

# 资源律动开放签

API 接口/SDK

私有化版本

V1.0

北京资源律动科技有限公司

2024 年 1 月

## 目录

一、 总体说明 .....	3
1.1 通信方式 .....	3
1.2 固定请求参数 .....	3
1.3 固定响应参数 .....	3
1.4 申请 token .....	3
二、 企业印章制作 .....	4
2.1 接口说明 .....	4
2.2 请求报文 .....	4
2.3 响应报文 .....	4
三、 签名证书（事件证书）签发 .....	5
3.1 接口说明 .....	5
3.2 调用流程 .....	6
3.3 请求报文 .....	6
3.4 响应报文 .....	7
四、 文件签署 .....	8
4.1 接口说明 .....	8
4.2 调用流程 .....	9
4.3 请求报文 .....	9
4.4 响应报文 .....	13
五、 SDK 集成 .....	14
5.1 SDK 说明 .....	14
5.2 接入步骤 .....	14
5.3 SDK 依赖 .....	15
5.4 接口调用 .....	16
5.5 获取结果 .....	17
附录 .....	18
返回码表 .....	18

# 一、 总体说明

## 1.1 通信方式

Http(s)、Json 报文格式、字符编码：UTF-8

访问地址：http(s)://本地服务 IP:端口

## 1.2 固定请求参数

Http 头：

字段 ID	值	备注
Content-Type	application/json	

公共请求参数：

字段 ID	字段名称	类型	必传	备注
uniqueCode	流水号	string (36)	是	36 位不重复的流水号，随机生成即可，无状态，仅为了标识某笔请求，方便排查问题

## 1.3 固定响应参数

字段 ID	字段名称	类型	必传	备注
code	返回码	integer(5)	是	10000 成功，其他参见附录
message	返回描述	string(100)	是	返回描述，不可用于流程控制
uniqueCode	流水号	string(36)	是	流水号，请求时传递的参数

## 1.4 申请 token

1. 在使用正式的 CA 证书之前请先联系北京资源律动科技有限公司获取授权 token，获取 token 后，将 token 配置在开放签接口服务项目中。
2. 用户必须严格保管授权 token，避免泄露，否则将危及财产安全。如已泄漏，请立刻联系北京资源律动科技有限公司更换授权 token。

## 二、企业印章制作

### 2.1 接口说明

地址：/kaifangqian/openAPI/signature/make

请求类型：POST

备注：

1. 支持企业圆章的制作；
2. 印章样式如下：



### 2.2 请求报文

data：印章制作信息 JSON 格式字符串，包含以下制作信息项

字段 ID	字段名称	类型	必传	备注
topText	环绕文字	string(20)	是	6~20 位，建议传递真实章的环绕文字
middleText	横排文字	string(10)	否	2~10 位，如：合同专用章

请求报文示例：

```
{
  "uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",
  "topText": "开放签演示企业专用",
  "middleText": "合同专用章"
}
```

### 2.3 响应报文

字段 ID	字段名称	类型	必返	备注
-------	------	----	----	----

entSeal	印章图片	string	否	制作完成的印章图片的 base64
---------	------	--------	---	-------------------

响应报文示例：

```
{
  "code": 10000,
  "message": "成功",
  "uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",
  "data": {
    "entSeal": "MIIKoAYJKoZIhvcNAQcCoIIKkTCCIFPT..."
  }
}
```

或

```
{
  "code": 21000,
  "message": "topText 参数缺失",
  "uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",
  "data": null
}
```

