

签约系统开发者手册

版本1.1
2015年03月23日
文档记录

版本号	修改日期	修订	修改说明
V1.0	2014-11-20	余意	初稿
V1.1	2014-03-23	朴建伟	根据具体实施情况 略有改动

1 概述

2 API接口

- 2.1 准备
- 2.2 请求格式
- 2.3 响应格式
- 2.4 客户端时间戳

3 API详细说明

- 3.1 用户信息
 - 3.1.1 创建新用户
 - 3.1.2 获取用户信息
- 3.2 数字证书
 - 3.2.1 获取用户证书信息
- 3.3 用户印章

3.3.1 为用户创建新的印章并设为默认

3.4 合同

3.4.1 签署合同

3.4.2 查验合同

1 概述

本文档描述的是签约系统向开发者提供的应用程序编程接口（API），如创建用户、创建合同、使用数字证书签署合同等。接口以 HTTP 协议形式提供，开发者可通过接口向签约系统发起请求以完成各项业务。

2 API 接口

2.1 准备

调用 API 前，须向签约支撑平台管理员申请取得 apiId 与 apiSecret，apiId 是 API 标识，apiSecret 是 API 密码。

2.2 请求格式

POST <<URI>> HTTP/1.1
Content-Type: application/json
Content-Signature: HMAC-SHA1 <<authCode>>

```
{  
  "apiId": <<apiId>>,  
  "timestamp": <<timestamp>>,  
  ...  
}
```

请求以 HTTP POST 发送，URI 是具体的 API 标识符。

请求头部：

Content-Type 为 application/json，文本编码为 UTF-8。

Content-Signature 为 API 认证所需信息，HMAC-SHA1 是认证算法，authCode 是认证码。每次请求前，客户端须利用 API 密码计算出认证码，计算方法为
$$\text{authCode} = \text{BASE64}(\text{HMAC-SHA1}(\text{contentString}, \text{apiSecret}))$$
，其中 contentString 是 HTTP 请求的正文部分，即 json 格式的字符串，apiSecret 是 API 密码。

例子：{Accept=application/json, Content-Type=application/json, Content-Signature=HMAC-SHA1 vYSMexdInYGqW7YyJJdUtzzN5Eo=}

请求正文：

正文部分为 json 格式。

apiId 是 API 标识。

timestamp 是客户端时间戳，以 Unix Time 表示，即 UTC 时间 1970 年 1 月 1 日 00:00:00 至当前经过的秒数。

以上两个参数每次请求都必须包含。
其它参数与具体的 API 有关，见 API 详细说明。

2.3 响应格式

```
HTTP/1.0 200 OK
Content-Type: application/json
{
  "isOK": <<returnFlag>>,
  "code": <<returnCode>>,
  "message": <<returnMessage>>,
  ...
}
```

HTTP返回码为200表示请求成功，其他返回码表示服务端错误，无法正常响应。
服务端响应也是json格式，文本编码也是UTF-8。
isOK表示请求是否被成功处理，成功为true，失败则为false。
code是返回码，成功为0，失败时为-1。
message是返回消息，特别是请求失败时用于描述原因。
以上三个字段每次响应都会包含，特别是请求失败时，服务端响应只有此三个字段。
其他字段与具体的API有关，见API详细说明。

2.4 客户端时间戳

服务端会检查请求中的timestamp参数，当客户端时间与服务端时间相差超过10分钟时，该请求不会被处理，服务端将会返回错误码及错误描述。
由于上述原因，客户端与服务端都应确保时间准确，避免误差过大。

3 API详细说明

每次请求都有的2个固定参数：apiId和timestamp，以下记为<<固定请求参数>>。

每次响应都有的三个参数：isOK、code和message，以下记为<<固定响应参数>>。

3.1 用户信息

个人用户信息，以下记为<<userPerson>>，格式为：

```
{
  "userId": <<用户标识，以UUID表示，创建新用户时无字段>>,
  "userType": 0,
  "name": <<姓名>>,
  "idCardNum": <<居民身份证号码>>,
  "mobile": <<手机号>>
}
```

公司用户信息，以下记为<<userOrg>>，格式为：

```
{
  "userId": <<用户标识，以UUID表示，创建新用户时无此字段>>,
  "userType": 1,
  "name": <<机构名称>>,
  "orgCode": <<可选字段，组织机构代码>>,
  "businessNum": <<可选字段，工商营业执照号>>,
  "legalPersonName": <<法人>>,
}
```

```
"transactorName": <<经办人姓名>>,
"transactorIdCardNum": <<经办人居民身份证号码>>,
"transactorMobile": <<经办人手机号>>
}
```

