

法大大——

## 《法大大电子合同 API 购买及使用指南》



## 使用说明

### 一、 客服帮助联系方式

用户在购买及使用流程中,有任何疑问,请及时联系客服(0755-23289719)

微信: 15116397749) 协助完成。

### 二、 API 产品组成部分及购买说明

本 API 产品由以下 **5 部分** 组合而成, 需分别购买后组合使用:

1) **【法大大电子合同 API 定制版—企业认证包】**

购买地址: <https://market.cloud.tencent.com/products/16837>



2) **【法大大电子合同 API 定制版—个人认证包】**

购买地址: <https://market.cloud.tencent.com/products/16257>



### 3) 【法大大电子合同 API 定制版—签署包】

购买地址：<https://market.cloud.tencent.com/products/16839>



### 4) 《法大大实名认证电子合同方案》API 文档.doc

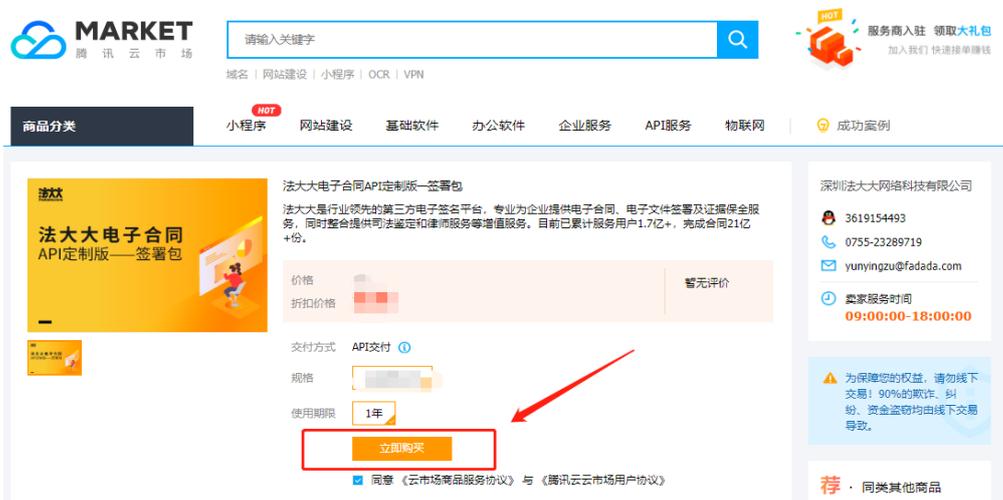
获取方式：于本指南第四部分，或联系法大大客服，或专属对接团队处获取；

### 5) 【法大大分配 app\_id 和 app\_secret】

获取方式：专属对接团队处获取

## 三、购买及调用流程

- 1、 登录[腾讯云市场](#)进入法大大电子合同 API 【签署包】、【个人认证包】、【企业认证包】商品页面并完成购买；

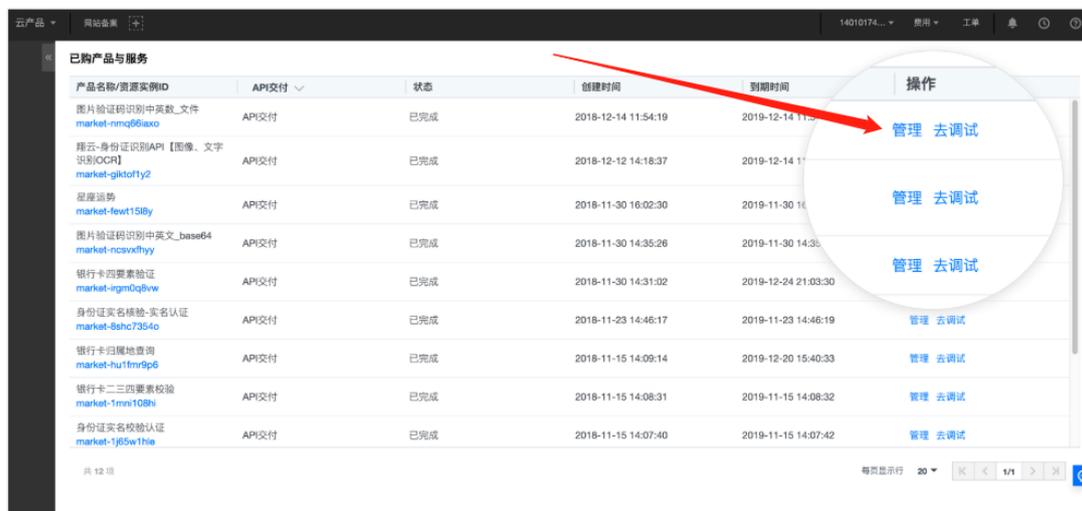


深圳法大大网络科技有限公司  
Shenzhen Fadada Internet Technology Company Limited  
www.fadada.com  
400-869-2012  
深圳市南山区高新科技园高新南1道6号TCL大厦A座607

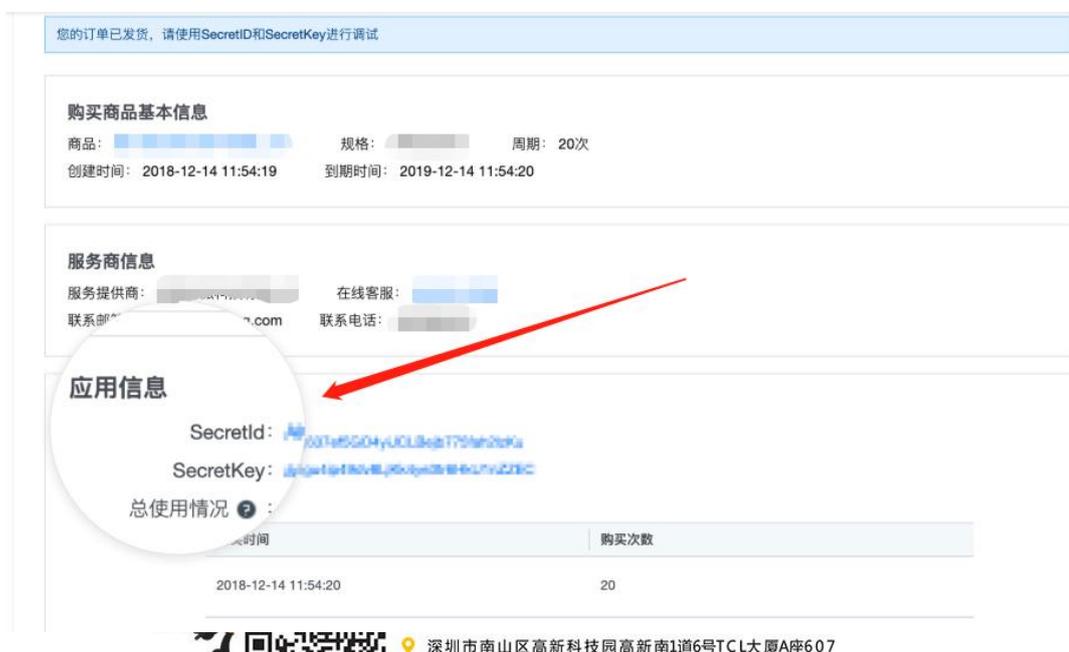
2、 完成购买后，点击进入**控制台**；



3、 在控制台点击**“管理”**，查看详情；



4、 获得对应商品 API 的 **SecretID** 和 **SecretKey** 后，即可按照商品介绍里的方式调用此部分 API；



```

public static void main(String[] args) throws NoSuchAlgorithmException, UnsupportedEncodingException, InvalidKeyException {
    //云市场分配的密钥id
    String secretId = "AK[redacted]CY0Jw18Z";
    //云市场分配的密钥Key
    String secretKey = "i[redacted]7N";
    String source = "market";

    Calendar cd = Calendar.getInstance();
    SimpleDateFormat sdf = new SimpleDateFormat("EEE, dd MMM yyyy HH:mm:ss GMT", Locale.US);
    sdf.setTimeZone(TimeZone.getTimeZone("GMT"));
    String datetime = sdf.format(cd.getTime());
    // 签名
    String auth = calcAuthorization(source, secretId, secretKey, datetime);
    // 请求方法
    String method = "POST";
    // 请求头
    Map<String, String> headers = new HashMap<String, String>();
    headers.put("X-Source", source);
    headers.put("X-Date", datetime);
    headers.put("Authorization", auth);
}

