

AppEmit 中间件 说明书

| | | |
|-------|--|--|
| 程序名称 | AppEmit.exe | |
| 程序版本 | v1.0.6 | |
| 说明书版本 | v1.0.6 | |
| 网址 | http://www.appemit.com | |
| Email | appemit@appemit.com | |
| 修改时间 | 2020 年 9 月 8 日 | |
| 修订 | v1.0.6 | |

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) 在几乎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) 在浏览器中调用第三方 DLL、OCX 等 ActiveX 组件以及系统 winApi 函数，tcc、python、lua 等
- 4) 开发本地硬件 DLL 的插件，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
- 5) 各个应用程序之间通信，比如聊天
- 6) 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、硬件等企业客户在各种浏览器中开发调用 dll、ocx、ActiveX、com 组件、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@master/dist/AppEmit.zip>

github 下载地址

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

1.1 具体功能

支持同步、异步（默认）处理消息或者消息组

支持一个页面打开多个 APP

支持本地文件和网络文件访问

支持获取硬件信息，包括系统、CPU、主板、显卡、内存、硬盘、网络等

支持 USB 监控

支持 pcomm 串口异步多线程调用

支持用户间通信、发送接收消息

支持使用 IE、webkit、blink 内核打开，互动网页

支持打开、互动 flash, 包括 ActiveX flash 和 NPflash
 支持多媒体播放
 支持打开、编辑、代码互动 microsoft office、金山 office 的 world、excel、ppt, 支持本地文档和网络文档处理
 支持打开 PDF
 支持调用第三方 dll、com 等链接库
 支持开发 dll 等

1.2 使用条件

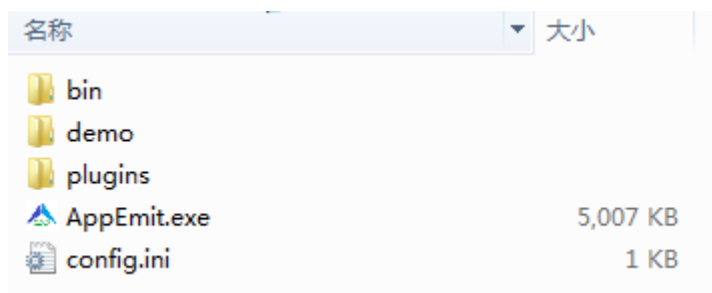
1.2.1 系统

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 AcitveX 控件, 打开 swf 文件 | 本地安装 flash AcitveX 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 | - | |

1.2.2 用法

在 windows 系统中, 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。

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 HPSocket (<https://github.com/ldcsaa/HP-Socket>), 支持 ssl。

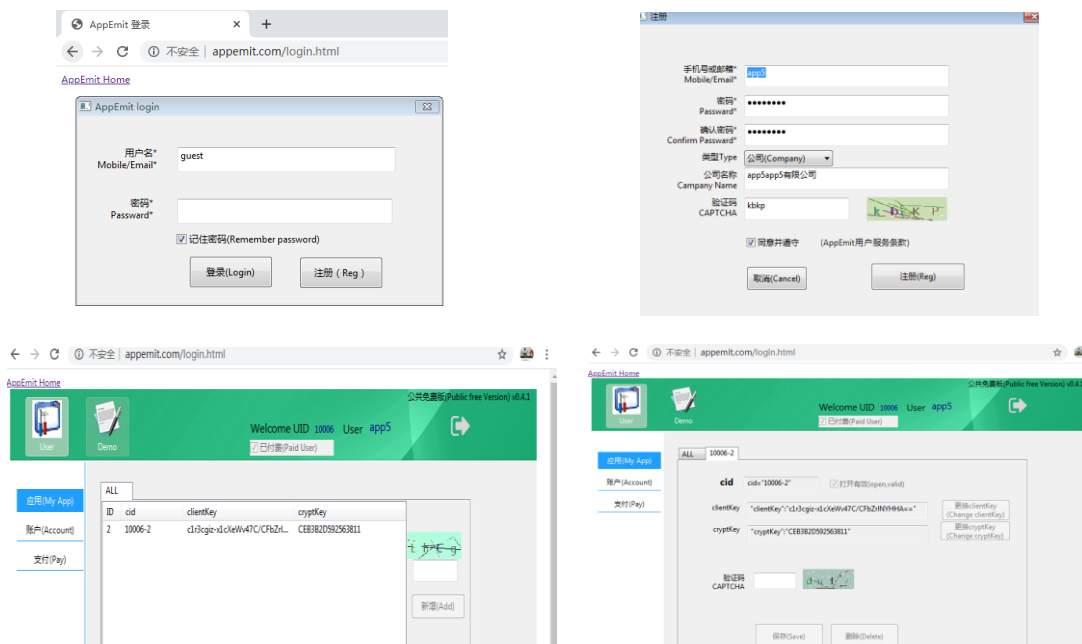
Dll 文件开放了 C 接口，可以在此基础二次开发控件, 主要调用

- HPSocket4C_U.dll
- HPSocket4C-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

AE_initAppWs("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 AE_initSet = {
    "emit": "init",
    "clientKey": "temp-0000000000", //
    "clientInfo": clientInfo,
    "wsUrl": wsUrl,
    // "flag": 0,
    // "sid": "123456", // 用户 session 或者用户
    // 名 ID, 唯一可以准确通话
    "gid": "[1,2]", // 用户群 ID, 一个用户可以加入多个群
    // "utf_escape": false, // 默认 false, 反馈的
```

data 编码转义

```
};  
AE_EmitReq_PIP(AE_initSet);  
4. 发送命令  
AE_OpenApp('{"emit":"hardWare","obj":"pc","par0":{"dev":["os","base"]}}')
```

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 示例参数

2.1 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。 | 依次尝试打开端口。若所有端口被占用 尝试关闭最后一个 port 的进程，打开最后一个端口 都可以在 config.in 修改 |
| | [443, 5124, 5125, 5126, 43100] | ssl 默认 443。 | 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 | |

2.2 初始化数据

设置授权信息等
{"emit":"init",...}

```
var AE_initSet = {  
    "emit":"init",  
    "clientKey": "temp-00000000000",    //
```

```

    "otp": "服务器动态生成",
    "clientInfo": clientInfo,
    "wsUrl": wsUrl,
    // "flag": 0,
    // "sid": "123456",           // 用户 session 或者用户名 ID, 唯一可
以准确通话
    "gid": "[1, 2]", // 用户群 ID, 一个用户可以加入多个群
    // "utf_escape": false,      // 默认 false, 反馈的 data 编码转义
};

```

| 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。 授权成功反馈 1 |
| OTP | | One-time password | 使用服务器 des 算法, 对 clientKey 的安全保护 授权成功反馈 2 |
| clientInfo | 对象 | 必需。使用浏览器。 默认 | |
| flag | 标识 | 默认为整体情况 0 生产环境 1 调试 2 压缩生产环境 3 调试压缩 | 同时可以在具体的一条命令中增加 flag 标识, 只处理本消息情况。比如在整体 flag=0 时, 发送带有 flag=1 的命令则此条命令的提示信息也会反馈, 但是压缩解压只能整体设置。 |
| 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 |
| utf_escape | false | 默认 false | 反馈的 data 编码转义 |

2.2.1 OTP

企业用户以上可以使用更安全的加密方法 OTP (One-time password)。OTP 是 AppEmit 对 clientKey 的安全保护措施, 安全可靠, 简单易用。将会有效杜绝他人通过在页面上获取 clientKey 的方式, 进行非法操作。

当在代码中 AppEmit 初始化时候, 传入一个新参数 OTP, 以 OTP 为验证, clientKey 此时可以不传递。

OTP 是一个在服务器端按 OTP 的算法规则来生成的加密字符串, AppEmit 将会对 OTP 进行合法性验证, 因为每个 OTP 只可以使用一次, 即使他人从页面代码中获得了 clientKey, 也无法推送或接收消息。

登录主页注册, 可以获得 clientKey 和 cryptKey。在服务器端 cryptKey 用于在生成 OTP 时, 作为加密的密钥。

2.2.2 在 server 端生成 OTP 规则

1. 声明一个字符串, 内容为"000"+网络 UTC 时间 (不是本地时间) 毫秒数
2. 将 cryptKey 作为密钥, 用 AES(CBC) 算法对字符串进行加密, 工作模式 CBC, 填充模式 PKCS5, 加密向量统一设为与密钥相同, 编码为 UTF-8
3. 使用 Base64 对加密结果进行编码, 考虑 URL 传递, 在此基础上做下一步变换

```

+   =>  -
/   =>  _
去除尾部的  =

```

其结果就是 OTP。

2.2.3 验证自己的 OTP 算法是否工作

cryptKey: 83AA400AF464C76D

UTC 毫秒数: 0001599193983184

OTP: cRwkend-1a00fBhEk0q2oDxADnN89uwqswaFR-Z6fHw

2.2.4 不同语言 AES 算法参考

不同编程语言中 AES 加解密结果要保持一致要注意以下一些要点：

- 1、工作模式 CBC，填充模式 PKCS5，不同语言要保持一致。
- 2、在下面的示例中，加密向量统一设为与密钥相同。
- 3、不同编程语言使用的文本编码要一致，同一个字符串，使用 UTF8 或 GBK 编码在内存中存储的实际数据可能是不一样的。默认编码为 UTF-8。
- 4、如果加密后返回的密文用了 BASE64 或 16 进制编码，那么在解密时同样也先做对应的逆向解码。

2.2.4.1 C#实现的 AES 加密、解密（CBC/PKCS5Padding）

```

using System;
using System.Text;
using System.Security.Cryptography;

namespace TestApp
{
    class Aes
    {
        static void Main(string[] args)
        {
            String str = "Test String";
            String encryptData = Aes.Encrypt(str, "83AA400AF464C76D",
"83AA400AF464C76D");
            Console.WriteLine(encryptData);

            String dstr = Aes.Decrypt("xL1eEwu9WCDRiscUbPPPSA==",
"83AA400AF464C76D", "83AA400AF464C76D");
            Console.WriteLine(dstr);
            Console.ReadKey();
        }

        public static string Encrypt(string toEncrypt, string key, string iv)
        {
            byte[] keyArray = UTF8Encoding.UTF8.GetBytes(key);
            byte[] ivArray = UTF8Encoding.UTF8.GetBytes(iv);
            byte[] toEncryptArray = UTF8Encoding.UTF8.GetBytes(toEncrypt);

```