## 三、 签名证书（事件证书）签发

### 3.1 接口说明

地址：/kaifangqian/openAPI/cert/event

请求类型：POST

备注：

1、当未获取授权 token 时，该接口签发的是测试证书，非 CA 签发，仅用于测试，不具备法律效力。

2、在上线生产环境时，请切换正式证书接口！

3、正式的事件证书：由权威的 CA 机构进行签发，有效期 1 天，超出有效期的证书无法进行签署；

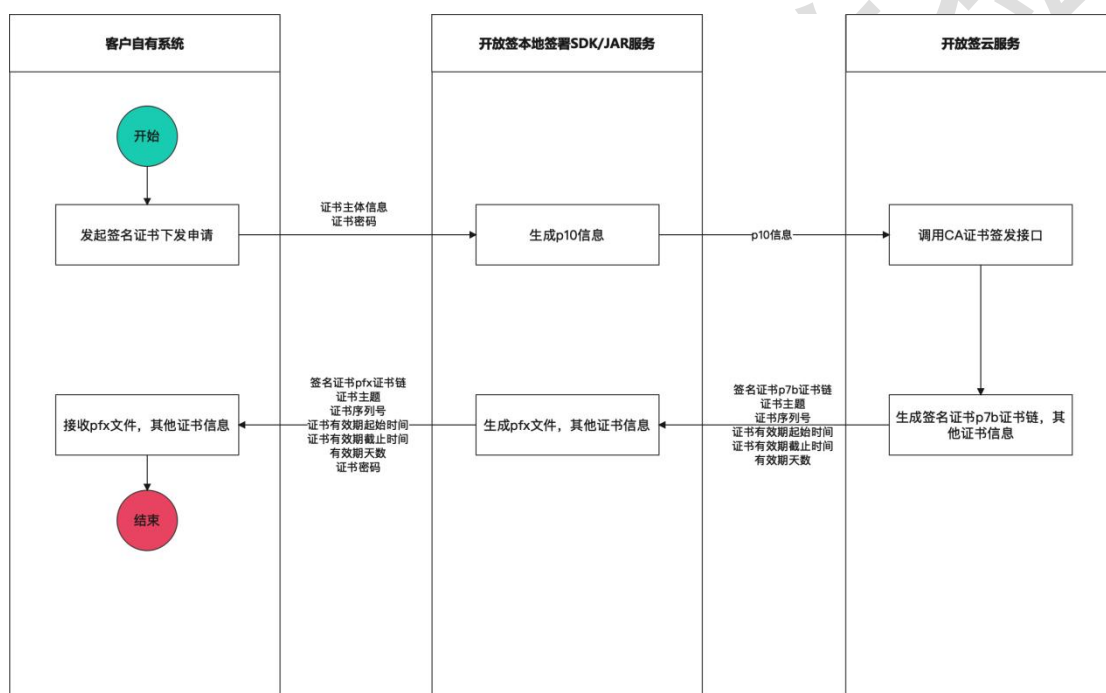
建议：事件证书是针对于某一事项签发的证书，时效性短，在使用对应的证书完成签署后，建议重新签发新的事件证书完成其他文件的签署。

4、所有证书主体（个人/企业）须先完成实名认证，只有完成实名认证的个人/企业才能为其签发证书，否则存在证书签发的法律风险！证书主体的身份真实性由服务调用方负责验证。

5、在文件签署环节，客户自有业务系统应明确告知签署人数字证书相关的使用协议，同时，在文件签署时，须获得签署人授权，完成签署人的意愿校验；

6、系统应保留文件签署的完整证据链，包括签署主体的实名认证记录、证书签发记录、签署记录、签署主体的意愿校验记录等。

## 3.2 调用流程



## 3.3 请求报文

data: 证书申请信息 JSON 格式字符串，包含以下申请信息项

字段 ID	字段名称	类型	必传	备注
certSubject	证书主题	string(100)	是	如果证书主体是个人，则证书主题为个人信息； 如果证书主体是企业，则证书主题为企业信息； 建议：各信息之间使用“@”拼接

				<p>示例：北京资源律动有限公司</p> <p>@91110108MA034NU401</p> <p>注意：证书主体中涉及个人敏感信息的可脱敏传输。</p> <p>示例：胡鑫@3412***855@133***11</p>
certPassword	证书密码	string(32)	是	<p>6~32 位，该密码不会传输到开放签云服务，主要用于在本地将 p7b 证书转换为 pfx</p>

