

# HTML5 手写签名使用说明文档

---

AnySign\_V1.3.3\_Web\_2.3.0

二〇一七年 八 月

**北京数字认证股份有限公司**

北京市海淀区北四环西路 68 号双桥大厦 15 层

( 左岸工社 )

TEL : 86-10-58045600

FAX : 86-10-58045678

邮政编码 : 100080



## 目录

|                                      |    |
|--------------------------------------|----|
| 1. 功能介绍.....                         | 4  |
| 2. 兼容性介绍.....                        | 4  |
| 2.1. 浏览器版本.....                      | 4  |
| 2.2. 手机操作系统版本.....                   | 4  |
| 3. C/S 架构 API 接口集成介绍.....            | 4  |
| 3.1. API 包结构及其介绍.....                | 4  |
| 3.2. 集成步骤（集成流程）.....                 | 5  |
| 3.3. 集成流程图.....                      | 7  |
| 3.4. 注意事项.....                       | 7  |
| 3.5. 手写识别功能.....                     | 8  |
| 3.6. Demo 示例.....                    | 8  |
| 3.7. 回调接口样例.....                     | 9  |
| 3.7.1. 签名回调.....                     | 9  |
| 3.7.2. 识别回调.....                     | 10 |
| 3.8. 接口说明.....                       | 11 |
| 3.8.1. 核心接口.....                     | 11 |
| 3.8.1.1. 初始化 API.....                | 11 |
| 3.8.1.2. 设置模板数据.....                 | 11 |
| 3.8.1.3. 添加签名信息.....                 | 12 |
| 3.8.1.4. 添加批注信息.....                 | 12 |
| 3.8.1.5. 添加公章配置信息.....               | 13 |
| 3.8.1.6. 显示签名框.....                  | 13 |
| 3.8.1.7. 显示批注框.....                  | 14 |
| 3.8.1.8. 添加证据配置信息.....               | 14 |
| 3.8.1.9. 完成配置接口.....                 | 15 |
| 3.8.1.10. 生成签名请求加密包.....             | 16 |
| 3.8.2. 附加接口.....                     | 16 |
| 3.8.2.1. 签名请求包是否准备就绪.....            | 16 |
| 3.8.2.2. 重新配置 API.....               | 16 |
| 3.8.2.3. 保存当前数据.....                 | 17 |
| 3.8.2.4. 恢复之前保存的数据.....              | 17 |
| 3.8.2.5. 初始化识别接口.....                | 18 |
| 3.9. 附录：.....                        | 18 |
| 3.9.1. 对象定义.....                     | 18 |
| 3.9.1.1. SignatureConfig 单签配置信息..... | 18 |
| 3.9.1.2. CommentConfig 批注配置信息.....   | 19 |
| 3.9.2. 常量表.....                      | 20 |
| 3.9.2.1. DataFormat.....             | 21 |
| 3.9.2.2. TemplateType.....           | 22 |
| 3.9.2.3. EncAlgType.....             | 22 |

|  |    |
|--|----|
| 3.9.2.4. BioType.....                                  | 22 |
| 3.9.3. SignRule 签名规则.....                              | 23 |
| 3.9.3.1. 关键字签名.....                                    | 23 |
| 3.9.3.1.1. 关键字定位方式，寻找 PDF 中的关键字，根据关键字位置放置<br>签名图片..... | 23 |
| 3.9.3.1.2. 根据关键字定位签名位置.....                            | 23 |
| 3.9.3.2. 坐标签名.....                                     | 24 |
| 3.9.3.3. 识别参数.....                                     | 24 |

# 1. 功能介绍

H5 签名是北京数字认证股份有限公司其中一款产品的简称。主要功能包括：对指定模板做手写签名、签名动作证据采集（拍照、录音等）、附件添加（动态设置）、加密打包等功能。同北京 CA 信手书后台服务紧密结合。