3.1.1 创建新用户

API地址：
/esp/api/createUser

请求：

```
{
<<固定请求参数>>,
"user": <<userPerson>>或<<userOrg>>
}
```

实例：

```
{
"timestamp":1428978181682,
"apiId":"itrus",
"user":{"autoCert":"true",
        "fullname":"姓名",
        "userType":0,
        "idCardNum":"111222333344445555",
        "mobile":"18899993333"}
}
```

成功响应：

```
{
<<固定响应参数>>,
"userId": <<用户标识>>
}
```

实例：

```
{
"isOK":true,
"userId":"65b10ab2b388454ea38e47fc37998767",
"code":0
}
```

注：创建新用户时会默认为用户申请证书及印章。

3.1.2 获取用户信息

API地址：
/esp/api/getUserByIdCardNum

请求：

```
{
<<固定请求参数>>,
"type": 0或1,(0:个人用户;1:企业用户)
"idCardNum"或"orgCode": <<用户证件号>>
}
```

实例：

```
{
```

```
"timestamp":1428978922559,
"apiId":"itrus",
"type":0,
"idCardNum":"111222333344445555"
}
```

成功响应：

```
{
<<固定响应参数>>,
"userId":<<用户标识>>,
"name":<<用户姓名>>,
}
```

实例：

```
{
"isOK":true,
"code":0,
"userId":"566ae4e9aaf04e0690e03fe04d8f28c1",
"name":"张三"
}
```

3.2 数字证书

数字证书信息，以下记为<<cert>>，格式为：

```
{
"certId": <<证书标识>>,
"status": <<证书状态，数字类型，0:有效，1:吊销，2:过期，3:冻结>>,
"statusDesc": <<证书状态描述>>,
"certSn": <<证书序列号>>,
"issuer": <<证书颁发者>>,
"subject": <<证书主题>>,
"notBefore": <<有效期起始时间>>,
"notAfter": <<有效期截止时间>>,
}
```

3.2.1 获取用户的数字证书信息

API地址：

/esp/api/getCertsByUser

请求：

```
{
<<固定请求参数>>,
"type": 0或1,(0:个人用户;1:企业用户)
"pageSize":一页包含个数
"pageNum":0是第一页
"userId": <<用户证件号>>
}
```

实例：

```
{
"timestamp":1428979432825,
"apiId":"itrus",
```

```
"type":0,
"userId":"111222333344445555",
"pageSize":10,
"pageNum":"0"
}
```

成功响应：

```
{
<<固定响应参数>>
certs: <<证书信息>>
}
```

实例：

```
{
"certs":{"total":12,
"list":[{"certId":"9480468fc3e5407b8372e9f5ee0254f1",
"certSN":"3A535B0F43B0DE7F5F72EC66957318CD003B2CA0",
"issuer":"C=CN, O=天威诚信, OU=RSA, CN=天威诚信RSA测试用户CA",
"notAfter":"2016-04-07 11:15:08.0",
"notBefore":"2015-04-08 11:15:08.0",
"status":"0",
"statusDesc":"",
"subject":"CN=张三, OU=RA, O=天威诚信RSA"}]},
"isOK":true,
"code":0
}
```

3.3 用户印章

印章信息，以下记为<<seal>>，格式为：

```
{
"sealId": <<印章标识，以UUID表示，创建新印章时无此字段>>,
"sealImage": <<以BASE64编码的印章图片>>,
"sealDesc": <<印章描述>>
}
```

3.3.1 为用户创建一个新的印章

API地址：

/esp/api/createSealByUserId

请求：

```
{
<<固定请求参数>>,
"type": 0或1,(0:个人用户;1:企业用户)
"userId": <<用户证件号>>,
"seal": <<seal>>
}
```

实例：

```
{
"timestamp":1428980520098,
```

```

"apiId":"itrus",
"seal":{"sealImage":"....",
        "sealDesc":"why!"},
"userId":"111222333344445555",
"type":0
}

```

成功响应：

```

{
<<固定响应参数>>,
"sealId": <<印章标识>>
}

```

实例：

```

{
"isOK":true,
"code":0,
"sealId":"b41f77b0732f42c3865d1a0aec681d16"
}

```

3.4 合同

签约系统支持pdf类型的文档创建成为合同。

一份合同上可以有多个签章。每个签章在页面上的位置以x-y坐标表示，positionX表示与页面左边的距离，positionY表示与页面底边的距离，该距离以72ppi为比例，例如A4尺寸的pdf文件，页面宽度为595像素，高度为842像素。

创建新合同所需的签章位置信息，以下记为<<signatureRaw>>，格式为：

```

{
"userType": 0或1,(0:个人用户;1:企业用户),
"idCardNum"或"orgCode": <<用户证件号>>,
"page": <<签章在合同文档中的页码>>,
"positionX": <<签章在页面内的x坐标>>,
"positionY": <<签章在页面内的y坐标>>
}