```

        RijndaelManaged rm = new RijndaelManaged();
        rm.Key = keyArray;
        rm.IV = ivArray;
        rm.Mode = CipherMode.CBC;
        rm.Padding = PaddingMode.PKCS7;

        ICryptoTransform cTransform = rm.CreateEncryptor();
        byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0,
toEncryptArray.Length);
        return Convert.ToBase64String(resultArray, 0, resultArray.Length);
    }

    public static string Decrypt(string toDecrypt, string key, string iv)
    {
        byte[] keyArray = UTF8Encoding.UTF8.GetBytes(key);
        byte[] ivArray = UTF8Encoding.UTF8.GetBytes(iv);
        byte[] toEncryptArray = Convert.FromBase64String(toDecrypt);

        RijndaelManaged rm = new RijndaelManaged();
        rm.Key = keyArray;
        rm.IV = ivArray;
        rm.Mode = CipherMode.CBC;
        rm.Padding = PaddingMode.PKCS7;

        ICryptoTransform cTransform = rm.CreateDecryptor();
        byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0,
toEncryptArray.Length);
        return UTF8Encoding.UTF8.GetString(resultArray);
    }
}
}
}

```

2.2.4. 2PHP 实现 AES 加密、解密（CBC/PKCS5Padding）

//AES 加密

```

function aes_encrypt($encryptKey,$encryptStr) {
    $localIV = $encryptKey;
    $encryptKey = $encryptKey;

    $module = mcrypt_module_open(MCRYPT_RIJNDAEL_128, "", MCRYPT_MODE_CBC,
$localIV);
    mcrypt_generic_init($module, $encryptKey, $localIV);

    $block = mcrypt_get_block_size(MCRYPT_RIJNDAEL_128, MCRYPT_MODE_CBC);
    $pad = $block - (strlen($encryptStr) % $block);
    $encryptStr .= str_repeat(chr($pad), $pad);

    $encrypted = mcrypt_generic($module, $encryptStr);
}

```



```

        mdecrypt_generic_deinit($module);
        mdecrypt_module_close($module);

        return base64_encode($encrypted);

    }

//AES 解密
function aes_decrypt($encryptKey,$encryptStr) {
    $localIV = $encryptKey;
    $encryptKey = $encryptKey;

    $module = mdecrypt_module_open(MCRYPT_RIJNDAEL_128, "", MCRYPT_MODE_CBC,
    $localIV);
    mdecrypt_generic_init($module, $encryptKey, $localIV);

    $encryptedData = base64_decode($encryptStr);
    $encryptedData = mdecrypt_generic($module, $encryptedData);

    $e = ord($encryptedData[strlen($encryptedData)-1]);
    if($e<=16)$encryptedData=substr($encryptedData, 0,strlen($encryptedData)-$e);
    return $encryptedData;
}

$result = aes_encrypt("83AA400AF464C76D",'0001599193983184');
$decryptString = aes_decrypt("83AA400AF464C76D",$result);
echo $result;
echo $decryptString;

```

2. 2. 4. 3PHP7. 1 实现 AES 加密、解密（CBC/PKCS5Padding）

```

<?php

//AES 加密
function aes_encrypt($key,$str)
{
    return base64_encode( openssl_encrypt($str, 'AES-128-
    CBC',$key,OPENSSL_RAW_DATA,$key) );;
}

//AES 解密
function aes_decrypt($key,$str)
{
    return openssl_decrypt(base64_decode($str), 'AES-128-CBC', $key,
    OPENSSL_RAW_DATA, $key);
}

```

```

$result = aes_encrypt("83AA400AF464C76D","0001599193983184");
$str = aes_decrypt("83AA400AF464C76D",$result);

echo $result."<br>";
echo $str;
?>

```

2. 2. 4. 4Java 实现的 AES 加密、解密代码（CBC/PKCS5Padding）

```

import javax.crypto.spec.IvParameterSpec;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;

public class AESCrypt {

    public static String encrypt(String source, String key) throws Exception {

        byte[] sourceBytes = source.getBytes("UTF-8");
        byte[] keyBytes = key.getBytes("UTF-8");

        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        IvParameterSpec ivParameterSpec = new IvParameterSpec(keyBytes);
        cipher.init(Cipher.ENCRYPT_MODE, new SecretKeySpec(keyBytes,
"AES"),ivParameterSpec);

        byte[] decrypted = cipher.doFinal(sourceBytes);
        return new sun.misc.BASE64Encoder().encode(decrypted);
    }

    public static String decrypt(String encryptStr, String key) throws Exception {
        byte[] sourceBytes = new sun.misc.BASE64Decoder().decodeBuffer(encryptStr);
        byte[] keyBytes = key.getBytes("UTF-8");

        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        IvParameterSpec ivParameterSpec = new IvParameterSpec(keyBytes);
        cipher.init(Cipher.DECRYPT_MODE, new SecretKeySpec(keyBytes,
"AES"),ivParameterSpec);

        byte[] decoded = cipher.doFinal(sourceBytes);
        return new String(decoded, "UTF-8");
    }
}

```

}

2.3 命令

说明，参数分组为 {a, par: {b}, par0: {c}}
abc 里面的字段名称都要不同。

参数形式如下

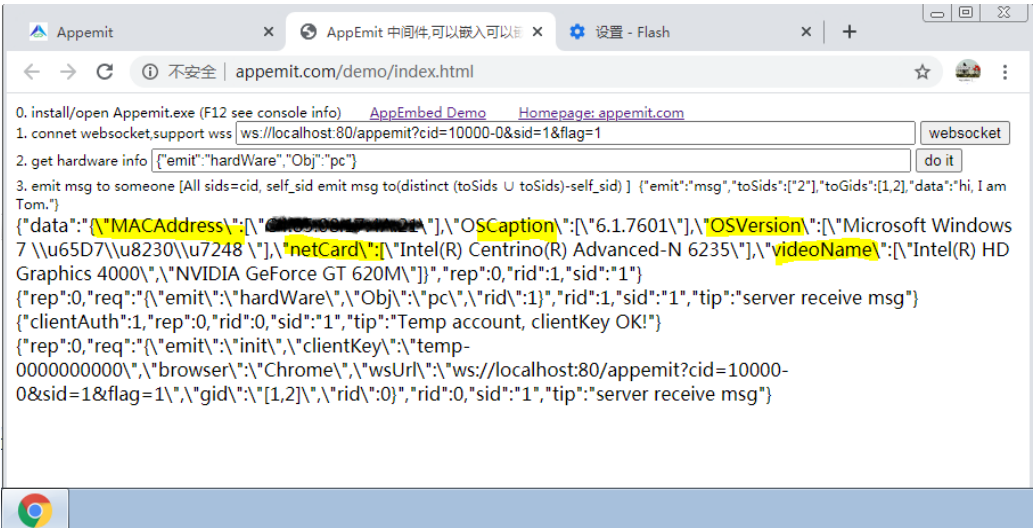
| 名称 | 设置 | 含义 | 说明 |
|-------------|-------------------------------------|--|--------------------------------------|
| emit | | init 初始化 open 打开 App runCmd 互动调用 App 运行命令 close closeAll 关闭 App msg 发送消息 hardware 获取硬件信息 lasterr | |
| Obj | | pc | |
| AppId | 1 | 1, 2, 3 | 一个页面一个 APP，默认为 1。 |
| AppType | 1, -1 | 1, 2, 3, 4... -1, -2, -3, -4 | 正数嵌入 负数浮动 |
| pos | 1, -1 [left, top, width, height] | 位置 1 更新位置 -1 删除位置数据 [left, top, width, height] 保留 | |
| data | | | |
| src | | | |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时，App 随浏览器移动变形隐藏显示等事件 |
| AppStatus | 1/0 | 系统自动反馈 1 打开 0 关闭 | 接收的数据 |
| par | | 必填参数 | |
| objName | | | |
| ... | | | |
| par0 | | 可填参数 | |
| header | | | |
| userAgent | | | |
| crossDomain | | | |
| ... | | | |

2.3.1 硬件

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。

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

```
AE_OpenApp('{"emit":"hardWare","Obj":"pc","par0":{"dev":["os","base"]}}')
```



2.3.1.1 硬件信息

```
{"emit":"hardWare","Obj":"pc","par0":{"dev":["device","os","display","cpu","net","memory","storage","base"]}}
```

| 名称 | 设置 | 含义 | 说明 |
|------|--------------------------------------|--|--|
| emit | hardWare | 必需。通信请求。 | |
| Obj | pc | 必需。目标对象。 | |
| par0 | | | |
| dev | 字符串 "base" 或者 数组 ["device"] | 非必需 ["device","os","display","cpu","net","memory","storage","base"], 取某个值都即可。 为空时默认设置为"base" | "base"可获取多个设备，但是对于 windows 精简版本可能无法获取信息。其它字段获取为第一个设备。 免费版本仅支持"base" |
| rid | 整数 | 必需。表示本次具体请求标号。 | 如果没有设置，则每次发送命令自动加 1。 后续省略描述。 |

2.3.1.2 串口 Pcomm

方法 1 使用 IE 控件打开网页最简单

方法 2 调用接口直接访问串口。下面为方法 2 的示例。

2.3.1.2.1 串口控件 Pcomm

在网页中调用 Pcomm.dll 的串口控件，支持多窗口同时异步跨线程调用串口。

```
{"emit":"open","Obj":"comm","AppType":1,"AppShow":true,"pos":1,"par":{"objName":"Pcomm1","Port":"com1","BaudRate":"9600","ByteSize":"8","Parity":"none","StopBits":"1"},"par0":{"Send_AsciiHex":"Ascii","Recv_AsciiHex":"Ascii","BaudRate_list":["50","75","110","134","150","300","600","1200","1800","2400","4800","7200",
```

