3 blink 0 关闭	或者设置为 0，把 dllJs 的 src 参数设置在 src 里面。 使用 runCmd 执行输入命令。
par0			
AppMethod	null	默认非必需。 POST。同步呼叫的控件，应该设置为 POST，采取 websocket 相同的异步通话	

在 dllJs.html 中也可以设置 src 参数。

webJs0	0	0 关闭本 js 的控制，即使 webJs 大于 0 注释后启用。	系统默认执行了 4 个过程接口 AppJs_init AppJs_loaded AppJs_destroy AppJs_closed
--------	---	-----------------------------------	--

4.3.3.12.2.1 嵌入网页 JS 控制

➤ AppJs_init=function(AppType)

如果在浏览器网页中没有设置 src，可以在这里设置。

```
src:[ {
// webJs0:0, //只
能设置为 0 关闭本 js 的控制；或者注释 为默认打开控制
createType:"createEmbed" //可视化的
控件 createEmbed(comCarrier,clsId,iid) , createObject(clsId,iid)
createInstance(clsId,itface) createUnknown(clsId,iid)
,comName:"reportXOCX"
,objName:"reportX"
// ,comCarrier:"custom" //
createEmbed 可视化 picture(图片控件设置)、static、winform 默认 推荐 custom
//不同类型组件采取不要载体

,dllFile:"/plugins/thirdparty/report/ReportX.ocx"
,clsId : "A5DA6E97-1D4C-4708-B705-
84A45716B4DD" //guid
,iid:null
```

```

        ,itface:null
        ,embedObject:null
    }    //暂时支持一个控件
]

```

- AppJs_loaded=function(AppType) {
//系统启动 APP ,产生 AppJsObject.reportX 对象后执行
- AppJs_destroy=function() { //AppJsObject.reportX 销毁前执行
- AppJs_closed=function(status) { //AppJsObject.reportX 关闭后执行。已经
没有 AppJsObject.reportX 对象了

4.3.3.12.2.2 浏览器网页执行代码命令控制

```

if(cmdId=="reportX_open"){
    codeStr =    txt2code(function() {/*
//AppJsObject.reportX.OpenReport(..io.fullpath("/demo/htmlDemo/file/rep1.rpxe"))
var  reportXPath=AppJs.dlgOpen(' rpxe|*.rpxe|所有文件|*.*|'|')
    AppJsVar_reportXPath=reportXPath;    //全局变量 ,没有 var
if  reportXPath  {AppJsObject.reportX.OpenReport(reportXPath)
    return {status="open reportX"; path=reportXPath}
}else {
    return null;
}
*/});
    }else if(cmdId=="reportX_save" ){
        //方案 1 通过 websocket 交互信息.
        /*
        if  ( !reportXPath) return;
        codeStr = ' AppJsObject.reportX.SaveReport("'" +reportXPath+' "') \
//通过 websocket 交互信息,或者直接采取全局变量
        return {status="save reportX"}' ;
        */

        // 方案 2 直接采取全局变量

        codeStr = ' AppJs.msgbox(AppJsVar_reportXPath);
AppJsObject.reportX.SaveReport(AppJsVar_reportXPath);'
        codeStr+= ' return {status="save reportX 2"}' ;

        // 方案 3 另存为

        //    codeStr = "var savePath=AppJs.dlgSave(' rpxe|*.rpxe|所有文件|*.*|'|',
另存为') if (savePath) AppJsObject.reportX.SaveReport(savePath);"

    }Var Req={"emit":"runCmd","Obj":"dll","codeStr": codeStr}

```


EmitReq(Req);

通过前面 dllJs.html 定义的 AppJsObject.reportX, 可以直接控制 reportX。当然也可以不需要 dllJs.html 定义, 能在浏览器网页中 websocket 发送使用“emit”:“runCmd”命令发送 AppJs_init、AppJs_loaded, 代码也可以。

4.3.3.12.3 第三方 dll 调用

1. dll_demo1.dll 安装在/plugins/thirdparty/dll_demo1.dll

能在网页中执行命令, 控制 dll_demo1, 举例见 demo/htmlDemo/dll.html

```
{ "emit": "open", "Obj": "dll", "AppType": 4, "src": [], "pos": 1, "par": { "htmlStr": null, "HttpServer_startUrl": null, "URL": "/demo/htmlDemo/js/dllJs.html", "webJs": 1, "AppShow": false }, "par0": { "AppMethod": "msg Synchronization component need set: POST" } }
```