## 2. 兼容性介绍

### 2.1. 浏览器版本

Internet Explorer 11 以上、Firefox、Opera、Chrome 以及 Safari 支持 <canvas> 及其属性和方法。

### 2.2. 手机操作系统版本

iOS 7.0 以上，Android4.0 以上。

## 3. C/S 架构 API 接口集成介绍

### 3.1. API 包结构及其介绍

CSS 包：定义样式结构包

Libs 包：

1) 第三方类库：

CryptoJS v3.1.2: JavaScript 加密库

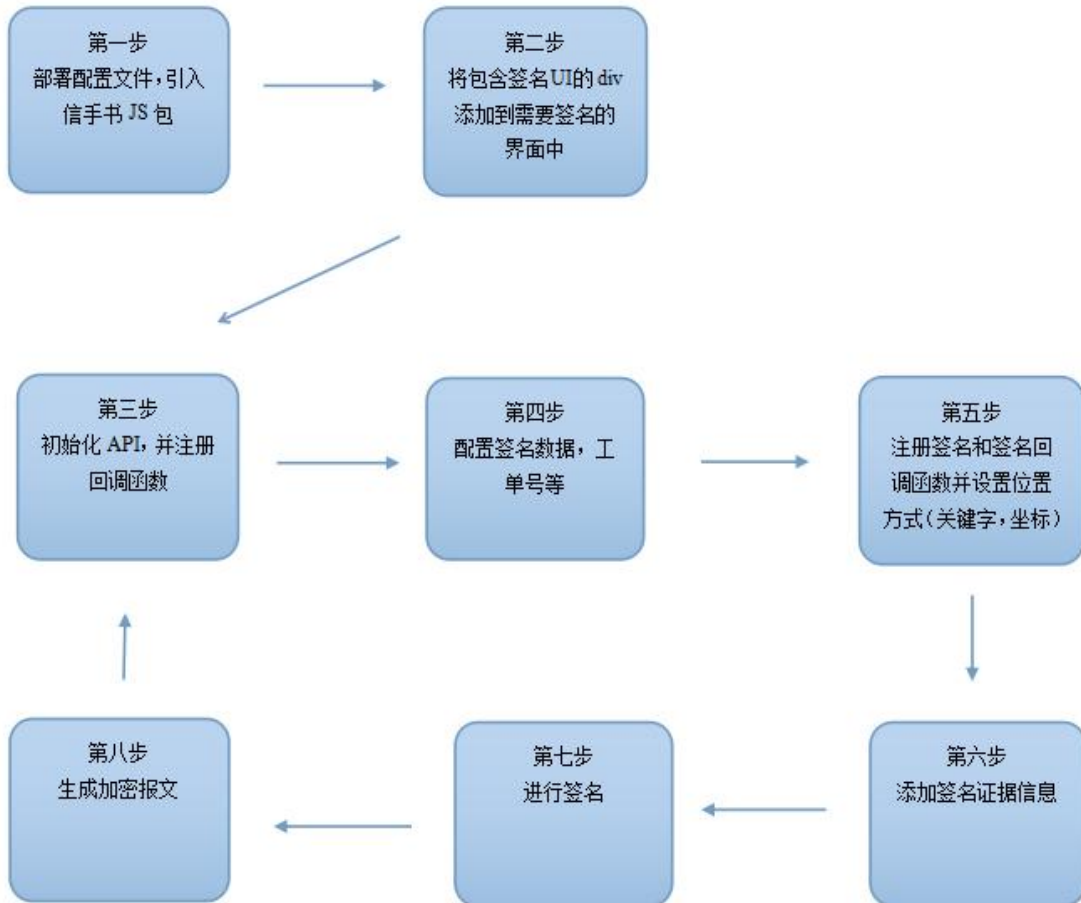
zlib: 提供数据压缩用的函数库

2) 核心加密代码：anysign\_all.js

接口类：anysignWebInterface 浏览器接口类

主要功能是给开发商提供信手书产品相关功能，开发人员直接集成信手书的 js 包，调用信手书的 API 函数即可集成相关功能。相关 API 参考附录 1。

### 3.2. 集成步骤（集成流程）

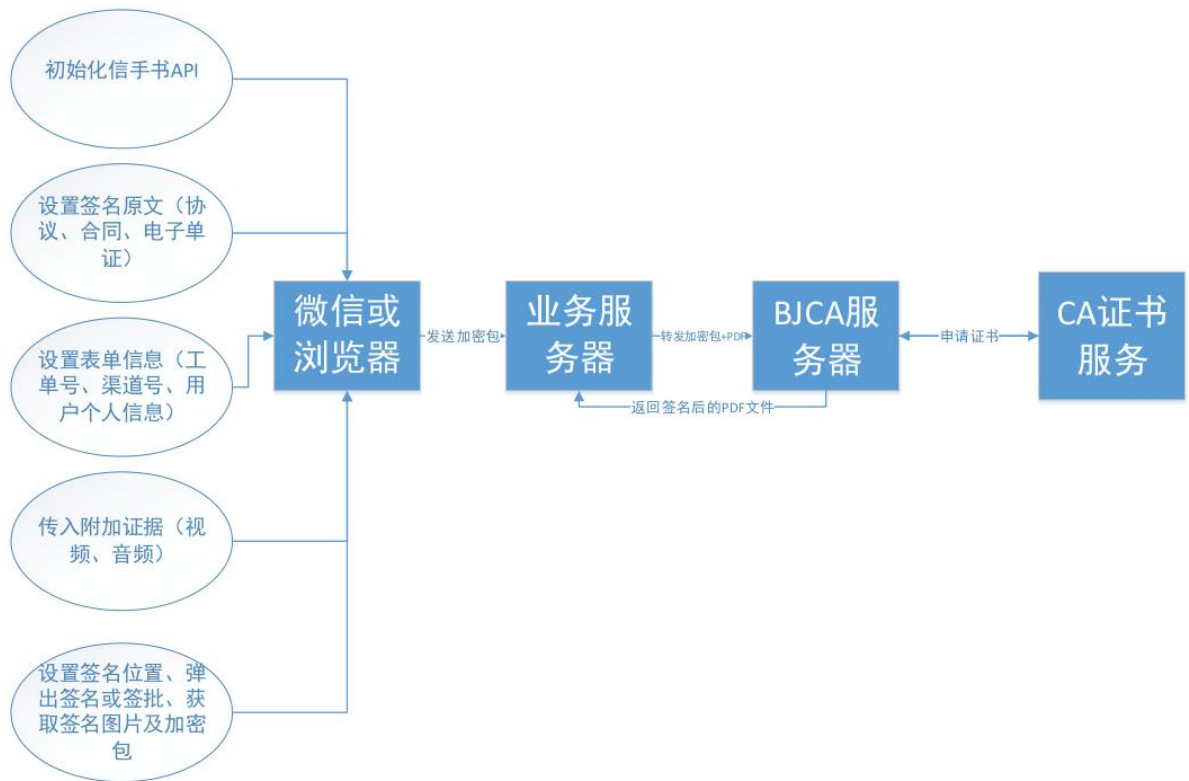


- 1) 调用 `initAnySignApi(callback, channel)` 初始化 AnySign JS api。callback 格式请参考附录；channel 为渠道号。
- 2) 调用 `resetConfig()` 打开配置接口开关，在每次办理业务的时候首先调用一次。当且仅当调用此接口之后，才可以使用 `addSignatureObj` 配置拍照和签名。
- 3) 调用 `addSignatureObj()` 配置若干个签名对象。
- 4) 调用 `addCommentObj()` 配置若干批注对象。

- 5) 3 步设置完，调用 `commitConfig()` 提交配置信息。此接口一旦调用，在下一次调用 `resetConfig()` 之前不可以再调用步骤 3)。
- 6) 调用 `setTemplate()` 设置模板数据。