```
"9600","19200","38400","57600","115200","230400","460800","921600"],"ByteSize_list":["5","6","7","8"],"Parity_list":["none","even","odd","spc","mrk"],"StopBits_list":["1","1.5","2"]}]}
```

| 名称 | 设置 | 含义 | 说明 |
|---------------|---------------|--------------------------------|-----------------------------|
| emit | open | 必需。打开串口事件请求。 | |
| Obj | comm | 必需。 | |
| AppId | 键值 | 必需。未定义时默认为 1 | 一个页面打开多个应用，应用的 AppId 必须不同。 |
| AppType | 1 | 必需。 1 Pcomm 2 comm | 有默认的右键菜单。 若为负数-1, 则是浮动窗口 |
| pos | 1 | 0 无界面不需要 1 有界面，位置自动识别 | |
| data | | 非必需。 | 启动后发送给串口的数据 |
| AppShow | false | 必需。空时默认窗口不可见。 有界面必需设置 true。 | |
| AppStatus | 1/0 | 系统自动反馈 1 打开 0 关闭 | 接收的数据 |
| par | | | |
| objName | Pcomm1 字符串 | 字符串变量， 字母开头、数字、 下划线、 | 用来在 js 调用变量 |
| Port | com1/1 | | |
| par0 | | | |
| Send_AsciiHex | Ascii/Hex | Ascii Hex | 全局写入方式 |
| Recv_AsciiHex | Ascii/Hex | Ascii Hex | 全局接收格式 |
| toFormat | Ascii/Hex | Ascii Hex | 对本次发送的 data 转换格式。 |
| BaudRate | 9600 | | |
| ByteSize | 8 | | |
| Parity | none | | |
| StopBits | 1 | | |
| | | | |
| | | | |

2.3.1.2.2 串口发送接收数据

自动开启 COM1 隐藏界面接收数据

```
{"emit":"open","Obj":"comm","AppType":1,"AppShow":0,"pos":1,"par":{"objName":"Pcomm1","Port":"COM1","BaudRate":"9600","ByteSize":"8","Parity":"none","StopBits":"1"},"par0":{"AppMethod":"POST","Send_AsciiHex":"Ascii","Recv_AsciiHex":"Ascii"}}
```

对 COM1 端口发送数据

```
{ "emit": "send", "Obj": "comm", "AppType": 1, "AppId": 1, "data": "可以, 第 1, from  
com1 ", "par0": { "Send_AsciiHex": "Hex", "toFormat": "Hex" } }
```

| 名称 | 设置 | 含义 | 说明 |
|---------------|-----------|--------------------------------|---|
| emit | send | 必需。串口发送事件请求。 | |
| Obj | comm | 必需。 | |
| AppId | 键值 | 必需。未定义时默认为 1 | 一个页面打开多个应用, 应用的 AppId 必须不同。 |
| AppType | 1 | 必需。 1 Pcomm 2 comm | 有默认的右键菜单。 若为负数-1, 则是浮动窗口 |
| pos | 1 | 0 无界面不需要 1 有界面, 位置自动识别 | |
| data | | 非必需。 | 启动后发送给串口的数据 |
| AppShow | false | 必需。空时默认窗口不可见。 有界面必需设置 true。 | |
| data | | 必需。发送数据内容 | |
| par | | | |
| objName | 字符串 | 字符串变量, 字母开头、数字、 下划线、 | 用来在 js 调用变量 |
| par0 | | | |
| Send_AsciiHex | Ascii/Hex | Ascii Hex | 本次写入方式。此次 send 命令 为空则默认为 open 事件中的 Send_AsciiHex |
| toFormat | Ascii/Hex | Ascii Hex | 对发送的 data 转换格式。此次 send 命令为空则默认为 open 事 件中的 toFormat |

2.3.1.2.3 互动调用串口其它命令

调用 runCmd 命令来操作 App 对象。

```
{ "emit": "runCmd", "Obj": "comm", "AppType": 1, "AppId": 1, "codeStr": "AppJsObject.P  
comml.write('动态写入 ascii 数据 1')"} }
```

下面是调用的接口函数, 和原含有略有调整。

open(__) = 修改端口号并打开端口, 成功返回 true, \n 失败返回 null, 错误信息, 错误代码

close() = 关闭串口\n 这个函数不会被析构函数自动调用, \n 应在确认不再使用时调用此函数关闭串口

getch() = 读取一个字节码\n 失败返回 null, 错误信息

putch() = 发送一个字节码\n 成功返回长度, 失败返回 null, 错误信息

read() = 读取数据, 可选指定读取缓冲区长度\n 成功返回数据, 失败返回 null, 错误信息
\n 如果没有指定长度且没有接收到数据返回 null

readBuffer(. (缓冲区, 读取长度) = 读取数据到 buffer 缓冲区, 读取长度可省略\n 成功返回读取长度, 失败返回 null, 错误信息

readHex() = 读取数据并以十六进制编码显示\n 成功返回数据, 失败返回 null, 错误信息

write(. (数据, 长度) = 写入数据\n 不指定长度时自动获取数据长度\n 成功返回写入长度, 失败返回 null, 错误信息

writeHex(. (数据) = 写入十六进制编码数据
 getBaudRate() = 返回串口波特率\n 失败返回 null, 错误信息
 setBaudRate(. (波特率) = 设置串口波特率\n 失败返回 null, 错误信息
 getMode() = 返回串口的工作模式\n 返回值为三个, 分别为: 数据位, 停止位, 校验位\n 失败返回 null, 错误信息
 ioctl(. (波特率, 数据位, 停止位, 校验位) = 设置串口的工作模式\n 波特率为数值, 省略时使用默认值 9600\n 数据位为数值, 可选值为 5, 6, 7, 8, 省略时默认值为 8\n 停止位可选值为 1, 2, 不指定时默认为 1\n 校验位使用字符串值指定, 可选值为 "even", "odd", "spc", "mrk", 可省略以设置空校验位
 flush(. (接收缓冲区, 发送缓冲区) = 清除接收、发送缓冲区\n 对应参数为 true 清除该缓冲区\n 无参数时清除接收缓冲区
 lctrl(. (dtr, rts) = 设置串口 RTS/DTS, 参数使用布尔值
 iqueue() = 返回输入缓冲区字符长度\n 失败返回 null, 错误信息
 oqueue() = 返回发送缓冲区中剩余的数据长度
 getLineStatus() = 获取串口的 CTS, DST, DCD, RI 线的状态
 setFlow(. (CTS 硬流控, RTS 硬流控, TX 软流控, RX 软流控) = 串口流控
 getFlow() = 返回 4 个值, 分别表示 CTS 硬流控, RTS 硬流控, TX 软流控, RX 软流控是否开启
 dataStatus() = 检查接收数据时是否遇到错误\n 0 表示无错误, 小于 0 表示函数执行错误, 大于 0 时各二进制位作用:\n bit 0 on - parity error\n bit 1 on - framing error\n bit 2 on - overrun error\n bit 3 on - overflow error
 abortRead() = 强制中止 read, getch 等读数据函数
 getReadTimeouts() = 返回读数据总超时, 间隔超时两个值, 单位毫秒
 setReadTimeouts(. (总超时, 间隔超时) = 设置读数据超时, 单位毫秒
 abortWrite() = 强制中止 Write, putch 等写数据函数
 getWriteTimeouts() = 返回写数据总超时, 间隔超时两个值, 单位毫秒
 setWriteTimeouts(. (总超时, 间隔超时) = 设置写数据超时, 单位毫秒
 termIrqThread(指定字节码, 回调函数, owner 对象) = @.termIrqThread(__/*线程接收到指定字节码时响应事件*/, function(port) {\n var sport = sio.port(port);\n })
 termCntIrqThread(指定长度, 回调函数, owner 对象) = @.termCntIrqThread(__/*接收到指定个字节时响应事件*/, function(port) {\n var sport = sio.port(port);\n })
 modemIrqThread(回调函数, owner 对象) = @.modemIrqThread(function(port) {\n var sport = sio.port(port); __/*当硬件线路 (CTS, DSR, CD, RI) 的电压发生变化时触发事件*/\n })
 breakIrqThread(回调函数, owner 对象) = @.breakIrqThread(function(port) {\n var sport = sio.port(port); __/*接收到中断信号时触发事件*/\n })
 txEmptyIrqThread(回调函数, owner 对象) = @.txEmptyIrqThread(function(port) {\n var sport = sio.port(port); __/*输出缓冲区最后一个字符发送后触发此事件*/\n })
 transmitAscii(文件路径, 进度回调函数) = @.transmitAscii("文件路径", \n function(length, bufferSize, buffer, totalLength) {\n __/* 传输文件协议: ASCII\n length 为当前传输大小, totalLength 为总大小, 其他参数仅用于调试*/\n })\n
 receiveAscii(超时秒数, 文件路径, 进度回调函数) = @.transmitAscii("文件路径", \n function(length, bufferSize, buffer, totalLength) {\n __/* 接收文件协议: ASCII\n length 为当前传输大小, totalLength 为总大小, 其他参数仅用于调试*/\n })\n
 transmitKermit(文件路径, 进度回调函数) = @.transmitKermit("文件路径", \n function(length, bufferSize, buffer, totalLength) {\n __/* 传输文件协议: Kermit\n length 为当前传输大小, totalLength 为总大小, 其他参数仅用于调试*/\n })\n
 receiveKermit(超时秒数, 文件路径, 进度回调函数) = @.receiveKermit("文件路径", \n