请求报文示例：

```
{
  "uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",
  "certSubject": "开放签科技有限公司@123456789987654321",
  "certPassword": "QExADALBgkqhkiG9w0BB"
}
```

### 3.4 响应报文

字段 ID	字段名称	类型	必返	备注
certSubject	证书主题	string(100)	否	证书主题
certSN	证书序列号	string(50)	否	证书序列号
pfx	签名证书文件	string	否	签名证书文件 base64
certPassword	证书密码	string(32)	否	请求报文中上传的证书密码
certValidityNotBefore	证书有效期起始时间	string(14)	否	证书有效期起始时间（格式：yyyyMMddHHmmss）
certValidityNotAfter	证书有效期截止时间	string(14)	否	证书有效期截止时间（格式：yyyyMMddHHmmss）
certValidity	有效期天数	integer(3)	否	有效期天数

响应报文示例：

```
{
  "code": 10000,
  "message": "成功",
}
```

```
"uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",
"data": {
  "pfx": "MIIKoAYJKoZIhvcNAQcCoIIKkTCCCo0CAQExADALBgkq...",
  "certPassword": "QExADALBgkqhkiG9w0BB",
  "certSN": "659CDF93A6BE1787",
  "certSubject": "开放签科技有限公司@123456789987654321",
  "certValidityNotBefore": "20240108150912",
  "certValidityNotAfter": "20240109150912",
  "certValidity": 1
}
```

或

```
{
  "code": 41000,
  "message": "token 无效",
  "uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",
  "data": null
}
```

## 四、 文件签署

### 4.1 接口说明

地址: /kaifangqian/openAPI/document/sign

请求类型: POST

备注:

1、该接口属于本地化调用接口，不会将签署相关的信息发送到开放签云平台。

2、签署文件类型: PDF 文件;