- 7) 调用 `showSignatureDialog(context_id)` 弹出单签输入框（注：此处的 `context_id` 与 3) 中设置的 `context_id` 要对应上），在签名成功，用户点击确认之后，回调函数会将签名预览图片返回。
- 8) 调用 `showCommentDialog(context_id)` 弹出批注签名框，在签名成功，用户点击确认之后，回调函数会将批注预览图片返回。
- 9) 调用 `getUploadDataGram()` 获取包含签字图片、表单信息等加密信息的报文用于上传服务端。注：必须在用户输入完成配置文件定义的所有手写数据、并且所有附件均已添加内容之后此接口才能传出 `xml`，否则会抛出异常。
- 10) 重复步骤 2)，进行新一次业务办理。

注：目前接口仅支持签名采集，拍照、录音等接口没有实现。

### 3.3. 集成流程图



### 3.4. 注意事项

- 1) 配置的签名的 index 必须与 showSignatureDialog(int context\_id) 或者 showCommentDialog(int context\_id) 中的 context\_id 一致，否则弹框无效。
- 2) 单签签名的 context\_id 范围为 [20, 29] 及 [200, 299]，批注为 [30, 39] 及 [300, 399]，不应该超过该范围。
- 3) 配置输入弹出框的时候务必要设置签名人信息，如 `var signer = new Signer("李明", "11011111111", IDType.IDENTITY_CARD_FRONT)`，否则会弹框无效。
- 4) 配置模板数据时，直接传入数据时需注意要转化为 base64 形式进行上传。
- 5) 在集成时，html 中的 body 中签名区域 id 需保持一致。比如：`<canvas id="anysignCanvas" width="2"></canvas>` 中的 id 值不应该改变。
- 6) 批注的 UI 界面开放，可进行修改批注的 UI。

