

多浏览器插件集成方法

成都鱼住未来科技有限公司

公司主页: <http://www.yzfuture.cn>

技术支持: faq@yzfuture.cn

售后: sales@yzfuture.cn

电话: [028-82880293](tel:028-82880293)

日期	版本	说明	作者
2020/07/01	V2.0.0	对多浏览器 的支持，包含 websocket 和 webapi	TygerZH
2023/12/06	V2.0.1	添加对港澳台居住证、外国人居住 证旧版和外国人居住证新版本的支 持	TygerZH

成都鱼住未来科技有限公司

1. WebSocket 接口调用

调用地址: ws://127.0.0.1:30004/ws

- **被动读身份证:** 只要客户端连接成功后, 不需要进行任何操作。当身份证解码服务解码成功后会给所有连接成功的 websock 客户端分发一次解码信息, 客户端解析并显示即可
- **主动读身份证:** 按以下格式发送 json 字符串, 然后接收服务器返回信息解析即可

```
{
  "Cmd": 10000,
  "Head": "YZWL",
  "IPFlag": "MGEyZmU1NmY5ODZlZGMgNg==",
  "UserParam": "解码信息的 base64 编码",
  "Version": "V1.0.0"
} // 上面为 json 头数据, 不需要修改
UserParam 字段的原始格式如下
{
  "DecodePhoto": true, // 是否需要服务端解码身份证图片, 如为 false 则需要客户端自己将 wlt 格式转换成 bmp 或其它图片格式
  "FaceCompare": false, // 扩展字段, 不用修改
  "PhotoFormat": 1, // 图片解码格式 0-bmp 1-jpg (此处最好设置为 1, 因为 bmp 图片过大有可能导致数据出错)
}
```

- **请求 A 卡 SN (不用修改)**

```
{
  "Cmd": 30400,
  "Head": "YZWL",
  "IPFlag": "YWYyNWMxOWQ1ZTY4ZmJhOQ==",
  "UserParam": "",
  "Version": "V1.0.0"
}
```

- **请求身份证 SN (不用修改)**

```
{
  "Cmd": 20400,
  "Head": "YZWL",
  "IPFlag": "YWYyNWMxOWQ1ZTY4ZmJhOQ==",
  "UserParam": "",
  "Version": "V1.0.0"
}
```

- **服务器返回身份证信息格式**

```

{
  "Cmd": 10001, // 命令编号, 不可修改
  "ErrInfo": "", // 错误原因
  "Head": "YZWL", // 不可修改
  "IPFlag": "OWRkYTkYjliMzQ2MjIzMQ==", // 不可修改
  "Ret": 0, // 返回值 0-成功 其它-失败
  "UserParam": "解码后的身份证信息 base64 编码", // 身份证信息
  "Version": "V1.0.0" // 不可修改
}

```

UserParam 字段的原始格式如下

```

{
  "CardType": 32, // 空格:身份证 J:港澳台居住证 I:旧版外国人永久居留身份证 Y:新版
  外国人永久居留身份证

```

```

  "CardInfo": {
    "EnName": "外文姓名 base64 编码", // unicode + base64
    "Country": "国籍", // unicode + base64
    "Version": "版本号", // unicode + base64

    "Name": "姓名 base64 编码", // unicode + base64
    "Sex": "性别 base64 编码", // unicode + base64
    "Nation": "民族 base64 编码", // unicode + base64
    "Birthday": "出生日期 base64 编码", // unicode + base64
    "Address": "地址 base64 编码", // unicode + base64
    "No": "身份证号 base64 编码", // unicode + base64
    "SignedDepartment": "签发机关 base64 编码", // unicode + base64
    "ValidityPeriodBegin": "开始有效期 base64 编码", // unicode + base64
    "ValidityPeriodEnd": "结束有效期 base64 编码", // unicode + base64
    "OtherNO": "通行证类号码 base64 编码", // unicode + base64
    "SignNum": "签发次数 base64 编码", // unicode + base64
    "Remark1": "备用 1base64 编码", // unicode + base64
    "Type": "证件类型标识 base64 编码", // unicode + base64
    "Remark2": "备用 2base64 编码", // unicode + base64
    "Photo": "身份证照片 base64 编码", // base64
    "Finger": "指纹 base64 编码", //base64
    "SN": "SN 的字符串形式", // 16 进制转字符串
    "DN": "DN 的字符串形式" // 16 进制转字符串
  },
  "FaceInfo": {
    "CompareResult": 0, // 0-成功 -1-设备不支持 其它-失败
    "CompareValue": 0.001, // 相似度
    "ErrInfo": "" // 错误信息
  },
  "OtherInfo": "", // 其它信息, 客户端 so 不负责解析这块, 透传出去, 用户自己负责处
  理

```

```

"SerialID": "解码流水号",
"PhotoFormat": 0, // 0-bmp 1-jpg
"BmpInfo": "图片的 base64 编码"
}

```

- 服务器返回 A 卡 SN

```

{
  "Cmd": 30401, // 命令编号, 不可修改
  "ErrInfo": "", // 错误原因
  "Head": "YZWL", // 不可修改
  "IPFlag": "OWRkYTkYjliMzQ2MjIzMQ==", // 不可修改
  "Ret": 0, // 返回值 0-成功 其它-失败
  "UserParam": "base64 编码", // 身份证信息
  "Version": "V1.0.0" // 不可修改
}