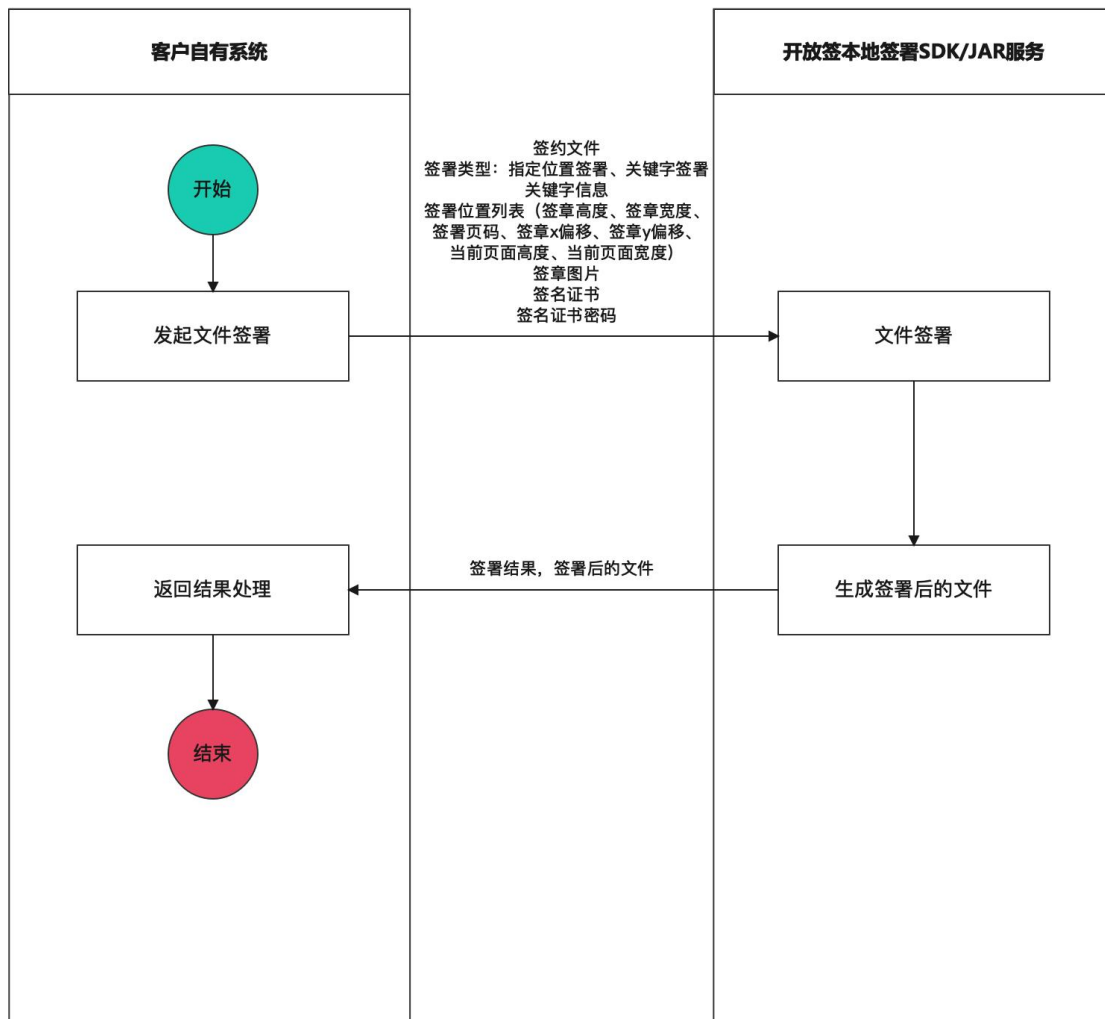
3、签名算法: RSA 国际标准算法

4、签署类型: 关键字签署、指定签署位置签署;

- 关键字签署: 在文档中查找对应关键字的位置进行签署, 若关键字在文件中存在多个, 则在多处关键字位置上均会进行签名;
- 指定位置签署: 在文档的指定位置进行签名。



## 4.2 调用流程



## 4.3 请求报文

data: 文件签署信息 JSON 格式字符串, 包含以下文件签署信息项

字段 ID	字段名称	类型	必传	备注
documentFile	待签署文件	string	否	待签署文件的 base64 格式 文件格式: PDF 建议文件大小不超过 50MB
documentNetworkURL	待签署文件网络地址	string(1000)	否	待签署文件的网络地址, 可通过该地址直接下载待签署文件