- 7) <canvas>标签的宽高设置只支持 width, height, 请不要用 style 或者 CSS 中的 width 和 height 进行设置。否则将会出现拉伸现象。
- 8) 在设置单签的 canvas 标签时, 其宽 width 的值为 0~10 时表示其签名框相对于屏幕的倍数, 比如: width=“2”时, 表示其 canvas 的宽是屏幕宽的 2 倍。大于 10 时是正常固定宽度 (width 为整数)。
- 9) 批注第一版本删除按钮只支持从后向前删除, 任意位置删除功能后续版本添加。
- 10) 不建议集成时进行横竖屏来回切换的操作。

### 3.5. 手写识别功能

手写签名服务端为客户端提供手写识别服务, 以 Tomcat 的形式部署在服务器上, H5 端方式采集到的用户的手写轨迹点数据通过 post 请求发送到服务端, 服务端收到轨迹数据之后按照手写识别协议进行封装发送 (post 请求) 到同属于一台服务器上的私有云服务, 私有云会返回识别后的文字对应的数字, 服务端收到识别结果之后, 按照 PC 端或者移动端和服务端的协议返回识别的文字、错误码、协议版本等信息 (json 形式)。

集成步骤:

1. 初始化识别回调函数: `setIdentifyCallBack(identify_callback);`
2. 在初始化签名信息时配置识别信息和参数。

调用时间: 调出签名或者批注框之前, 实现以上接口即为开启识别, 不实现默认不开启。调用手写识别应用前, 在配置信息中必须设置手写识别信息, 即 `obj.ocrCapture` 必须赋值。同时配置签名人人和批注信息中分别有 `obj.isdistinguish` 开关, 设置为必需设置为 true 则会开通。

### 3.6. Demo 示例

示例代码详见集成 demo。



## 3.7. 回调接口样例

### 3.7.1. 签名回调

回调接口必须至少包含 3 个参数，分别用于传递 context\_id、context\_type、data，其中 context\_id 和 context\_type 分别用于告知回调函数，产生此次回调的对象 id 和回调类型，具体定义参考 3.8 附录 常量表。

编写完成回调函数之后，在初始化接口 initAnySignApi(callback,channel)中以参数形式传给信手书接口。如下图所示：

```
var callback = function(context_id,context_type,val)
{
    document.getElementById("other").style.display = "block";
    if(context_type == CALLBACK_TYPE_START_RECORDING || context_type == CALLBACK_TYPE_STOP_RECORDING)
    {
        return;
    }

    if(context_type == CALLBACK_TYPE_SIGNATURE)
    {
        //签名回显
        document.getElementById("xss_20").src = "data:image/png;base64," + val;
        var aImg=document.getElementById("xss_20");
        for(var i=0;i<aImg.length;i++){
            aImg[i].style.height="1500";
            aImg[i].style.width="1500";
        }
    }
    else if(context_type == CALLBACK_TYPE_COMMENTSIGN)
    {
        //签名回显
        document.getElementById("xss_21").src = "data:image/png;base64," + val;
        var aImg=document.getElementById("xss_21");
        for(var i=0;i<aImg.length;i++){
            aImg[i].style.height="250";
            aImg[i].style.width="250";
        }
    }
    else if(context_type == CALLBACK_TYPE_ON_PICTURE_TAKEN)
    {
        document.getElementById("preview").src = "data:image/png;base64," + val;
    }
    else if(context_type == CALLBACK_TYPE_ON_MEDIA_DATA)
    {
        var audio = document.createElement("audio");
        if (audio != null && audio.canPlayType && audio.canPlayType("audio/mpeg"))
        {
            audio.src = "data:image/gif;base64," + val;
            audio.play();
        }
    }
}
```

其中：

**CALLBACK\_TYPE\_SIGNATURE**：单签签名框点击确认之后的回调，回调中包含签名图片。

**CALLBACK\_TYPE\_COMMENTSIGN**：批注签名框点击确认之后的回调，回调中包含签名图片。

**CALLBACK\_TYPE\_DIALOG\_DISMISS**：签名框 dismiss 时的回调。

**CALLBACK\_TYPE\_DIALOG\_CANCEL:** 点击签名框"取消"按钮时的回调，同时也会触发 dismiss 回调。

**CALLBACK\_TYPE\_BUFFER\_SAVED:** 签名数据缓存成功的回调，目前未用到此功能。

### 3.7.2.识别回调

```
//识别回调接口
var identify_callback = function(errCode){
    if(errCode == SUCCESS){
        return;
    }
    if(errCode == DATA_CANNOT_PARSED) {
        alert("输入数据项无法解析!");
    } else if(errCode == SERVICE_SYSTEM_EXCEPTION) {
        alert("服务系统异常错误!");
    } else if(errCode == RECOGNITION_RESULT_EMPTY) {
        alert("识别结果为空!");
    } else if(errCode == CONNECTION_SERVICE_TIMEOUT) {
        alert("连接识别服务超时!");
    } else if(errCode == CONNECTION_RECOGNITION_EXCEPTION) {
        alert("连接识别服务异常!");
    } else if(errCode == RECOGNITION_FALSE) {
        alert("书写错误!");
    } else {
        alert(errCode);
    }
}
```