```

function(length, bufferSize, buffer, totalLength) {\n    __/* 接收文件协
议:Kermit\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于调试*/\n }\n)
transmitYmodem(文件路径, 进度回调函数)= @.transmitYmodem("文件路径", \n
function(length, bufferSize, buffer, totalLength) {\n    __/* 传输文件协
议:Ymodem\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于调试*/\n }\n)
receiveYmodem(超时秒数, 文件路径, 进度回调函数)= @.receiveYmodem("文件路径", \n
function(length, bufferSize, buffer, totalLength) {\n    __/* 接收文件协
议:Ymodem\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于调试*/\n }\n)
transmitZmodem(文件路径, 进度回调函数)= @.transmitZmodem("文件路径", \n
function(length, bufferSize, buffer, totalLength) {\n    __/* 传输文件协
议:Zmodem\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于调试*/\n }\n)
receiveZmodem(超时秒数, 文件路径, 进度回调函数)= @.receiveZmodem("文件路径", \n
function(length, bufferSize, buffer, totalLength) {\n    __/* 接收文件协
议:Zmodem\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于调试*/\n }\n)
transmitXmodem1kCrc(文件路径, 进度回调函数)= @.transmitXmodem1kCrc("文件路径
", \n function(length, bufferSize, buffer, totalLength) {\n    __/* 传输文件协
议:XMODEM, 1K block size, 16 bit CRC\nlength为当前传输大小, totalLength为总大小,
其他参数仅用于调试*/\n }\n)
receiveXmodem1kCrc(超时秒数, 文件路径, 进度回调函数)= @.receiveXmodem1kCrc("文
件路径", \n function(length, bufferSize, buffer, totalLength) {\n    __/* 接收文件
协议:XMODEM, 1K block size, 16 bit CRC\nlength为当前传输大小, totalLength为总大
小, 其他参数仅用于调试*/\n }\n)
transmitXmodemChecksum(文件路径, 进度回调函数)= @.transmitXmodemChecksum("文件
路径", \n function(length, bufferSize, buffer, totalLength) {\n    __/* 传输文件
协议:XMODEM, CHECKSUM\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于调
试*/\n }\n)
receiveXmodemChecksum( 超 时 秒 数 , 文 件 路 径 , 进 度 回 调 函 数 )=
@.receiveXmodemChecksum(" 文 件 路 径 ", \n
function(length, bufferSize, buffer, totalLength) {\n    __/* 接收文件协
议:XMODEM, CHECKSUM\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于调试
*/\n }\n)
transmitXmodemCrc(文件路径, 进度回调函数)= @.transmitXmodemCrc("文件路径", \n
function(length, bufferSize, buffer, totalLength) {\n    __/* 传输文件协
议:XMODEM, 16 bit CRC\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于
调试*/\n }\n)
receiveXmodemCrc(超时秒数, 文件路径, 进度回调函数)= @.receiveXmodemCrc("文件路
径", \n function(length, bufferSize, buffer, totalLength) {\n    __/* 接收文件协
议:XMODEM, 16 bit CRC\nlength为当前传输大小, totalLength为总大小, 其他参数仅用于
调试*/\n }\n)

```

2.3.1.3 串口 mscomm32 控件

comm32 效能略低, 新版的 MSComm32. OCX 中存在一个影响传输二进制数据的 Bug。

使用 IE 控件打开网页最简单。

```

{"emit": "open", "Obj": "comm", "AppType": 2, "AppShow": false, "par": {"objName": "ms
comm1", "Port": "COM1", "Settings": "9600, N, 8, 1"}, "par0": {"Recv_AsciiHex": "Ascii"}}
发送数据
{"emit": "send", "Obj": "comm", "AppType": 2, "AppId": 1, "data": "from mscomm1"}
{"emit": "runCmd", "Obj": "comm", "AppType": 2, "AppId": 1, "codeStr": "AppJsObject.m

```



```
scomm1.output='mscomm1 write ascii'"}

```

| 名称 | 设置 | 含义 | 说明 |
|-------------------|-----------------|--------------------------------|-----------------------------|
| emit | open | 必需。打开串口事件请求。 | |
| Obj | comm | 必需。 | |
| AppId | 键值 | 必需。未定义时默认为 1 | 一个页面打开多个应用，应用的 AppId 必须不同。 |
| AppType | 2 | 必需。 1 Pcomm 2 comm | 有默认的右键菜单。 若为负数-1, 则是浮动窗口 |
| pos | 1 | 0 无界面不需要 1 有界面，位置自动识别 | |
| data | | 非必需。 | 启动后发送给串口的数据 |
| AppShow | false | 必需。空时默认窗口不可见。 有界面必需设置 true。 | |
| AppStatus | 1/0 | 系统自动反馈 1 打开 0 关闭 | 接收的数据 |
| par | | | |
| objName | mscomm1 | 字符串变量，字母开头、数字、下划线、 | 用来在 js 调用变量 |
| CommPort | com1/1 | 设置串口 | |
| Settings | "9600, N, 8, 1" | 设置波特率等参 波特率，校验位，数据位，停止位 | |
| InBufferSize | 1024 | 设置输入缓冲区大小 | |
| OutBufferSize | 1024 | 设置输出缓冲区大小 | |
| RThreshold | 1 | 设置收到多少个字符后触发 OnComm 事件 | |
| InputMode | 1 | 设置输入方式 二进制方式 1 文本模式 0 | |
| InputLen | 0 | 设置当前接收长度为 0 | |
| DataBits | 8 | | |
| StopBits | 1 | | |
| BaudRate | 9600 | | |
| DTREnable | 1 | | |
| Handshaking | 0 | | |
| NullDiscard | 0 | | |
| ParityReplacement | "?" | | |
| RTSEnable | 1 | | |
| SThreshold | 2 | | |
| EOFEnable | 0 | | |
| par0 | | | |

| | | | |
|---------------|------------|---|---------------------------|
| tostring | true/false | 默认 true false 反馈为 buffer | 反馈的数据默认由 buffer 转为 string |
| Send_AsciiHex | Ascii/Hex | Ascii Hex | 全局写入方式 |
| Recv_AsciiHex | Ascii/Hex | Ascii Hex | 全局接收格式 |
| toFormat | Ascii/Hex | Ascii Hex | 对本次发送的 data 转换格式。 |
| backNoJson | true/false | 默认 false, 反馈为 json true, 调用的 dll 或者 ocx 反馈的数据直接发送, 不处理编码或者 json 转换 | runCmd 命令反馈的数据处理 |

2.3.1.4USB 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。 | |
| | | | |

2.3.1.5热拔事件

{"emit": "open", "Obj": "driveNotify", "AppId": "notifyAppId1", "AppType": 2, "AppShow": false}

| 名称 | 设置 | 含义 | 说明 |
|---------|-----------------------|---|----|
| emit | open | 必需。监控热拔请求。 | |
| Obj | driveNotify | | |
| AppId | notifyAppId1 | 键值。最好设置为特殊的 | |
| AppType | 数值或者数组 2 , [1, 2], | 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 | |
| AppShow | false | 无界面 | |

插入 U 盘

Obj: "driveNotify"

data: {deviceName: "\\Device\\HarddiskVolume16\\", drive: "G:", driveType: 2, eventType: "insert", flags: 0}

eventType: "insert"

insertCnt: 1

rep: 0

sid: "1"

拔出 U 盘反馈

Obj: "driveNotify"

```
data: {drive: "G:", driveType: 1, eventType: "remove", flags: 0}
eventType: "remove"
removeCnt: 1
rep: 0
sid: "1"
```

关闭

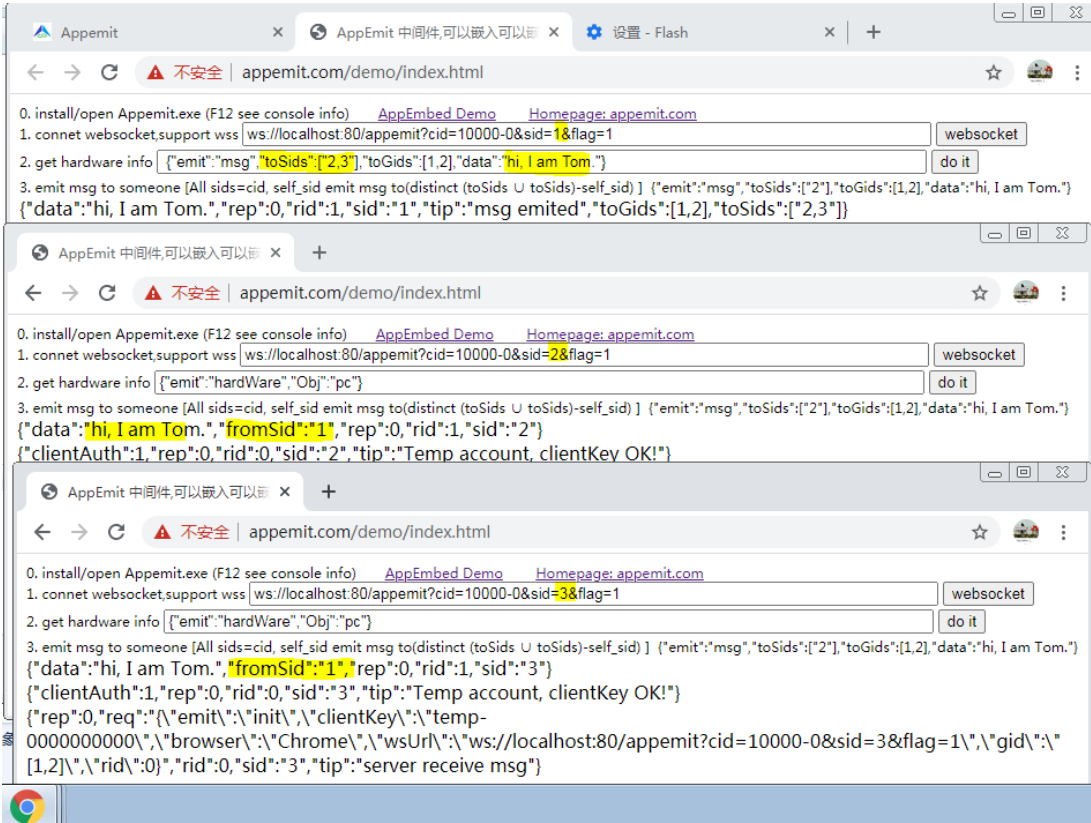
```
{"emit": "close", "Obj": "driveNotify", "AppId": "notifyAppId1"}
```

2.3.2 用户间通话

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", "Obj": "sid", "toSids": ["2"], "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 交集剔除重, 并排除自身。

```
{"emit": "msg", "Obj": "sid", "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。 | |
| Obj | sid | 用户自行设定的 sid | |
| toSids | 必需要有一 | 非必需。可以是数组。 | All sids=cid, self_sid |

| | | | |
|--------|---|------------|---|
| toGids | 个 | 非必需。可以是数组。 | emit msg to(distinct (toSids ∪ toSids)-self_sid) 发送的全集是：发送的用户和发送所在的群（频道、小组）的用户的全集剔除后，排除自身 |
| data | | 必需。 | |

2.3.3 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 {"emit": "setPar", "Obj": "flash", "topMost": false} |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时，App 随浏览器移动变形隐藏显示等事件 |
| par0 | | | 注意不同 APP 可能不同 |
| attach | 1 | 附着 APP 方式 | wps 最好采用此方式。 如果程序出现未知错误，可采用此方式 |
| 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 |
| | | | |