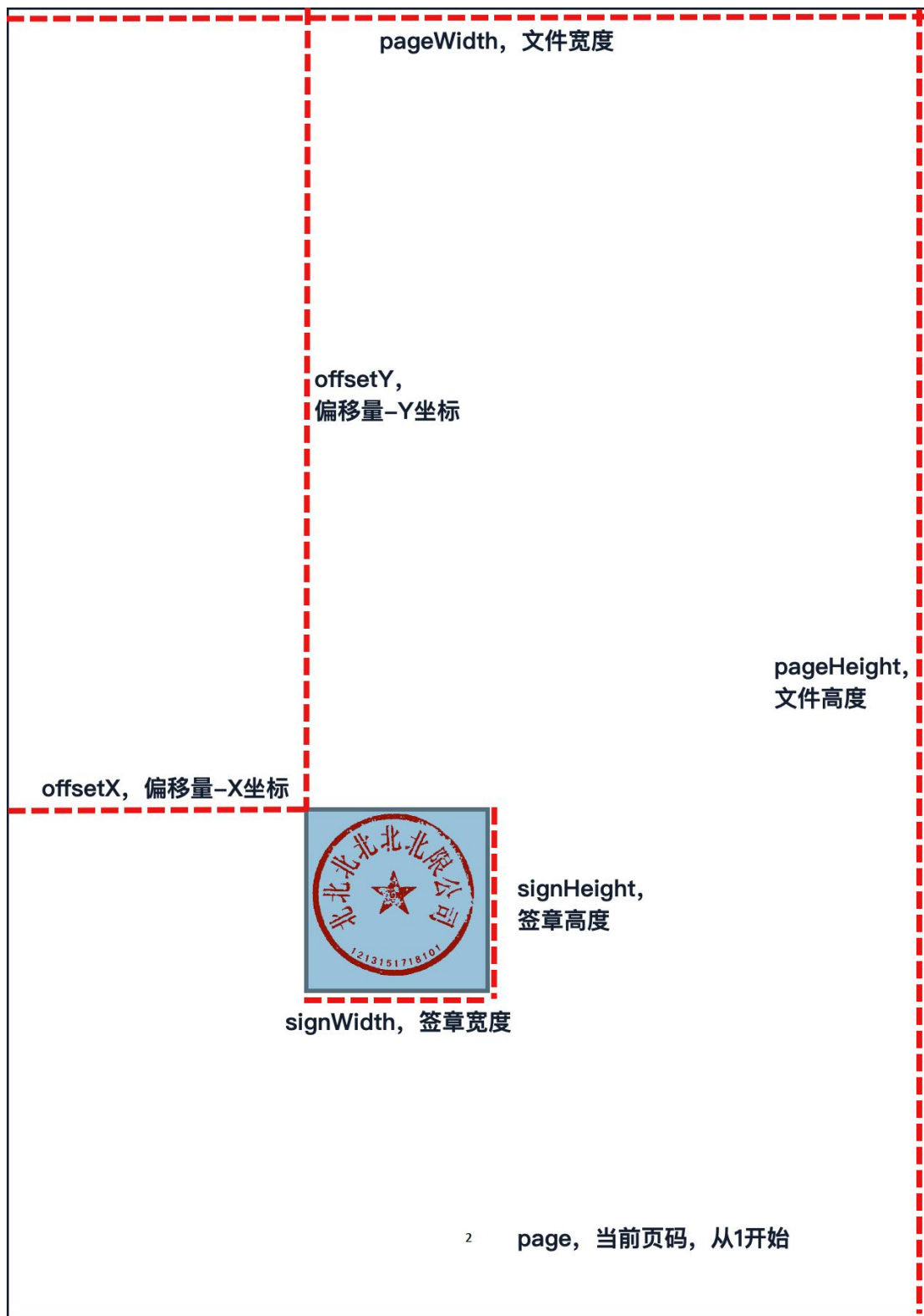
				文件格式：PDF  注意：documentFile 与 documentNetworkURL 两个参数至少有一个传值，若均传值，则优先使用 documentNetworkURL
signType	签署类型	integer(1)	是	1:关键字签署； 2:指定位置签署
keywords	关键字	string(50)	否	当 signType 为 1 时，必传
signLocationList	签署位置集合	array[SignLocation]	否	当 signType 为 2 时，必传
signatureFile	签章图片	string	否	签章图片的 base64 格式 文件格式:支持 JPG(JPEG), BMP, PNG, GIF, TIFF 建议文件大小限制: <1MB
signatureNetworkURL	签章图片网络地址	string(1000)	否	签章图片的网络地址,可通过该地址直接下载签章图片 文件格式:支持 JPG(JPEG), BMP, PNG, GIF, TIFF 建议文件大小限制: <1MB 注意: signatureFile 与 signatureNetworkURL 两个参数至少有一个传值，若均传值，则优先使用 signatureNetworkURL
pfx	签名证书	string	否	签名证书的 base64 格式 文件格式: pfx
pfxNetworkURL	签名证书网络地址	string(1000)	否	签名证书的网络地址,可通过该地址直接下载签名证

				<p>书</p> <p>文件格式: pfx</p> <p>注意: pfx 与 pfxNetworkURL 两个参数至少有一个传值, 若均传值, 则优先使用 pfxNetworkURL</p>
certPassword	证书密码	string(32)	是	<p>签发证书时生成的证书密码, 在本接口中用于证书签名</p>