可以在 dllJs.html 进一步设置参数 AppType==4 情况, 和过程调用代码。

同时也能通过浏览器网页发布命令。

名称	设置	含义	说明
emit	open	必需。打开 dll 事件请求。	
Obj	dll	必需。	
AppType	4	必需。 根据自己的代码确定。	有默认的右键菜单。 若为负数-1, 则是浮动窗口
src	字符串数组	打开多个 DLL, 只能一个可视化	
pos	0	无界面不需要	
par			
AppShow	false	必需。窗口不可见。 无界面必需设置 false。	
htmlStr		Html 代码	下面三个参数必须有一个。
HttpServer_startUrl		以服务器形式打开本地 html 文件路径, 可以是绝对或者相对路径。/为分隔符。	
URL	http://www.appemit.com.../dllJs.html 或者 /demo/htmlDemo/js/dllJs.html	支持网页地址或者本地 html 文件路径。	
webJs	1	启用 par 的设置 1 IE 2 webkit 3 blink 0 关闭	或者设置为 0, 把 dllJs 的 src 参数设置在 src 里面。 使用 runCmd 执行输入命令。
par0			
AppMethod	null	默认非必需。 POST。同步呼叫的控件, 应该设置为 POST, 采取 websocket 相同的异步通话	

在 dllJs.html 中也可以设置 src 参数。

webJs0	0	0 关闭本 js 的控制，即使 webJs 大于 0 注释后启用。	系统默认执行了 4 个过程接口 AppJs_init AppJs_loaded AppJs_destroy AppJs_closed
--------	---	-----------------------------------	--

4.3.3.12.3.1 嵌入网页 JS 控制

演示调用 dll 的 add 方法，主要是初始化后，声明 `dll.api("Add", "int(int a, int b)", "cdecl");` //必须是 cdecl 声明

➤ `AppJs_init=function(AppType)`

如果在浏览器网页中没有设置 src，可以在这里设置。

```
src:[ {  
    // webJs0:0, //只  
    能设置为 0 关闭本 js 的控制；或者注释 为默认打开控制  
    createType:"createDll" //  
    dll 调用  
    ,objName:"dll_demo1"  
    ,dllFile:"/plugins/private/50FCF891-1B93-  
4AE5-8A66-AB26A3C03378/dll_demo1.dll" // 必须 \\ / 放在  
/plugins/private/clsId/文件夹下面  
    ,clsId : "50FCF891-1B93-4AE5-8A66-  
AB26A3C03378" //guid 发送给 dll 信息  
    ,iid:null  
    ,data:'{"msg":"init from JS","info":1}'  
    //发送给 dll 数据  
    } //暂时支持一个控件  
]  
    ,AppShow:false //整体不可见 必须设  
置，默认可见  
  
// dll 接口__declspec(dllexport) int Add(int a, int b )  
//自行声明接口，再调用。  
  
AppJsObject.dll_demo1.myAdd =AppJsObject.dll_demo1.dll.api("Add", "int(int a, int  
b)", "cdecl"); //必须是 cdecl 声明  
  
var sum=AppJsObject.dll_demo1.myAdd(3, 5);  
    //AppJs.log(sum);  
AppJs.msgbox(sum);  
return {data=sum}; //结果在 external.out 中
```

4.3.3.12.3.2浏览器网页执行代码命令控制

```
//或者浏览器网页 发布命令 runCmd, 执行 AppDll_RevMsg 方法
codeStr= txt2code(function() {/      *
var sum=AppJsObject.dll_demo1.myAdd(3, 5);
return sum;
*   /});

var Req={"emit":"runCmd","Obj":"dll","codeStr":codeStr }
EmitReq(Req);
```

通过前面 dllJs.html 定义的 AppJsObject.dll_demo1，可以直接控制 dll_demo1。当然也可以不需要 dllJs.html 定义，能在浏览器网页中 websocket 发送使用“emit”:“runCmd”命令发送 AppJs_init、AppJs_loaded，代码也可以。

4.3.3.13 自行 DLL 开发调用

4.3.3.13.1 dll_demo1 调用

2. dll_demo1.dll 安 装 在 /plugins/private/50FCF891-1B93-4AE5-8A66-AB26A3C03378/dll_demo1.dll”