2.3.3.1 打开文件

以系统默认的程序，打开文件。

为了安全，不支持直接打开的文件类型包括
"exe";"msi";"cmd";"js";"jar";"inf";"com";"scr";"reg";"bat";"vbs";"py"
发送命令打开文件，以独立的方式打开，和浏览器进程独立，没有联系。
{ "emit": "open",
"Obj": "file", "AppShow": false, "src": ["/demo/htmlDemo/file/dl.docx", null] }

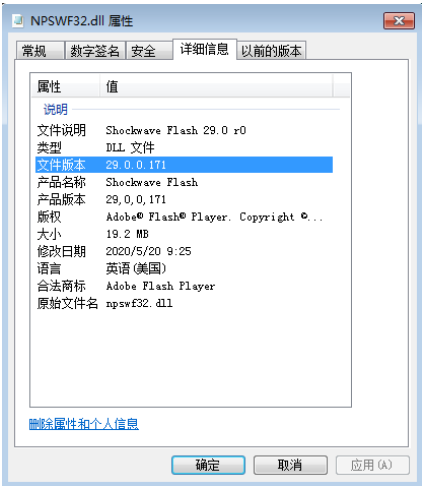
src: EXE 文件路径， 启动参数="", 操作类型 = "open", 显示属性 _SW_, 工作目录=""

| 名称 | 设置 | 含义 | 说明 |
|---------|----------|--|-----------------------|
| emit | open | 必需。打开控件 APP 通信事件请求。 | |
| Obj | file | 必需。 | |
| AppId | 1 | 必需。 | |
| src | | 必需。 | [file, 参数, startInfo] |
| pos | -1 | | |
| AppShow | false | 必需。 | 不发送 pos 位置等数据 |
| par0 | | | |
| UWP | 1、0、null | null 默认。以传统方式打开可执行文件，若失败尝试 UWP 文件打开 0 只以传统方式打开可执行文件 1 只打开 UWP 文件 | |

2.3.3.2 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



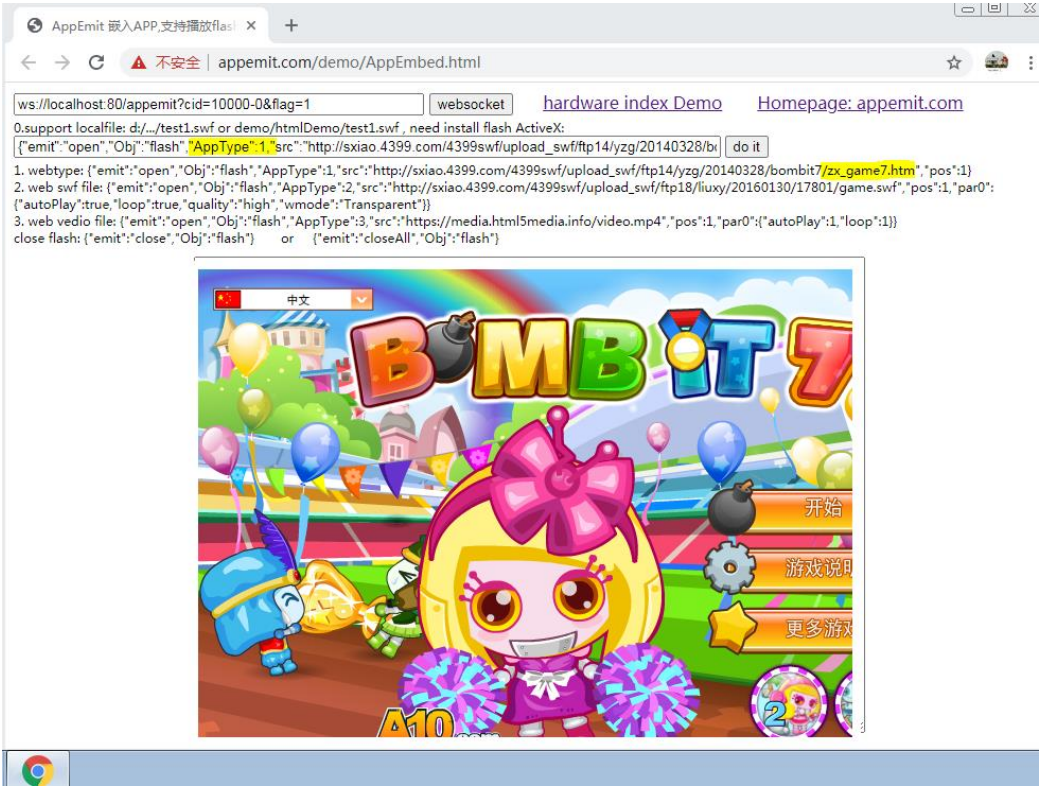
2.3.3.2.1 NPApi 打开 flash 网页 webkit 内核

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_ga
me7.htm”,“pos”:1}



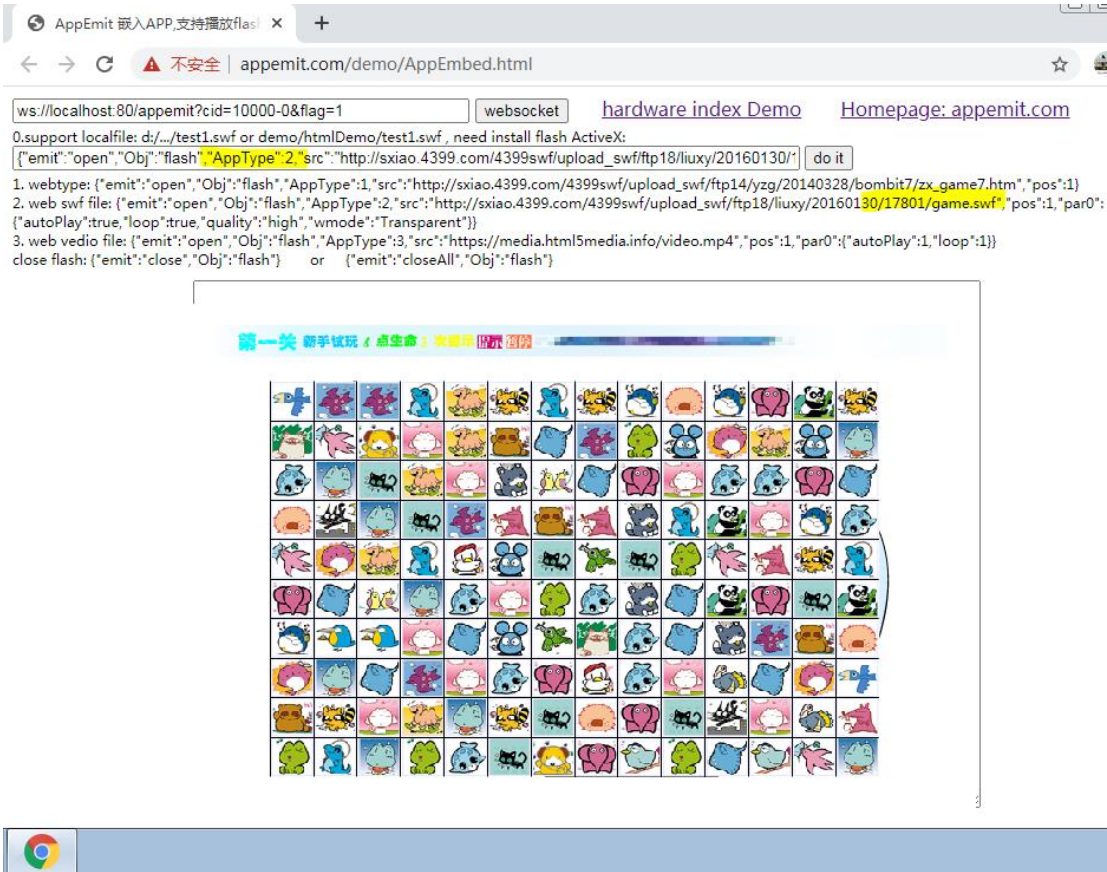
{“emit”:“open”,“Obj”:“flash”,“AppType”:1,“src”:“http://sxiao.4399.com/4399swf/upload_swf/ftp14/yzg/20140328/bombit7/zx_game7.htm”,“pos”:1,“par0”:{“header”:null,“userAgent”:null,“crossDomain”:true}}

| 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 |
| AppShow | | 必需。 | 通常空时默认为 true |

| | | | |
|-----------------|------|---------------------------------------|--------------------------------------|
| | | true App 可见 false App 不可见 | 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时，App 随浏览器移动变形隐藏显示等事件 |
| 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 | |
| | | | |

2.3.3.2.2 NPAPI 打开网络 flash

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”}}



{“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}}

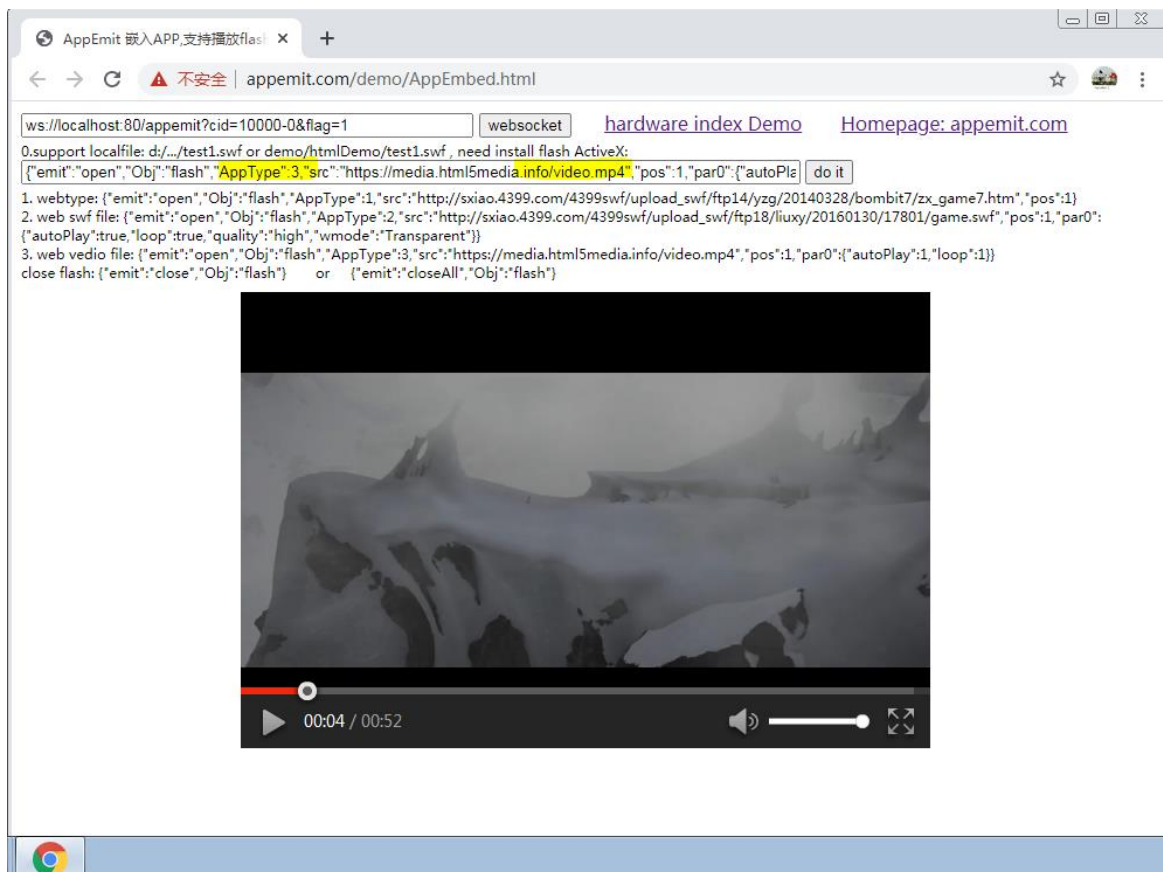
| 名称 | 设置 | 含义 | 说明 |
|------|------|-------------------|----|
| emit | open | 必需。打开控件 APP 通信事件请 | |