```

5、联系法大大**专属对接团队**，获取【法大大分配 app\_id 和 app\_secret】及《法大大实名认证电子合同方案》API 文档.doc

6、**阅读文档**及使用【法大大分配 app\_id 和 app\_secret】即可调用完整法大大实名认证电子合同 API。

```

// 查询参数
Map<String, String> queryParams = new HashMap<String, String>();

// body参数
Map<String, String> bodyParams = new HashMap<String, String>();

//法大大公共参数
bodyParams.put("app_id", "988888");
bodyParams.put("timestamp", "20191022160303");
bodyParams.put("v", "2.0");
bodyParams.put("msg_digest", "RkJBNjExREUwNUeWmUI4RDUwQUM4OURBm0EzQjY3ODIwQjhDOEYyMA=");

//法大大业务参数
bodyParams.put("account_type", "1");
bodyParams.put("open_id", "afdafafsa888855555");

// url参数拼接
String url = "http://service-i7iwxtym-1255747603.gz.apigw.tencentcs.com/release/api/account_register.api";
if (!queryParams.isEmpty()) {
    url += "?" + urlEncode(queryParams);
}

```



#### 四、 《法大大实名认证电子合同方案》 API 文档

# 法大大实名认证电子合同方案

## Revision History

Name	Date	Reason For Changes
黎伟合	2019/3/25	1.申请实名认证证书接口和自动签接口添加调用接口说明。
黎伟合	2019/3/27	5.16.2 增加认证同步通知描述



## 目录

1.....	引言
8	
1.1. 编写目的.....	8
1.2. 背景.....	8
1.3. 定义.....	8
1.4. 参考资料.....	8
2. 概述.....	8
3. 接口调用时序图.....	9
3.1. 实名认证流程.....	9
3.2 上传合同并签署.....	9
3.2 模板生成合同并签署.....	10
4. 规范（重要！请阅读！！）.....	11
5、接口清单.....	11
5.1 注册账号.....	11
5.2 获取企业实名认证地址.....	12
5.3 获取个人实名认证地址.....	17
5.4 实名证书申请.....	21
5.5 编号证书申请.....	21
5.6 印章上传.....	22
5.7 自定义印章.....	23
5.8 合同上传.....	24
5.9 模板上传.....	26
5.10 模板填充.....	27
5.11 自动签署.....	31
5.12 手动签署接口.....	34
5.13 合同查看.....	36
5.14 合同下载.....	37
5.15 合同归档.....	38
5.16 查询个人实名认证信息.....	错误!未定义书签。
5.17 查询企业实名认证信息.....	错误!未定义书签。



# 1. 引言

## 1.1. 编写目的

为法大大接入平台方开发人员提供一个规范化的接口协议，以实现与法大大平台的互操作，进行业务整合嵌入。

## 1.2. 背景

本协议适用于法大大电子签章平台与第三方接入平台间的行为、数据及事件的交互与传递。本协议承载于 HTTP 协议，严格遵守 HTTP 协议规范。

## 1.3. 定义

CA: Certificate Authority 证书授权机构

## 1.4. 参考资料

RFC2616, HTTP 协议

# 2. 概述

接入方平台需向法大大申请接入（须填写接入申请表格，包括平台域名、服务器 IP，客户实名认证模式等），法大大审核后将给接入平台颁发 app\_id 和 app\_secret。app\_id 为 6 位数字，app\_secret 为 24 位字符串。接入方需提供其服务器 IP 以做绑定。

### 典型业务流程（个人用户）：

1. 客户在接入方平台注册账号
2. 接入方将自己系统的账号同步到法大大注册账号，法大大返回客户编号，接入方需要保存该客户编号
3. 接入方接入法大大实名认证。
  - 3.1 . 请求获取个人或企业实名认证地址接口，法大大返回实名认证地址和实名认证流水号
  - 3.2 . 浏览器打开实名认证地址，根据页面提示填写实名信息进行实名认证
  - 3.3 . 法大大审核通过后通过异步回调的方式通知接入方实名认证结果
  - 3.4 . 接入方根据实名认证流水号查询认证结果和实名信息
4. 接入方调用法大大证书申请接口，绑定客户编号和实名信息，并颁发证书
5. 上传待签署合同
6. 客户在接入平台页面上向法大大发起签章请求，法大大提供签章页面，供客户嵌入接入平台流程或是跳转到该页面。