SignLocation 签署位置数据格式:

字段 ID	字段名称	类型	必传	备注
pageWidth	文件宽度	number(4)	是	签署文件的宽度
pageHeight	文件高度	number(4)	是	签署文件的高度
page	页码	integer(4)	是	签署位置所在页码
offsetX	偏移量-X 坐标	number(4)	是	偏移量, 签名位置 X 坐标(左上角)
offsetY	偏移量-Y 坐标	number(4)	是	偏移量, 签名位置 Y 坐标(左上角)
signWidth	签章宽度	number(4)	是	签章加盖到文件上的宽度
signHeight	签章高度	number(4)	是	签章加盖到文件上的高度

签署位置计算示例：



请求报文示例：

```
{  
  "uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",  
}
```

```
{
  "documentFile": "MIIKoAYJKoZIhvcNAQcCoIIKkTCCCo0CAQExUQ0EwHh...",
  "documentNetworkURL": "",
  "signType": 1,
  "keywords": "甲方盖章处",
  "signLocationList": [
    {
      "pageWidth": 800,
      "pageHeight": 1131,
      "page": 1,
      "offsetX": 91,
      "offsetY": 63,
      "signWidth": 160,
      "signHeight": 160
    },
    {
      "pageWidth": 800,
      "pageHeight": 1131,
      "page": 3,
      "offsetX": 209,
      "offsetY": 32,
      "signWidth": 160,
      "signHeight": 80
    }
  ],
  "signatureFile": "MIIKoAYJKoZIhvcNAQcCoIIKkTCCCo0T09UQ0EwHh...",
  "signatureNetworkURL": "",
  "pfx": "MIIKoAYJKoZIhvcNAQcCoIIKkTCCCQ0EwHh...",
  "pfxNetworkURL": "",
  "certPassword": "QExADALBgkqhkiG9w0BB"
}
```

## 4.4 响应报文

字段 ID	字段名称	类型	必返	备注
documentFile	签署后的文件	string	否	签署后的文件的 base64 格式

响应报文示例：

```
{
  "code": 10000,
  "message": "成功",
  "uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",
  "data": {
```

```
"documentFile": "MIIKoAYJKoZIhvcNAQcCoIIKkTCCCo0CAQPT..."
}
```

或

```
{
  "code": 31000,
  "message": "业务处理失败,证书已失效,无法签署",
  "uniqueCode": "debc5486-3140-4631-b20b-7019d9ce66dc",
  "data": null
}
```

## 五、 SDK 集成

### 5.1 SDK 说明

SDK 目前支持 java 版本，如使用的非 java 语言，请直接调用开放签本地签署 jar 服务。

java jdk 版本 1.8.x ，如果使用低版本，请自行解决语法报错的问题。

### 5.2 接入步骤

1. 获取 SDK 源码，将源码进行编译，生成 SDK 的 jar，将 jar 放入 maven 私服或者 install 到本地 maven 仓库。

```
# 将本地 jar 导入本地仓库
mvn install:install-file -Dfile=target/kaifangqian-sdk.jar -DartifactId=kaifangqian-sdk -DgroupId=org.resrun -Dversion=1.1 -Dpackaging=jar
```

```
将本地 jar 导入本地仓库
<dependency>
  <groupId>org.resrun</groupId>
  <artifactId>kaifangqian-sdk</artifactId>
  <version>1.1</version>
</dependency>
```

2. 初始化配置

SpringBoot 配置方式

```
第一步
# application.yml 中新增如下配置，在调用正式 CA 证书时，请更新一下配置
```

#### Kaifangqian:

```
#签发CA 证书路径
cert-apply-url: https://localhost/service/cert/event
#授权 token
token: 123456
# 默认 false 签发本地测试证书    true 签发CA 证书
prod: false
```

#### 第二步

```
# 初始化 SDKClientConfig
@Configuration
public class SpringBeanManage {
    @Bean
    @ConfigurationProperties(prefix = "kaifangqian")
    public SDKClientConfig clientConfig(){
        return new SDKClientConfig();
    }
}
```