| | | | |
|-----------------|---|---|--|
| | | 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":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 |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏显示等事件 |
| 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 | |
| | | | |

2.3.3.2.3 NPAPI 打开 flash 等媒体文件

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}}
```

```
{ "emit": "open", "Obj": "flash", "AppType": 3, "src": "https://cdn.jsdelivr.net/gh/appemit/appemit/docs/vedio/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 3 web 媒体文件 | 若为负数-3, 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": 37, "top": 20, "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 |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏显示等事件 |
| 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 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 | |
| | | | |

2.3.3.2.4 ActiveX 打开 flash 文件

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/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" } }
```

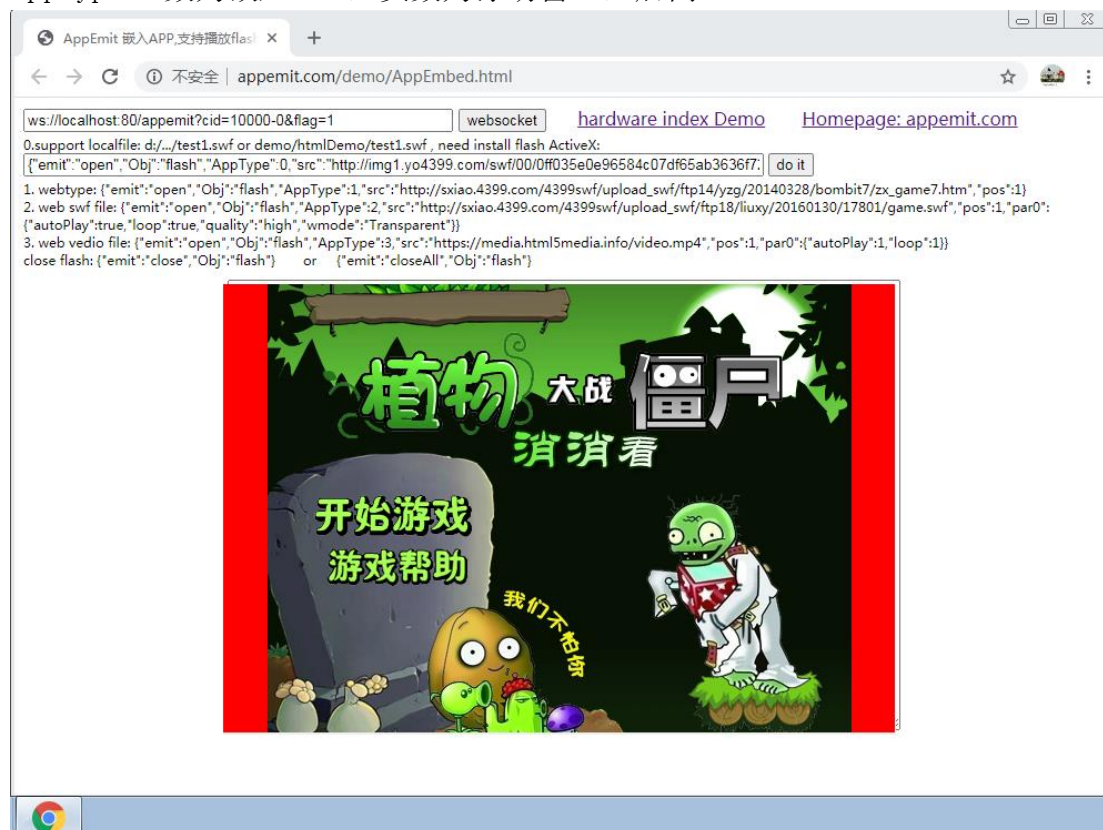
注意事项:

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。

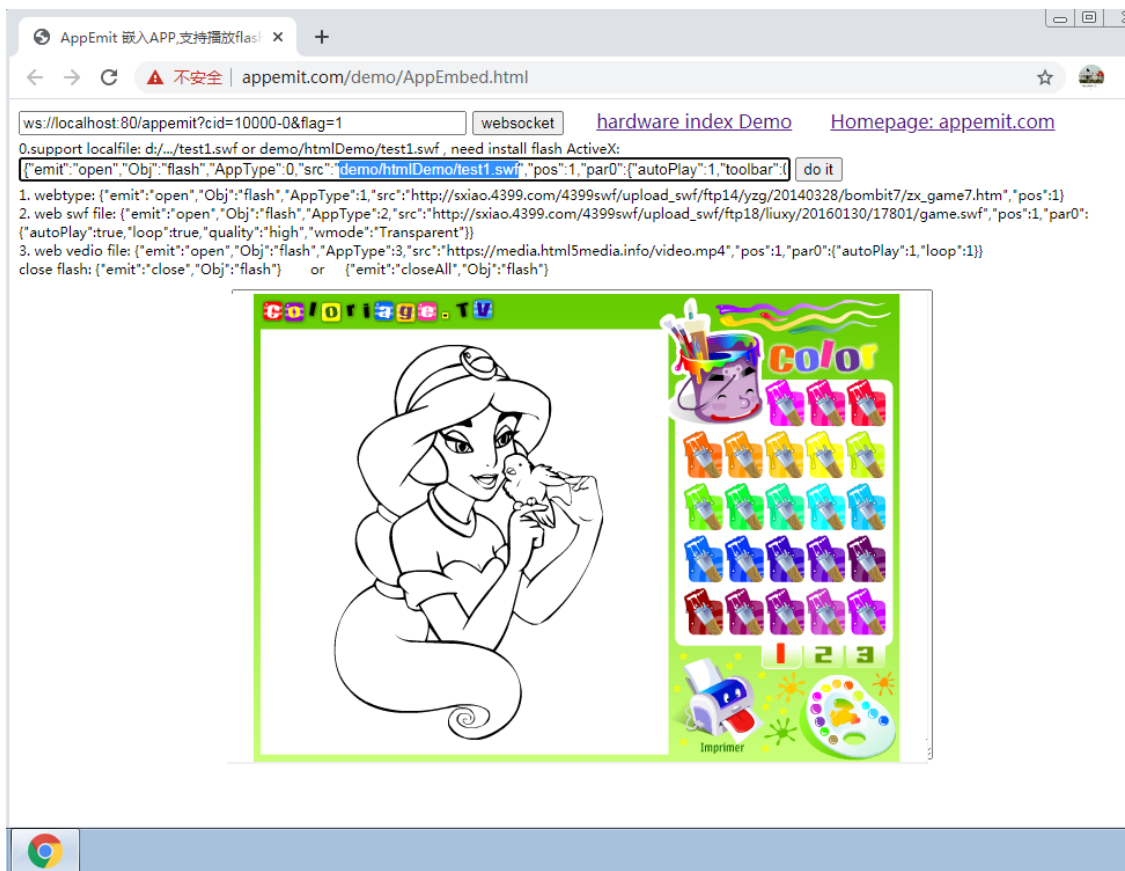


刷新即可关闭 flash

2) 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"}}
```



```
{"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":37 | 必需。 | |

| | | | |
|-----------------|--|--|--|
| | 2, "top":20 3, "width": 606, "height":406} | 1 默认使用代码自动识别的位置。 | |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏 显示等事件 |
| 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 | |
| | | | |

2.3.3.3 打开网页

2.3.3.3.1 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":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 |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏 显示等事件 |
| 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 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。 | |
| emulation | 若不指定, 默认最新的 IE 兼容版本 | 11001 页面始终以 IE11 模式显示, 而不考虑 <!DOCTYPE> 指令 11000 包含基于标准的 <!DOCTYPE> 指令的页面将以 IE11 模式显示 10001 页面始终以 IE10 模式显示, 而不考虑 <!DOCTYPE> 指令 10000 包含基于标准的 <!DOCTYPE> 指令的页面将以 IE10 模式显示 9999 页面始终以 IE9 模式显示, 而 不考虑 <!DOCTYPE> 指令 9000 包含基于标准的 <!DOCTYPE> 指令的页面将以 IE9 模式显示 8888 页面始终以 IE8 模式显示, 而 不考虑 <!DOCTYPE> 指令 8000 包含基于标准的 <!DOCTYPE> 指令的页面将以 IE8 模式显示 7000 包含基于标准的 <!DOCTYPE> 指令的页面将以 IE7 模式显示 | 参考 https://www.csdn.net/article/2012-10-22/2811049 |
| 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*/ _UIFLAG_ENABLE_FORMS_AUTOCOMPLETE=@0x4000/*_UIFLAG_ENABLE_FORMS_AUTOCOMPLETE*/ _UIFLAG_ENABLE_INPLACE_NAVIGATION=@0x10000/*_UIFLAG_ENABLE_INPLACE_NAVIGATION*/ _UIFLAG_IME_ENABLE_RECONVERSION=@0x20000/*_UIFLAG_IME_ENABLE_RECONVERSION*/ _UIFLAG_THEME=@0x40000/*_UIFLAG_THEME*/ _UIFLAG_NOTHEME=@0x80000/*_UIFLAG_NOTHEME*/ | 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 _FLAG_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 允许表单自动完成 _UIFLAG_ENABLE_INPLACE_NAVIGATION 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_NOPICS=@0x100000/*_UIFLAG_NOPICS*/ _UIFLAG_NO3DOUTERBORDER=@0x200000/*_UIFLAG_NO3DOUTERBORDER*/ _UIFLAG_DISABLE_EDIT_NS_FIXUP=@0x400000/*_UIFLAG_DISABLE_EDIT_NS_FIXUP*/ _UIFLAG_LOCAL_MACHINE_ACCESS_CHECK=@0x800000/*_UIFLAG_LOCAL_MACHINE_ACCESS_CHECK*/ _UIFLAG_DISABLE_UNTRUSTEDPROTOCOL=@0x1000000/*_UIFLAG_DISABLE_UNTRUSTEDPROTOCOL*/ _UIFLAG_HOST_NAVIGATES=@0x200000/*_UIFLAG_HOST_NAVIGATES*/ _UIFLAG_ENABLE_REDIRECT_NOTIFICATION=@0x4000000/*_UIFLAG_ENABLE_REDIRECT_NOTIFICATION*/ _UIFLAG_USE_WINDOWLESS_SELECTCONTROL=@0x8000000/*_UIFLAG_USE_WINDOWLESS_SELECTCONTROL*/ _UIFLAG_USE_WINDOWED_SELECTCONTROL=@0x10000000/*_UIFLAG_USE_WINDOWED_SELECTCONTROL*/ _UIFLAG_ENABLE_ACTIVEX_INACTIVATE_MODE=@0x20000000/*_UIFLAG_ENABLE_ACTIVEX_INACTIVATE_MODE*/ _UIFLAG_DPI_AWARE=@0x40000000/*_UIFLAG_DPI_AWARE*/ | _UIFLAG_LOCAL_MACHINE_ACCESS_CHECK 防止远程网页导航到本地计算机 _UIFLAG_DISABLE_UNTRUSTEDPROTOCOL 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_VIDEOS*/ _DLCTL_BGSOUNDS=@0x40/*_DLCTL_BGSOUNDS*/ _DLCTL_NO_SCRIPTS=@0x80/*_DLCTL_NO_SCRIPTS*/ _DLCTL_NO_JAVA=@0x100/*_DLCTL_NO_JAVA*/ _DLCTL_NO_RUNACTIVEXCTLS=@0x200/*_DLCTL_NO_RUNACTIVEXCTLS*/ _DLCTL_NO_DLACTIVEXCTLS=@0x400/*_DLCTL_NO_DLACTIVEXCTLS*/ _DLCTL_DOWNLOADONLY=@0x800/*_DLCTL_DOWNLOADONLY*/ _DLCTL_NO_FRAMEDOWNLOAD=@0x1000/*_DLCTL_NO_FRAMEDOWNLOAD*/ | 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。 _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。 _DLCTL_BGSOUNDS 允许播放文档指定的背景声音 _DLCTL_NO_SCRIPTS web 窗体不执行任何页面脚本 (指 javascript 等) _DLCTL_NO_JAVA web 窗体 |

| | | | |
|-------------|------|---|--|
| | | _DLCTL_RESYNCHRONIZE=@0x2000/* _DLCTL_RESYNCHRONIZE*/ _DLCTL_PRAGMA_NO_CACHE=@0x4000 /*_DLCTL_PRAGMA_NO_CACHE*/ _DLCTL_NO_BEHAVIORS=@0x8000/*_ DLCTL_NO_BEHAVIORS*/ _DLCTL_NO_METACHARSET=@0x10000 /*_DLCTL_NO_METACHARSET*/ _DLCTL_URL_ENCODING_DISABLE_UTF F8=@0x20000/*_DLCTL_URL_ENCODING_DISABLE_UTF8*/ _DLCTL_URL_ENCODING_ENABLE_UTF 8=@0x40000/*_DLCTL_URL_ENCODING_ENABLE_UTF8*/ _DLCTL_NOFRAMES=@0x80000/*_DLCTL_NOFRAMES*/ _DLCTL_FORCEOFFLINE=@0x1000000 0/*_DLCTL_FORCEOFFLINE*/ _DLCTL_NO_CLIENTPULL=@0x200000 00/*_DLCTL_NO_CLIENTPULL*/ _DLCTL_SILENT=@0x40000000/*_DLCTL_SILENT*/ _DLCTL_OFFLINEIFNOTCONNECTED=@ 0x80000000/*_DLCTL_OFFLINEIFNOTCONNECTED*/ _DLCTL_OFFLINE=@0x80000000/*_DLCTL_OFFLINE*/ | 不执行任何 Java applet _DLCTL_NO_RUNACTIVEXCTLS web 窗体不执行文档中的任何 ActiveX 控件; _DLCTL_NO_DLACTIVEXCTLS web 窗体不下载文档中的任何 ActiveX 控件; _DLCTL_DOWNLOADONLY web 窗体下载网页, 但不显示 _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, _DLCTL_RESYNCHRONIZE web 窗体忽略缓存中的数据并向服务器请求更新 _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. _DLCTL_NO_METACHARSET 隐藏文档中的 META 元素指示的字符集; _DLCTL_URL_ENCODING_DISABLE_UTF8 禁止 UTF-8 编码 _DLCTL_URL_ENCODING_ENABLE_UTF8 允许 UTF-8 编码 _DLCTL_NOFRAMES 禁止框架 _DLCTL_FORCEOFFLINE web 窗体工作在脱机方式 _DLCTL_NO_CLIENTPULL web 窗体不执行任何客户端的 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 | 各个含义不同, 请注意。 |

| | | | |
|-----------------|---|---------------------------------------|--|
| | | 0/null 禁用菜单 1 自定义简单菜单 | -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 | |
| | | | |

2.3.3.3.2 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": 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 |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏显示等事件 |
| 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 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。 | |
| header | | 头部 | |
| userAgent | | 代理 | 只写属性, 不可读。 |
| crossDomain | bool | 默认 true | 是否跨域 |

| | | | |
|-----------------|------|--|---|
| | | Ture 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 |
| | | | |

2.3.3.3.3 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": 37, "top": 20, "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 |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏显示等事件 |
| 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 或者 /demo/html | 支持网页地址或者本地 html 文件路径。 | |

| | | | |
|-----------------|----------------|---|---|
| | 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 |

2.3.3.4 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/AppEmitWeb@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 |
| AppShow | | 必需。 true App 可见 | 通常空时默认为 true 调用 comm 为 false |

| | | | |
|-----------------|---|--|--|
| | | false App 不可见 | |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏显示等事件 |
| par | | 必需。 | |
| URL | https://cdn.jsdelivr.net/gh/appeemitWeb/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 | |
| | | | |

2.3.3.5 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。

2.3.3.5.1 msOffice