```

UserParam 字段的原始格式如下

```

{
  "SN": "SN 的字符串形式", // 16 进制转字符串
}

```

- 服务器返回 B 卡 SN

```

{
  "Cmd": 20401, // 命令编号, 不可修改
  "ErrInfo": "", // 错误原因
  "Head": "YZWL", // 不可修改
  "IPFlag": "OWRkYTkYjliMzQ2MjIzMQ==", // 不可修改
  "Ret": 0, // 返回值 0-成功 其它-失败
  "UserParam": "base64 编码", // 身份证信息
  "Version": "V1.0.0" // 不可修改
}

```

UserParam 字段的原始格式如下

```

{
  "SN": "SN 的字符串形式", // 16 进制转字符串
}

```

2. Webapi 接口调用

无被动模式，每次服务器读卡成功后，需要客户端调用对应的接口才可以获取到解码后的数据。

- 获取身份证信息

接口: <http://127.0.0.1:30004/api/info>

返回: 与 websocket 中 userparam 数据相同

```

{
  "CardType": 32, // 空格:身份证 J:港澳台居住证 I:旧版外国人永久居留身份证 Y:新版

```

外国人永久居留身份证

```

"CardInfo": {
    "EnName": "外文姓名 base64 编码",      // unicode + base64
    "Country": "国籍",                      // unicode + base64
    "Version": "版本号",                    // unicode + base64

    "Name": "姓名 base64 编码",             // unicode + base64
    "Sex": "性别 base64 编码",               // unicode + base64
    "Nation": "民族 base64 编码",            // unicode + base64
    "Birthday": "出生日期 base64 编码",      // unicode + base64
    "Address": "地址 base64 编码",           // unicode + base64
    "No": "身份证号 base64 编码",            // unicode + base64
    "SignedDepartment": "签发机关 base64 编码", // unicode + base64
    "ValidityPeriodBegin": "开始有效期 base64 编码", // unicode + base64
    "ValidityPeriodEnd": "结束有效期 base64 编码", // unicode + base64
    "OtherNO": "通行证类号码 base64 编码",    // unicode + base64
    "SignNum": "签发次数 base64 编码",         // unicode + base64
    "Remark1": "备用 1base64 编码",            // unicode + base64
    "Type": "证件类型标识 base64 编码",        // unicode + base64
    "Remark2": "备用 2base64 编码",            // unicode + base64
    "Photo": "身份证照片 base64 编码",         // base64
    "Finger": "指纹 base64 编码",              //base64
    "SN": "SN 的字符串形式",                   // 16 进制转字符串
    "DN": "DN 的字符串形式"                    // 16 进制转字符串
},
"FaceInfo": {
    "CompareResult": 0, // 0-成功 -1-设备不支持 其它-失败
    "CompareValue": 0.001, // 相似度
    "ErrInfo": "" // 错误信息
},
"OtherInfo": "", // 其它信息, 客户端 so 不负责解析这块, 透传出去, 用户自己负责处
理
"SerialID": "解码流水号",
"PhotoFormat": 0, // 0-bmp 1-jpg
"BmpInfo": "图片的 base64 编码"
}

```

● 获取 A 卡 SN

接口: <http://127.0.0.1:30004/api/asn>

返回: 与 websocket 中 userparam 数据相同

```

{
    "SN": "SN 的字符串形式",      // 16 进制转字符串
}

```

● 获取 B 卡 SN

接口: <http://127.0.0.1:30004/api/bsn>

返回: 与 websocket 中 userparam 数据相同

```
{
  "SN": "SN 的字符串形式",    // 16 进制转字符串
}
```

3. 问题总结

- 是否支持离线版读卡器

支持离线版读卡器。YNR101 和 YZR201 都支持, 身份证放于读卡器上即可。

- 关于浏览器跨域

- ✓ 是否支持跨域

服务支持跨域的, 可以在 IE11 或 360 浏览器上测试。

- ✓ 跨域解决方法

1. 前端使用 https 请求

2. 修改代码或改 dns, 两种资源都改成内网或者外网 IP

3. 配置 chrome 选项为 disable

chrome://flags/#block-insecure-private-network-requests

4. 前端使用 jsonp 方式调用接口, 即在 \$.ajax 中添加这一句: `dataType: "JSONP"`,

- ✓ 浏览器报错: `has been blocked by CORS policy: The request client is not a secure context and the resource is in more-private address space 'local'.`

这是因为最新版的浏览器支持了 CORS-RFC1918 这个标准, 它不允许公共网络资源请求私有网络资源导致的。也就是说最新版本的浏览器有可能会禁止服务器代码使用本地资源和文件, 所以有这种方式调用的用户请自己修改前端代码, 不然有可能以前可以正常使用的功能在浏览器更新后无法使用。具体问题请查看 CORS-RFC1918 标准。可参考我公司在公网上的网页 demo:<https://aidoing.com.cn/webDemo/index.html>

- ✓ 浏览器报错: `No 'Access-Control-Allow-Origin' header is present on the requested resource`

这是因为前端代码存在跨域问题, 需要添加 Access-Control-Allow-Origin 属性, 具体步骤请针对自己代码百度

4.