#### Spring 配置方式

```
<!-- applicationContext.xml 中新增如下配置 -->
<bean id="clientConfig" class="org.resrun.sdk.config.SDKClientConfig">
<!-- 默认 false 签发本地测试证书    true 签发CA 证书 -->
<property name="prod" value="false" />
<!-- 授权 token -->
<property name="token" value="${token}" />
<!-- 证书申请路径 -->
<property name="certApplyUrl" value="${certApplyUrl}" />
</bean>
```

## 5.3 SDK 依赖

```
<dependency>
<!-- 版本根据自身 spring-boot 而定 -->
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-autoconfigure</artifactId>
</dependency>
<!-- pdf 签名 -->
<dependency>
<groupId>org.apache.pdfbox</groupId>
<artifactId>pdfbox-tools</artifactId>
<version>3.0.1</version>
```

```
</dependency>
<!-- 加密相关 -->
<dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcpkix-jdk15on</artifactId>
    <version>1.70</version>
</dependency>
<dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcprov-ext-jdk15on</artifactId>
    <version>1.70</version>
</dependency>
<!-- http 工具类 -->
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.13</version>
</dependency>
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpcore</artifactId>
    <version>4.4.13</version>
</dependency>
<!-- Json 处理-->
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.83</version>
</dependency>
```

## 5.4 接口调用

```
#注入 sdkService
@Autowired
private SDKService sdkService;

#调用证书颁发参数与API 请求参数一致，请参照对应的API 请求报文
sdkService.certEvent(certEventRequest)

#调用证书颁发参数与API 请求参数一致，请参照对应的API 请求报文
sdkService.documentSign(documentSignRequest)
```



## 5.5 获取结果

sdk 响应结果与 API 结果一致，请参照对应的 API 响应报文。

资源库-开放签

## 附录

### 返回码表

10000	成功
21000	`\${参数名}` 参数缺失
22000	`\${参数名}` 参数不能为空
23000	<p>`\${参数名}` 参数格式不正确, {reason}</p> <p>reason:</p> <ol style="list-style-type: none"> <li>1、长度不合法</li> <li>2、文件格式不支持</li> <li>3、文件超出限制大小</li> <li>4、超出枚举范围</li> <li>5、不能包含空格</li> <li>6、时间格式不正确</li> <li>9、数值范围不合法</li> <li>10、时间范围不合法</li> </ol>
31000	<p>业务处理失败, {reason}</p> <p>reason:</p> <ol style="list-style-type: none"> <li>1、证书已失效, 无法签署</li> <li>2、证书解析失败, 请检查证书和密码</li> <li>3、文件签署失败</li> <li>4、签署关键字在文件中不存在, 无法签署</li> <li>5、`\${参数名}` 文件网络地址下载失败</li> <li>6、`\${参数名}` 文件格式不正确或文件解析失败</li> <li>7、page 页码不合规, 规范: <math>0 &lt; \text{page} \leq \text{当前 pdf 文件的页数}</math></li> <li>8、signWidth 参数超出范围, 该值需要小于 pageWidth</li> <li>9、signHeight 参数超出范围, 该值需要小于 pageHeight</li> <li>10、offsetX 参数超出范围, 该值需要小于 pageWidth</li> <li>11、offsetY 参数超出范围, 该值需要小于 pageHeight</li> </ol>

	12、业务处理失败, 签署文件 documentFile 和 documentNetworkURL 不能同时为空;  13、业务处理失败, 签名证书 pfx 和 pfxNetworkURL 不能同时为空;  14、业务处理失败, 签章图片 signatureFile 和 signatureNetworkURL 不能同时为空;
41000	token 无效
42000	token 中未授权该接口调用权限
43000	调用证书签发服务超时
51000	请勿 SQL 攻击
52000	请勿跨站点攻击
91000	系统未知错误