能在网页中执行命令，控制 dll_demo1, 举例见 demo/htmlDemo/dll.html

```
{"emit":"open","Obj":"dll","AppType":3,"src":[],"pos":0,"par":{"htmlStr":null,"HttpServer_startUrl":null,"URL":"/demo/htmlDemo/js/dllJs.html","webJs":1,"AppShow":false},"par0":{"AppMethod":"msg Synchronization component need set: POST" }}
```

可以在 dllJs.html 进一步设置参数 AppType==3 情况，和过程调用代码。

同时也能通过浏览器网页发布命令。

名称	设置	含义	说明
emit	open	必需。打开 dll 事件请求。	
Obj	dll	必需。	
AppType	3	必需。 根据自己的代码确定。	有默认的右键菜单。 若为负数-1, 则是浮动窗口
src	字符串数组	打开多个 DLL，只能一个可视化	
pos	0	无界面不需要	
par			
AppShow	false	必需。窗口不可见。 无界面必需设置 false。	
htmlStr		Html 代码	下面三个参数必须有一个。
HttpServer_startUrl		以服务器形式打开本地 html 文件路径，可以是绝对或者相对路径。/为分隔符。	
URL	http://www.appemit.com.../dllJs.html 或者 /demo/htmlDemo/js/	支持网页地址或者本地 html 文件路径。	

	dllJs.html		
webJs	1	启用 par 的设置的嵌入网页 JS 1 IE 2 webkit 3 blink 0 关闭	或者设置为 0，把 dllJs 的 src 参数设置在 src 里面。 使用 runCmd 执行输入命令。
par0			
AppMethod	null	默认非必需。 POST。同步呼叫的控件，应该设置为 POST，采取 websocket 相同的异步通话	

在 dllJs.html 中也可以设置 src 参数。

webJs0	0	0 关闭本 js 的控制，即使 webJs 大于 0 注释后启用。	系统默认执行了 4 个过程接口 AppJs_init AppJs_loaded AppJs_destroy AppJs_closed
--------	---	-----------------------------------	--

4.3.3.13.1.1 嵌入网页 JS 控制

```

➤ AppJs_init=function(AppType)
如果在浏览器网页中没有设置 src，可以在这里设置。
src:[ {
// webJs0:0, //只
能设置为 0 关闭本 js 的控制；或者注释为默认打开控制
createType:"createDll" //
dll 调用
, objName:"dll_demo1"
, dllFile:"/plugins/private/50FCF891-1B93-
4AE5-8A66-AB26A3C03378/dll_demo1.dll" // 必须 \\ / 放在
/plugins/private/clsId/文件夹下面
, clsId : "50FCF891-1B93-4AE5-8A66-
AB26A3C03378" //guid 发送给 dll 信息
, iid:null
, data:'{"msg":"init from JS","info":1}'
//发送给 dll 数据

} //暂时支持一个控件
]
, AppShow:false //整体不可见 必须设置，默认可见
➤ AppJs_loaded=function(AppType) {
//系统启动 APP，产生 AppJsObject. shockwave 对象后执行
➤ AppJsObject.dll_demo1.AppDll_loaded(AppDll_hwnd, "", {"msg":"loaded
from JS","info":2}');

```

```

➤ AppJs_destroy=function() { //AppJsObject. shockwave 销毁前执行
AppJsObject.dll_demo1.AppDll_destroy(AppDll_hwnd, "",' {"msg":"destroy" from
JS","info":3}');
➤ AppJs_closed=function(status) { //AppJsObject 关闭后执行。已

```

4.3.3.13.1.2 浏览器网页执行代码命令控制

```

codeStr= txt2code(function() {/*
AppJsObject.dll_demo1.AppDll_RevMsg(AppDll_hwnd, '',' {"data":"send" from
JS","info":3}');
* /});
var Req={"emit":"runCmd","Obj":"dll","codeStr":codeStr }
EmitReq(Req);

```

通过前面 dllJs.html 定义的 AppJsObject.dll_demo1，可以直接控制 dll_demo1。当然也可以不需要 dllJs.html 定义，能在浏览器网页中 websocket 发送使用“emit”:“runCmd”命令发送 AppJs_init、AppJs_loaded，代码也可以。