回调包含错误码，对错误信息进行处理。

**var DATA\_CANNOT\_PARSED = "10003";** //输入数据项无法解析

**var SERVICE\_SYSTEM\_EXCEPTION = "10011";** //服务系统异常错误

**var RECOGNITION\_RESULT\_EMPTY = "10100";** //识别结果为空

**var CONNECTION\_SERVICE\_TIMEOUT = "10101";** //连接识别服务超时

**var CONNECTION\_RECOGNITION\_EXCEPTION = "10102";** //连接识别服务异常

**var SUCCESS = "0";** //识别成功

**var RECOGNITION\_FALSE = "-1";** //识别错误

## 3.8. 接口说明

### 3.8.1. 核心接口

#### 3.8.1.1. 初始化 API

功能:

初始化签名对象, 通常从打开客户端到关闭客户端, 中间只需要初始化一次。

函数定义:

```
bool function initAnySignApi(callback, channel)
```

参数:

**callback:** 回调函数, 用法详见 3.6.

**channel:** 设置业务渠道号, 渠道号应该为小于 20 位的数字, 不能包含字母。

返回值:

**true:** 初始化成功

**false:** 初始化失败

#### 3.8.1.2. 设置模板数据

功能:

设置表单数据, 每次业务都需要 set 一次

函数定义:

```
bool function setTemplate(template_type, contentUtf8Str, businessId,  
    template_serial);
```

参数:

**template\_type:** 模板类型 id:TemplateType.XML: XML 格式, TemplateType.HTML:

HTML 格式, TemplateType.PDF: PDF 格式

**contentUtf8Str:** 表单数据, 类型为 base64 编码的 Utf8 字符串

**businessId:** 业务流水号，用于生成 pdf 的唯一标识，通常为集成信手书的业务系统中的业务流水号

**template\_serial:** 后台签名所用的模板 id，只有 template\_type 为 xml 的时候有效，其他情况可随意设置，需要在签名服务器添加配置信息，可向北京 CA 后台集成人员询问具体值。

返回值：

**true:** 成功

**false:** 失败

### 3.8.1.3. 添加签名信息

功能：

自定义配置一个签名，在 commitConfig() 之前可以调用，context\_id 区间为 [20, 30) 和 [200, 300)，20~29 为普通签名，200~299 为扩展域，意义同 20~29 一致。接口根据 signatureConfig 配置签字相应属性。

函数定义：

```
bool function addSignatrueObj(context_id, signatrueConfig) ;
```

参数：

**context\_id:** 配置一个签名，context\_id 区间为 [20, 30) && [200, 300)。

**signatrueConfig:** 配置签字相应属性

返回值：

**true:** 成功

**false:** 失败

### 3.8.1.4. 添加批注信息

功能：

注册一个批注对象。

函数定义：

```
bool function addCommentObj(context_id, signatrueConfig) ;
```

参数:

**context\_id:** 配置一个签名, context\_id 区间为[30, 40)&&[300, 400)。

**signatrueConfig:** 配置签字相应属性

返回值:

**true:**成功

**false:** 失败

### 3.8.1.5. 添加公章配置信息

功能:

自定义配置一个单位签章, 在 commitConfig() 之前可以调用。

函数定义:

```
bool function addCachetObj(cachetConfig);
```

参数:

**cachetConfig:** 签章配置信息, 参见下文 CachetConfig 对象定义。

返回值:

**true:**成功

**false:** 失败

### 3.8.1.6. 显示签名框

功能:

录入手写签名

函数定义:

```
bool function showSignatrueDialog (context_id)
```

参数:

**context\_id:** 签名的索引值, context\_id 区间为[20, 30)&&[200, 300)

返回值:

**true:** 成功

**false:** 失败

### 3.8.1.7. 显示批注框

功能:

录入手写批注

函数定义:

```
bool function showCommentDialog (context_id)
```

参数:

**context\_id:** 签名的索引值, context\_id 区间为[30, 40)&&[300, 400)

返回值:

**true:** 成功

**false:** 失败

### 3.8.1.8. 添加证据配置信息

功能:

设置第三方自采证据。

函数定义:

```
bool function addEvidence(context_id, content, dataFormat, bioType, index);
```

参数:

**context\_id:** 签名的索引值。

**content:** 证据原文对应的字符串为 base64 编码之后的数据。

**dataFormat:** 格式类型。具体参考附录 3.7.2.1。

**bioType:** 证据类型。具体参考附录 3.7.2.4。

**index:** 证据的索引值, 默认从 0 开始, 最大为 99, 代表第几个证据。

返回值:

**true:** 添加证据成功

**false:** 添加证据失败

### 3.8.1.9. 添加证据 Hash 配置信息接口

功能:

设置第三方自采证据。

函数定义:

```
bool function addEvidenceHash(context_id, content, dataFormat, bioType, index);
```

参数:

**context\_id**: 签名的索引值。

**content**: 证据原文, 用户自己算 hash 值传入。

**dataFormat**: 格式类型。具体参考附录 3.7.2.1。

**bioType**: 证据类型。具体参考附录 3.7.2.4。

**index**: 证据的索引值, 默认从 0 开始, 最大为 99, 代表第几个证据。

返回值:

**true**: 添加证据成功

**false**: 添加证据失败

### 3.8.1.10. 完成配置接口

功能:

提交更改, 一旦调用, 在本次签名流程中不允许再设置表单数据和签名、拍照配置等信息

函数定义:

```
bool function commitConfig ();
```

参数:

无

返回值:

**true**: 成功

**false**: 失败

### 3.8.1.11. 生成签名请求加密包

功能:

产生上传服务端的数据报文，需要手写签名、证据数据均签署完毕。

函数定义:

```
String function int getUploadDataGram()
```

参数:

无

返回值:

加密后的字符串

## 3.8.2.附加接口

### 3.8.2.1. 签名请求包是否准备就绪

功能:

是否签名均已签署，以及附件均已添加等。

函数定义:

```
bool function isReadyToUpload()
```

参数:

无

返回值:

**true:** 成功

**false:** 失败

### 3.8.2.2. 重新配置 API

功能:

重新配置 Api，调用之后可以设置表单数据和签名、拍照配置等信息

函数定义:

```
bool function resetConfig();
```



参数:

无

返回值:

**true:** 成功

**false:** 失败

### 3.8.2.3. 保存当前数据

功能:

保存当前定义的数据, 初始化 API 和原文数据等信息;

函数定义:

```
bool function saveBusiness(encKey);
```

参数:

**encKey:** 对信息数据进行加密所用到的加密 key

返回值:

**true:** 成功

**false:** 失败

### 3.8.2.4. 恢复之前保存的数据

功能:

恢复之前保存的数据

函数定义:

```
bool function restoreBusiness(encData, desKey);
```

参数:

**encData:** 经过加密的初始化 API 和原文数据等信息;

**desKey:** 解密 key;

返回值:

**true:** 成功

**false:** 失败

### 3.8.2.5. 初始化识别接口

函数定义:

```
void function setIdentifyCallBack(callback);
```

参数:

**callback:** 手写识别回调接口。详情参见 3.7.2

## 3.9. 附录:

### 3.9.1.对象定义

#### 3.9.1.1. SignatureConfig 单签配置信息

```
this.signer = signer;//签名人信息，为必填项
```

```
this.signRule = signRule;//签名放置到文档中的规则，如位置，大小等，为必填项
```

```
this.cid = 0;//手写签名对象索引值
```

```
this.title = "请投保人刘伟签名";//签字输入有效，签字框标题
```

```
this.titleSpanFromOffset = 4;//当为普通签名时有效，表示 title 中需要放大显示字体的数组起始 index
```

```
this.titleSpanToOffset = 5;//当为普通签名时有效，表示 title 中需要放大显示字体的数组结束 index
```

```
this.isTSS = false;//是否开启 时间戳，默认为 false
```

```
this.nessesary = false;//是否为必签项
```

```
this.singleWidth = 500;//(只针对签名)生成的签字最大宽度(不排除实际签名宽度小于此值)，单位像素
```

```
this.singleHeight = 500;//(只针对签名)生成的签字最大高度(不排除实际签名高度小于此值)，单位像素
```

`this.penColor = "#000000";`//RGB, 默认为黑色, 每通道为 0~255 的 16 进制值, 如#ffffff 为白色

`this.signatureImgRatio = 1.0;`//保存到加密包中的图片 相对于设置大小的倍数  
如设置为 100\*160, 该值为 2.0 时, 则保存图片为 100\*2.0 \*160\*2.0, 该值越大, 则生成 PDF 中的签名越清晰, 并且所占空间越大