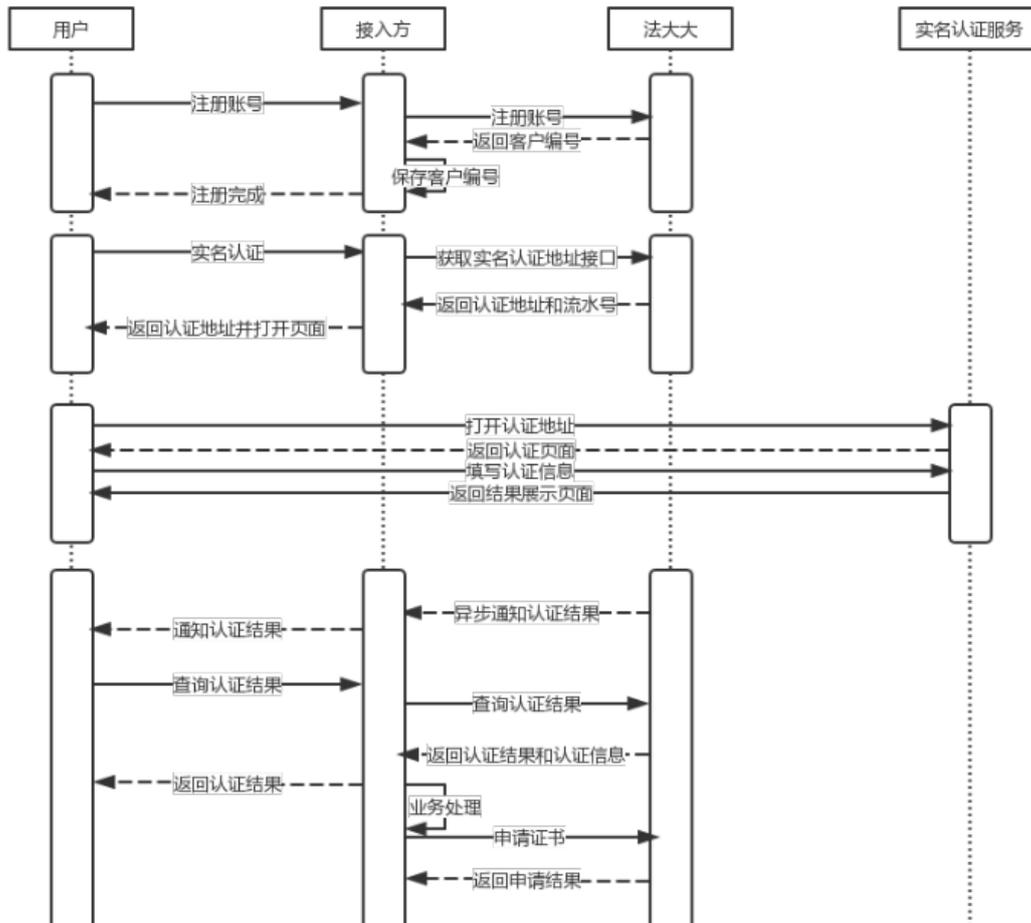


7. 客户在法大大页面上完成电子签章后（需要验证法大大平台下发给客户的签章验证短信），返回接入方平台

### 3. 接口调用时序图

#### 3.1. 实名认证流程

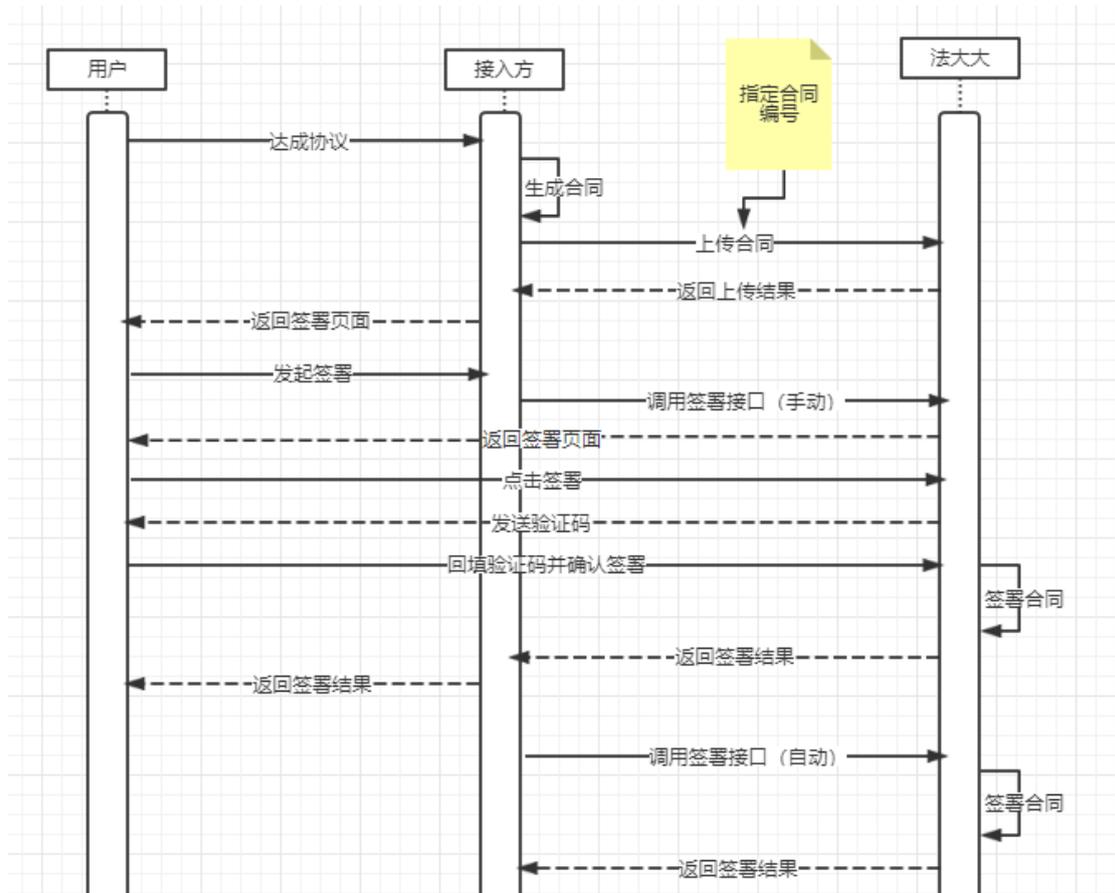
注册账号 ——> 实名认证（个人 / 企业） ——> 证书申请



#### 3.2 上传合同并签署

上传合同 ——> 第一次调用签署接口（传甲方客户编号） ——> 第二次调用签署接口（传乙方客户编号） ——> …… ——> 第 N 次调用签署接口

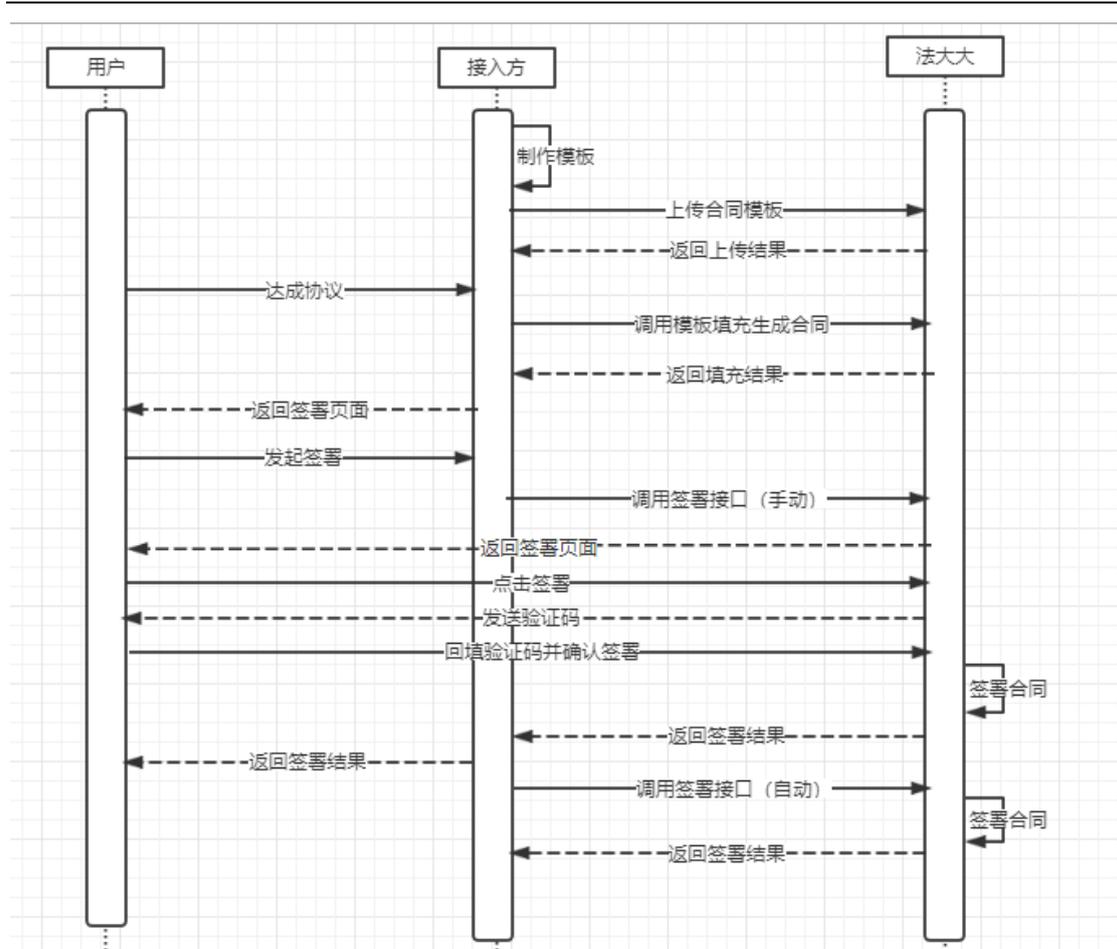




### 3.3 模板生成合同并签署

上传合同模板——>模板填充生成合同——>第一次调用签署接口（传甲方客户编号）——>第二次调用签署接口（传乙方客户编号）——>……——>第 N 次调用签署接口





## 4. 规范（重要！请阅读!!!）

- 1.FDDServer: 法大大接口服务器域名, 参见邮件内容
- 2.Port: 法大大接口服务器端口, 参见邮件内容