4.3.4 关闭 APP

4.3.4.1 关闭 sid 对应 APP

```
{"emit":"close","Obj":"flash"}
```

名称	设置	含义	说明
emit	close	必需。关闭控件 APP 通信事件请求。	
Obj	flash	必需。	

4.3.4.2 关闭 cid 所有 APP

```
{"emit":"closeAll","Obj":"flash"}
```

名称	设置	含义	说明
emit	close	必需。关闭所有控件 APP 通信事件请求。	关闭在 cid 下运行的所有控件 APP
Obj	flash	必需。	

4.3.5 窗口

4.3.5.1 显示

```
{"emit":"show","Obj":"flash"}
```

名称	设置	含义	说明
emit	show	必需。显示窗口请求。	
Obj	flash	必需。	
AppShow	true/null	默认 null 可视化 false 不可见	
pos		可选	
par0			
debounce	300	默认 300 毫秒	防抖函数，在限定的延 时内仅保留最后一次调用， 延时默认 300 毫秒

4.3.5.2 隐藏

```
{"emit": "hide", "Obj": "flash"}
```

名称	设置	含义	说明
emit	hide	必需。隐藏窗口请求。	
Obj	flash	必需。	

4.3.5.3 移动

```
{"emit": "move", "Obj": "flash", "pos": {"left": 10, "top": 6, "width": 206, "height": 106}}
```

名称	设置	含义	说明
emit	move	必需。移动窗口请求。	
Obj	flash	必需。	
pos	屏幕绝对位置	必需。	
par0			
debounce	300	默认 300 毫秒	防抖函数，在限定的延 时内仅保留最后一次调用， 延时默认 300 毫秒

4.4 获得参数

```
{"emit": "getPar", "Obj": "clientAuth"}
```

名称	设置	含义	说明
emit	getPar	必需。获得参数请求。	
Obj	clientAuth	是否授权 1 授权 0 无	

4.5 设置参数

```
{"emit": "setPar", "Obj": "flash", "topMost": true}
```

名称	设置	含义	说明
emit	getPar	必需。获得参数请求。	
Obj	flash	必需。	
topMost		必需。	

		True 置顶 False 取消置顶	
--	--	-----------------------	--

4.6 AppEmit 操作

4.6.1 错误信息

```
{"emit": "lasterr"}
```

名称	设置	含义	说明
emit	lasterr	必需。获得最近错误请求。	"

4.6.2 重启

```
{"emit": "restart", "Obj": "AppEmit"}
```

名称	设置	含义	说明
emit	restart	必需。AppEmit 重启请求。	重启后 client 需要重新连接。
Obj	AppEmit	必需。	

4.6.3 更新

```
{"emit": "update", "Obj": "AppEmit"}
```

名称	设置	含义	说明
emit	update	必需。询问 AppEmit 是否更新程序请求。	默认强制更新。如果 config.ini 里面设置 autoUpdate=0, 则询问更新。
Obj	AppEmit	必需。	

4.6.4 关于

```
{"emit": "about", "Obj": "AppEmit"}
```

名称	设置	含义	说明
emit	about	必需。获得关于请求。	返回 {"data": {"appName": "AppEmit", "url": "http://www.appemit.com/", "verDesc": "\u516C\u5171\u514D\u8D39\u7248 (Public free Version)", "verType": 0, "version": "0.3.5"}, "}
Obj	AppEmit	必需。	

4. 6. 5 版本信息

```
{"emit": "version", "Obj": "AppEmit"}
```

名称	设置	含义	说明
emit	version	必需。获得版本请求。	{"data": {"verDesc": "\u516C\u5171\u514D\u8D39\u7248\u672C (Public free Version)", "verType": 0, "version": "0.3.5"}, "
Obj	AppEmit	必需。	

5 问题

- 1. 支持 linux mac?
目前版本不支持，使用在 windows 系统上。
- 2. 免费版本有何限制条件?
每 80 次消息发送时有弹窗。
收费版本没有限制，包括支持局域网。
- 3. 测试点击连接，为何没有反应?
首先要打开 AppEmit.exe 服务，可以 F12 查看报错情况。重启系统后，AppEmit.exe 进程自动开启，没有被关闭。
- 4. 如何开发插件?
使用 HPSocket 的 C 接口。目前在测试中。