`this.timeTag = null;`//签名人信息, 为必填项, 1:时间在上、2: 时间在下、3: 时间在右

`this.isdistinguish = false;`//是否开启识别

`this.ocrCapture = null;`//识别参数

### 3.9.1.2. CommentConfig 批注配置信息

`this.signer = signer;`//签名人信息, 为必填项

`this.signRule = signRule;`//签名放置到文档中的规则, 如位置, 大小等, 为必填项

`this.cid = 30;`//手写签名对象索引值

`this.commitment = "本人已阅读保险条款、产品说明书和投保提示书, 了解本产品的特点和保单利益的不确定性。";`//批注内容

`this.mass_words_in_single_line = 25;`//生成的签名图片中单行显示的字数

`this.mass_word_height = 75;`//生成的签名图片中单个字的高

`this.mass_word_width = 50;`//生成的签名图片中单个字的宽

`this.penColor = "#000000";`//RGB, 默认为黑色, 每通道为 0~255 的 16 进制值, 如#ffffff 为白色

`this.nessesary = false;`//是否为必签项, 设置为 true 时必须进行签名

`this.isdistinguish = false;`//是否开启识别

`this.ocrCapture = null;`//识别参数

### 3.9.2.常量表

`this.CALLBACK_TYPE_SIGNATURE = 1;` //签名框点击确认之后的回调，回调中包含签名快照

`this.CALLBACK_TYPE_DIALOG_DISMISS = 2;` //签名框 dismiss 时的回调

`this.CALLBACK_TYPE_DIALOG_CANCEL = 3;` //点击签名框“取消”按钮时的回调，同时也会触发 dismiss 回调

`this.CALLBACK_TYPE_BUFFER_SAVED = 4;` //签名数据缓存成功的回调，目前未用到此功能

`this.CALLBACK_TYPE_ON_PICTURE_TAKEN = 5;` //拍照成功回调，回调中包含照片快照

`this.CALLBACK_TYPE_DATA_DELETED = 6;` //成功删除数据回调

`this.CALLBACK_TYPE_START_RECORDING = 7;` //开始多媒体记录回调

`this.CALLBACK_TYPE_STOP_RECORDING = 8;` //结束多媒体记录回调

`this.CALLBACK_TYPE_ON_MEDIA_DATA = 9;` //多媒体数据处理完毕，回传给 js 的回调，回调函数的 data 参数为多媒体数据

`this.CALLBACK_TYPE_COMMENTSIGN = 10;` //批注签名框点击确认之后的回调，回调中包含签名快照

`this.CALLBACK_TYPE_ERROR = -1;` //发生错误时的回调，错误信息如以下 EC\_XXX 定义

`this.CALLBACK_TYPE_ERROR_PICTURE = -2;` //在处理照片时发生异常，具体信息见 value 字段

`this.CALLBACK_TYPE_OPERATION_CANCELED = -3;` //用户取消操作，如取消拍照

`this.CALLBACK_TYPE_GETVERSION = 10;` //获得版本号

//functions return value

`this.RESULT_OK = 0;` //操作成功

`this.RESULT_ERROR = -1;` //操作失败

`this.EC_API_NOT_INITED = 1;` //接口未初始化错误

this.EC\_WRONG\_CONTEXT\_ID = 2;//context\_id 传值错误

this.EC\_CAMERA\_INIT\_FAILED = 3;//摄像头初始化失败，通常重启设备可以解决

this.EC\_NATIVE\_EXCEPTION = 4;// 浏览器 内部 错误 ， 可 至  
Android/sdcard/AnySign/lastCrash.txt 中查询具体错误

this.EC\_DEVICE\_NOT\_SUPPORTED = 5;//当前设备不被浏览器支持，只能调用浏览器功能，信手书模块功能禁用

this.EC\_TEMPLATE\_NOT\_SET = 6;//当没有配置模板时试图调用签名时弹出，因为不能对空数据做签名

### 3.9.2.1. DataFormat

```
var DataFormat =
```

```
{  
    IMAGE_GIF : "image/gif",  
    IMAGE_JPEG : "image/jpeg",  
    IMAGE_PNG : "image/png",  
    MEDIA_AU : "media/au",  
    MEDIA_AIFF : "media/aiff",  
    MEDIA_WAVE : "media/wave",  
    MEDIA_MIDI : "media/midi",  
    MEDIA_MP4 : "media/mp4",  
    MEDIA_M4V : "media/m4v",  
    MEDIA_3G2 : "media/3g2",  
    MEDIA_3GP2 : "media/3gp2",  
    MEDIA_3GP : "media/3gp",  
    MEDIA_3GPP : "media/3gpp"  
}
```