```
{ "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 对于特殊 wps, 通常采用 |

| | | | |
|-----------|--|---|--|
| | | | attach 方式打开。 |
| 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, 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] | Word 系列文件 Excel 系列文件 PPT 系列文件 不同应用的 openPar 不同, 详细见后 |
| 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 |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏 显示等事件 |
| Par0 | | 可选。 | |
| progId | [] | 默认为 MS office ["Word.Application", "Kwps. Application", "WPS. Application"] ["Excel.Application", "Ket. Application", "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。 默认打开微软 office\金山 wps |
| attach | 1 | 附着 APP 方式 | wps 最好采用此方式。 如果程序出现未知错误, 可采用此方式 |
| noMinMax | 1 | 1 禁用关闭 最小 最大化按钮 0/null 可最小化 关闭 | 遮盖关闭 最小 最大化按钮 |
| 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": "url", "valid": 1, "method": "POST" "username": "name" "username": "*" "cookie": null } | 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。 |
| 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 | Toolbars、btnFile 同时 |

| | | | |
|-----------------|---|---|--------------------------|
| | | 1 显示保存 0 不显示 | 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 | |

2.3.3.5.2 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, E | <i>Type 必需</i> 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 | 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 |

| | | | | | | | | | | | | | | | | | | | | | |
|------------------|---|--|---|-------|----|------|----|---|---------|----|---|----------|----|-----------------------------|--------|----|-------------------------------|------------------|----|------------------------|--|
| | <div>nforceStyle</div> <div>Lock]</div> | <div>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。</div> <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执行格式设置限制。</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执行格式设置限制。 | <div>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</div> |
| 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执行格式设置限制。 | | | | | | | | | | | | | | | | | | | |
| Unprotect | pwd | 文档解锁密码 | <div>对保护的文档 (type!= -1) 输入 解 锁 密 码 Unprotect 不保护。</div> | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | |

2.3.3.5.3 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、 Corrupt Load) | ["/demo/htmlDemo/file/x1.xlsx", null, 1] [[https://cdn.jsdelivr.net/gh/appemit/AppEmitWeb@master/docs/demo/file/x1.tmp, "x11.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、 AllowFormattingC | 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 |

| | | | |
|-------------|------------|--|--------------------------------------|
| | | <code>columns</code> 、 <code>AllowFormattingRows</code> 、 <code>AllowInsertingColumns</code> 、 <code>AllowInsertingRows</code> 、 <code>AllowInsertingHyperlinks</code> 、 <code>AllowDeletingColumns</code> 、 <code>AllowDeletingRows</code> 、 <code>AllowSorting</code> 、 <code>AllowFiltering</code> 、 <code>AllowUsingPivotTables</code>) | e1.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 |

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

2.3.3.5.5 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/AppEmitWeb@master/docs/demo/file/x1.tmp", "a1.xlsx" ], "pos": 1, "par0": { "T
itlebar": 0, "Menubar": 1, "Toolbars": 1, "btnFile": 0, "HttpPost": { "URL": "http://eu.htt
pbin.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/AppEmitWeb@master/docs/demo/file/p1.tmp", "p1.pptx" ], "pos": 1, "par0": { "V
iewType": 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_PR
OPS": 0, "show_UpdateTool": true } }
```

2.3.3.5.6 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

2.3.3.6WPS

创建、打开阅读、编辑金山 Office 文件, 支持本地和网络文件, 包括 wps\et\wpp 文件
更多 VBA 官方参考 <https://open.wps.cn/docs/office> wps 客户端开发
和 office 基本一致, 主要设置 progId 参数和增加了 attach 参数
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": { "attach": 1, "progId":
```