```

带标识的签章信息，见于服务端响应，以下记为<<signature>>，格式为：

```

{
"signatureId": <<签章标识，以UUID表示>>,
"contractId": <<合同标识，以UUID表示>>,
"signatureNum": <<合同创建者制定的签章编号，字符串型（若创建时未传无此字段）>>,
"userId": <<合同参与者的用户标识，指明签章由此用户签署，以UUID表示>>,
"page": <<签章在合同文档中的页码>>,
"positionX": <<签章在页面内的x坐标>>,
"positionY": <<签章在页面内的y坐标>>,
"isSigned": <<签章是否已被签署，布尔型，false表示未签，true表示已签>>,
"signingTime": <<签署时间（已签署则有此字段）>>,

```

```

"sealId": <<签署所使用的印章标识（已签署则有此字段）>>,
"certId": <<签署所使用的证书标识（已签署则有此字段）>>
}

```

创建新合同时的合同信息，以下记为<<contractRaw>>，格式为：

```

{
"title": <<合同标题>>,
"docNum": <<合同创建者制定的文档编号，字符串型>>,
"docType": <<合同文档格式，支持pdf、doc和docx>>,
"doc": <<以BASE64编码的合同文档>>,
"signatures": [
<<signatureRaw>>,
...
]
}

```

带标识的合同信息，见于服务端响应，以下记为<<contract>>，格式为：

```

{
"contractId": <<合同标识，以UUID表示>>,
"title": <<合同标题>>,
"docNum": <<合同创建者制定的文档编号>>,
"docType": <<合同文档格式，支持pdf、doc和docx>>,
"doc": <<以BASE64编码的合同文档，默认无此字段，仅客户端要求时包含>>,
"signatures": [
<<signature>>,
...
],
"status": <<合同状态，数字类型，0:未签署，1:签署中，2:签署完成，3:签署中止>>,
"dateCreated": <<创建时间>>,
"lastUpdated": <<最后更新时间>>
}

```

3.4.1 签署

API地址：

/esp/api/sign

请求：

```

{
<<固定请求参数>>,
"contracts": <<contractRaw>>, <<contractRaw>>,.....
"docRequired": true
}

```

实例：

```

{
"timestamp":1428980990791,
"apiId":"itrus",
"docRequired":true,
"contracts":[{"docNum":"20150410",
"title":"合同",
"docType":"pdf",
"fileName":"jkxy_FS0820150128001-02.pdf",

```



```

        "doc": "...",
        "signatures": [
            {
                "page": 1,
                "positionX": 100,
                "positionY": 100,
                "userType": 0,
                "idCardNum": "111222333344445555"
            },
            {
                "page": 1,
                "positionX": 200,
                "positionY": 200,
                "userType": 1,
                "orgCode": "123-456"
            }
        ]
    }

```

成功响应：

```

{
    <<固定响应参数>>,
    "contract": <<contract>>
}

```

实例：

```

{
    "message": "success",
    "isOK": true,
    "contracts": [
        {
            "contractId": "92ea3f60f2b5462f80610e44f6098b99",
            "title": "合同",
            "docNum": "20150410",
            "docType": "pdf",
            "doc": "...",
            "signatures": [
                {
                    "signatureId": "78e1978831c54538bb46e5e0fee7885e",
                    "contractId": "78e1978831c54538bb46e5e0fee7885e",
                    "userId": "566ae4e9aaf04e0690e03fe04d8f28c1",
                    "page": 1,
                    "positionX": 100,
                    "positionY": 100,
                    "isSigned": true,
                    "signingTime": "2015-04-14T11:14:42+0800",
                    "certId": "9480468fc3e5407b8372e9f5ee0254f1",
                    "sealId": "b41f77b0732f42c3865d1a0aec681d16"
                },
                {
                    "signatureId": "d34e12918f6a4dcf98c7ba5ac4129a67",
                    "contractId": "d34e12918f6a4dcf98c7ba5ac4129a67",
                    "userId": "7dd3315ba88c4512899e7b75727fec72",
                    "page": 1,
                    "positionX": 200,
                    "positionY": 200,
                    "isSigned": true,
                    "signingTime": "2015-04-14T11:14:42+0800",
                    "certId": "1a083a0263024e8dbeea32667899d160",
                    "sealId": "8608351ff21d407e93bc953c8bf6e7a8"
                }
            ],
            "status": 2,
            "dateCreated": "2015-04-14T11:14:42+0800",
            "lastUpdated": "2015-04-14T11:14:42+0800"
        }
    ],
    "code": 0,
}

```

```
"isResult":false  
}
```

3.4.2 查验合同

API地址:

/esp/contracts/verify

请求：

```
{  
<<固定请求参数>>,  
"contractId": <<合同标识>>  
}
```

成功响应：

```
{  
<<固定响应参数>>,  
"result": <<查验结果，true表示通过，false表示未通过>>,  
"report": <<进一步的描述>>  
}
```