### 3.9.2.2. TemplateType

```
var TemplateType =  
{  
    XML : 10,  
    HTML : 11,  
    PDF : 12,  
    JSON : 13,  
    PRESERVED : 19  
};
```

### 3.9.2.3. EncAlgType

```
var EncAlgType =  
{  
    /** 签名算法 默认为 RSA **/  
    EncAlg : "RSA"  
};
```

### 3.9.2.4. BioType

```
var BioType =  
{  
    /** 签名人居民身份证正面 **/  
    PHOTO_SIGNER_IDENTITY_CARD_FRONT : 0,  
    /** 签名人居民身份证背面 **/  
    PHOTO_SIGNER_IDENTITY_CARD_BACK : 1,  
    /** 签名人复述录音 **/  
    SOUND_SIGNER_RETELL : 2,  
    /** 签名人自定义录音 **/  
};
```

```
SOUND_SIGNER_OTHER : 3  
};
```

### 3.9.3.SignRule 签名规则

#### 3.9.3.1. 关键字签名

##### 3.9.3.1.1. 关键字定位方式，寻找 PDF 中的关键字，根据关键字位置 放置签名图片

```
function SignRule_KeyWord(keyword, keyWordAlignMethod, keyWordOffset,  
    pageNo, KWIndex);
```

参数：

**keyword** 关键字；

**keyWordAlignMethod** 签字图片和关键字位置关系：等于 1 时，签字图片和关键字矩形重心重合；等于 2 时，签字图片位于关键字正下方，中心线对齐；等于 3 时，签字图片位于关键字正右方，中心线对齐；等于 4 时，签字图片左上角和关键字右下角重合，可能额外附加偏移量。

**keyWordOffset** 当 keyWordAlignMethod 非零时，额外附加的偏移量，单位 pt

**pageNo** 寻找关键字的 PDF 起始页码

**KWIndex** 第几个关键字

##### 3.9.3.1.2. 根据关键字定位签名位置

```
function SignRule_KeyWordV2(keyword, xOffset, yOffset, pageNo, KWIndex);
```

参数：

**Keyword:** 关键字

**xOffset:** X 轴偏移量，适配关键字和规则。

**yOffset:** Y 轴偏移量。

**pageNo:** 寻找关键字的 PDF 起始页码

**KWIndex:** 第几个关键字

### 3.9.3.2. 坐标签名

```
function SignRule_XYZ(left, top, right, bottom, pageNo, unit);
```

**参数:**

**Left:** 签名图片最左边坐标值，相对于 PDF 当页最左下角 (0, 0) 点，向上和向右分别为 X 轴、Y 轴正方向

**Top:** 签名图片顶边坐标值，相对于 PDF 当页最左下角 (0, 0) 点，向上和向右分别为 X 轴、Y 轴正方向

**Right:** 签名图片最右边坐标值，相对于 PDF 当页最左下角 (0, 0) 点，向上和向右分别为 X 轴、Y 轴正方向

**Bottom:** 签名图片底边坐标值，相对于 PDF 当页最左下角 (0, 0) 点，向上和向右分别为 X 轴、Y 轴正方向

**pageNo:** 签名在 PDF 中的页码，从 1 开始

**Unit:** 坐标系单位，目前支持“dp”和“pt”

### 3.9.3.3. 识别参数

```
function OCRCapture()
```

**text:** 待校验内容（姓名），字符串类型，必须传值

**IPAddress:** 服务器地址，字符串类型，必须传值

**appID:** 应用 ID, 字符串类型，必须传值

**serviceID:** 渠道号，字符串类型，必须传值

**resolution:** 相似度，数字类型，可选传值，默认 0

**language:** 语言，枚举类型，可选传值，默认为中文简体 CHS，可选中文繁体 CHT，英文 ENG



**count**:返回候选的数目，数字类型, 可选传值，默认为 5

```
var Language =  
{  
    CHS : 1,  
    CHT : 2  
}
```

**识别错误码:**

| 错误类型      | 错误码   | 错误来源 |
|-----------|-------|------|
| 识别成功      | 0     | 无    |
| 识别失败      | -1    | 无    |
| 输入数据项无法解析 | 10003 | 服务端  |
| 服务系统异常错误  | 10011 | 服务端  |
| 识别结果为空    | 10100 | 服务端  |
| 连接识别服务超时  | 10101 | 服务端  |
| 连接识别服务异常  | 10102 | 服务端  |