"KWPS.Application", "noMinMax":1, "ViewType":1, "Titlebar":0, "Menubar":1, "Toolbars":1, "btnFile":0}}

| 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。 4 word 5 excel 6 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, 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] | Word 系列文件 Excel 系列文件 PPT 系列文件 不同应用的 openPar 不同, 详细见后 |
| 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 |
| AppShow | | 必需。 true App 可见 false App 不可见 | 通常空时默认为 true 调用 comm 为 false |
| AppFollow | | 必需。空时默认为 true 1 跟随变化 0 不变化 | AppShow 为 true 时, App 随浏览器移动变形隐藏 显示等事件 |
| Par0 | | 可选。 | |
| progId | "KWPS.Application" 或者[] | 默认为 MS office ["KWPS.Application", "WPS.Application"] ["KET.Application", "ET.Application"] ["WPP.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。 |
| attach | 1 | 附着 APP 方式 | wps 最好采用此方式 |
| noMinMax | 1 | 1 禁用关闭 最小 最大化按钮 0/null 可最小化 关闭 | 遮盖 |

其它参数见 Office 章节, 注意金山 wps 的 VBA 代码和微软的不一定相同。

2.3.3.70office javascript

2.3.3.7.1 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] | 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 | 或者设置为 0，把 officeJs 的 |

| | | | |
|--|--|-------------------------------------|--|
| | | 1 IE 2 webkit 3 blink 0 关闭 | 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

2.3.3.7.2 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。

2.3.3.80office 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 网址或者本地相对或者绝对路径 或者数组 [src, flag, ProgId , username, pwd, fileType] | src 为必需。 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/x1.xls", true, "Excel. Sheet", "name", "PWD"} | Word 系列文件 Excel 系列文件 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" "username": "*" " "cookie": null } | 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。 |
| 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 | Toolbars、btnFile 同时 |

| | | | |
|-----------------|---|---|------------------------------|
| | | 1 显示页面设置 0 不显示 | 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 | |

2.3.3.8.1 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/AppEmitWeb@master/docs/demo/file/dl.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/AppEmitWeb@master/docs/demo/file/xl.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/pl.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}}
```

2.3.3.8.2 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 | wdFormatDocument = 0 wdFormatTemplate = 1 wdFormatText = 2 wdFormatTextLineBreaks = | ppSaveAsPresentation = 1 ppSaveAsPowerPoint7 = 2 ppSaveAsPowerPoint4 = 3 ppSaveAsPowerPoint3 = 4 |

| | | |
|---|--|---|
| 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 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 | 3 wdFormatDOSText = 4 wdFormatDOSTextLineBreaks = 5 wdFormatRTF = 6 wdFormatUnicodeText = 7 wdFormatEncodedText = 7 wdFormatHTML = 8 | 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 |
|---|--|---|

2.3.3.8.3 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。

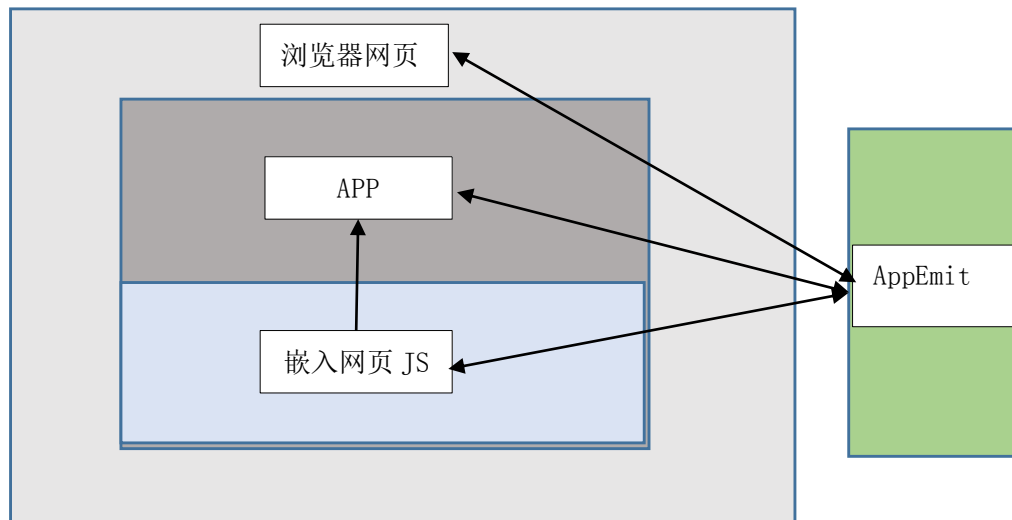
选择保存时如果时间过长，会出现服务器正在运行中，需要在任务管理器关闭对应的 word\excel\ppt 的进程，重新启动。

2.3.3.9 第三方 DLL\组件调用

主要有两种方法来实现。

- 1、通过打开 IE 内核，来 createObject 调用 ActiveX，一般调用 ActiveX 需要已经注册
- 2、JS 控制 AppEmit 调用 DLL 接口，互动信息，有些 ActiveX 可以免注册。

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。



2.3.3.9.1 使用 IE 内核打开 ActiveX

使用 IE 内核打开网页,在非 IE 浏览器中调用 activeX 控件,及使用原有的 html 的代码即可。这种方法通常需要注册对应的 activeX

```

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

```

2.3.3.9.2 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 /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 |
|--------|---|-----------------------------------|--|

2.3.3.9.2.1 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。

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
没有 AppJsObject.shockwave 对象了

2.3.3.9.2 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。

2.3.3.9.3 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":nul  
l,"HttpServer_startUrl":null,"URL":"/demo/htmlDemo/js/dllJs.html","webJs":1},"pa  
r0":{"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 |

2.3.3.9.3.1 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

2.3.3.9.3.2 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。

2.3.3.9.4 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":[],"AppShow":false,"par":{"htmlStr":null,"HttpServer_startUrl":null,"URL":"/demo/htmlDemo/js/dllJs.html","webJs":1},"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 | |
| AppShow | false | 必需。窗口不可见。 无界面必需设置 false。 | |
| par | | | |
| 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 的设置的嵌入网页 JS 1 IE 2 webkit | 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。 |

| | | | |
|------------|------------|--|------------------|
| | | 3 blink 0 关闭 | |
| 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 | |
| backNoJson | true/false | 默认 false，反馈为 json true，调用的 dll 或者 ocx 反馈的数据直接发送，不处理编码或者 json 转换 | 反馈的数据处理 |
| codePage | 65001 | 65001 utf8 1200 UCS-2LE 1201 UCS-2BE 0 不处理 | 反馈的数据编码，默认 65001 |

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 |
|--------|---|-----------------------------------|--|

2.3.3.9.4.1 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
```

2.3.3.9.4.2 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。

2.3.3.9.5 自行 Dll 开发调用

可以通过 2 种方法来实现

1. 开发的 DLL 可以和 AppEmit 保持接口一致, 有些 ActiveX 可以免注册。完成通信调用。
2. 注册开发的 DLL, 在非 IE 浏览器打开 IE 内核来调用, 过打开 IE 内核, 来 createObject 调用 ActiveX , 需要注册 ActiveX 见 web IE 内核章节。

下面介绍第一种方法, 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

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。

2.3.3.9.5.1 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")->valuestring, mycid)==0                &&  
strcmp(cJSON_GetObjectItem(ids_json, "AuthKey")->valuestring, "000")==0            &&  
strcmp(cJSON_GetObjectItem(ids_json, "clsId")->valuestring, 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 websocket 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 websocket 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
websocket 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;
}

```

2.3.3.9.5.2 嵌入网页 JS 控制

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": [], "AppShow": false, "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/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。

下面为 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": [], "AppShow": false, "pos": 0, "par": { "htmlStr": null, "HttpServer_startUrl": null, "URL": "/demo/htmlDemo/js/dllJs.html", "webJs": 1 }, "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 | |
| AppShow | false | 必需。窗口不可见。 无界面必需设置 false。 | |
| par | | | |
| 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 | |
| codePage | 65001 | 65001 utf8 1200 UCS-2LE 1201 UCS-2BE 0 不处理 | 反馈的数据编码，默认 65001 |
| backNoJson | true/false | 默认 false，反馈为 json true，调用的 dll 或者 ocx 反馈的数据直接发送，不处理编码或者 json 转换 | 反馈的数据处理 |

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 |
|--------|---|-----------------------------------|--|

```

➤ 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
```

2.3.3.9.5.3 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。

2.3.4 关闭 APP

刷新即可关闭 APP

```
 {"emit":"close","AppId":1} //空时默认追加 AppId 1
 {"emit":"closeAll"}
```

2.3.4.1关闭 cid 对应 APP

```
 {"emit":"close","AppId ":1}
```

| 名称 | 设置 | 含义 | 说明 |
|-------|---------|---------------------|----|
| emit | close | 必需。关闭控件 APP 通信事件请求。 | |
| AppId | 1, 2... | 必需。具体的可调整。 | |
| Obj | flash | 非必需。 | |

2.3.4.2关闭 cid 所有 APP

```
 {"emit":"closeAll"}
```

| 名称 | 设置 | 含义 | 说明 |
|------|----------|-----------------------|----------------------|
| emit | closeAll | 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 |

2.3.5 APP 窗口控制

2.3.5.1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 |

2.3.5.2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。 | |

2.3.5.3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 |
| | | | |

2.4 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 | |
| | | | |

2.5 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 | |

2.6 appemit 操作

2.6.1 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。 | " |

2.6.2 重启

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

只有一个用户使用的时候才能重启程序。

| 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。 | |

2.6.3 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。 | |

2.6.4 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。 | |
|-----|---------|-----|--|

2.6.5 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 (Public free Version)", "verType": 0, "version": "0.3.5"}, " |
| Obj | appemit | 必需。 | |

2.7 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 |

3 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?

初始时和每 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。

版本分为免费版、个人版、企业版、VIP 版，支持的功能不同。

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。