- 3.APP\_ID: 接入方的 ID, 参见邮件内容
- 4.APP\_SECRET: 接入方的密钥, 参见邮件内容
- 5.Triple DES (3DES): DESede/ECB/PKCS5Padding; 密钥为 APP\_SECRET
- 6.php 计算摘要时, md5 和 sha1 的加密结果要转大写, 再进行下一步计算
- 7.使用 java sdk 时, url 为 FDDServer:Port, 不用加接口名
- 8.如果发现文档中有错误请在 QQ 对接群中提出, Thank You!

## 5、接口清单

### 5.1 注册账号

接口类型: HTTP POST 请求

接口说明: 此接口针对接入平台已有账号体系, 判断 open\_id 是否存在, 存在则返回对应账号。否则随机生成账号并返回账号



调用方向：接入平台->法大大

接口地址：https://FDDServer:Port/api/account\_register.api

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如：900001
timestamp	String	是	请 求 时 间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret+ account_type+open_id) ) ) 标记红色部分规则：对所有 业务参数进行 ascll 码排序 后拼接	
业务参数				
参数	类型	是否必填	描述	备注
open_id	String	是	用户在接入方的唯一标识	字 符 (len<=64)
account_type	String	是	1:个人, 2:企业	

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
code	String	是	状态码	
msg	String	是	状态描述	
data	String	否		customer_id

## SDK 使用说明

```
// JAVA
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
String open_id = "";//用户在接入方的唯一标识
String account_type = "";//账号类型
String result =base.invokeRegisterAccount(open_id,account_type);
```

返回示例:



正例: {"code":1,"data":"D129A83BE4CBE33C1A64C18EEB230086","msg":"success"}

反例: {"code":1004,"data":null,"msg":"请求参数非法"}

## 5.2 获取企业实名认证地址

接口类型: HTTP POST 请求

接口说明: 此接口返回法大大实名认证页面 url。接入方拿到 url 后在浏览器中打开认证页面

调用方向: 接入平台->法大大

接口地址: https://FDDServer:Port/api/get\_company\_verify\_url.api

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64 ( SHA1 ( app_id +MD5(timestamp) +SHA1(app_secret+ agent_info+bank_info+cert_fl ag+company_info+company _principal_type+customer_id +legal_info+ notify_url+page_modify+resu lt_type+return_url+verified_w ay) ) ) 标记红色部分规则: 对所有业 务参数进行 ascll 码排序后拼 接	
业务参数				
参数	类型	是否必填	描述	备注
customer_id	String	是	客户编号	注册账号时 返回
verified_way	String	否	实名认证套餐类型 0: 标准方案 (对公打款+纸质 审核); 1: 对公打款; 2: 纸质审核,	默认为 0:标准 方案



m_verified_way	String	否	实名认证套餐类型 0:三要素标准方案; 1:三要素补充方案; 2:四要素标准方案; 3:四要素补充方案',	0, 2 标准方案 [运营商三要素或者银行卡四要素是否通过都要继续人脸识别认证]; 1, 3 补充方案 [运营商三要素/银行卡四要素通过的情况下认证成功, 不通过的情况下进行人脸识别认证]
page_modify	String	是	是否允许用户页面修改	1 允许, 2 不允许
notify_url	String	是	回调地址	异步通知认证结果
return_url	String	否	同步通知 url	认证结果返回地址
company_info	CompanyInfo	否	企业信息	
bank_info	BankInfo	否	对公账号信息	
company_principal_type	String	否	企业负责人身份: 1. 法人, 2. 代理人	默认是 1 法人
legal_info	LegalInfo	否	法人信息	company_principal_type=1 时 必选
agent_info	AgentInfo	否	代理人信息	company_principal_type=2 时 必选
result_type	String	否	刷脸是否显示结果页面 参数值为“1”: 直接跳转到 return_url 或法大大指定页面, 参数值为“2”: 需要用户点击确认后跳转到 return_url 或法大大指定页面	
cert_flag	String	否	是否认证成功后自动申请实名认证证书 参数值为“0”: 不申请,	



			参数值为“1”：自动申请	
m_verified_way	String	否	实名认证套餐类型 0:三要素标准方案; 6 补充三要素方案+人工审核	
option	String	否	add (新增)、modify (修改)	不传默认 add
verified_serialNo	String	否	管理员认证流水号	
authorization_file	File	否	企业注册申请表	

#### AgentInfo

参数	类型	是否必填	描述	备注
agent_name	String	否	代理人姓名	
agent_id	String	否	代理人证件号	
agent_mobile	String	否	代理人手机号	
agent_id_front_path	String	否	代理人证件正面照下载地址	
bank_card_no	String	否	代理人银行卡号	

#### BankInfo

参数	类型	是否必填	描述	备注
bank_name	String	否	银行名称	
bank_id	String	否	银行帐号	
subbranch_name	String	否	开户支行名称	

#### CompanyInfo

参数	类型	是否必填	描述	备注
company_name	String	否	企业名称	
credit_no	String	否	统一社会信用代码	
credit_image_path	String	否	统一社会信用代码证件照路径	

#### LegallInfo

参数	类型	是否必填	描述	备注
legal_name	String	否	法人姓名	
legal_id	String	否	法人证件号	
legal_mobile	String	否	法人手机号(仅支持国内运营商)	
legal_id_front_path	String	否	法人证件正面照下载地址	
bank_card_no	String	否	法人银行卡号	

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
----	----	------	----	----



code	String	是	状态码	
msg	String	是	状态描述	
data	String	否	认证序列号和编码后的认证 url 地址	

data

参数	类型	是否必填	描述	备注
transactionNo	String	是	交易号 (需要保存, 用于证书申请和实名认证查询)	
url	String	是	地址 (建议接入方保存)	需要 Base64 解码

## SDK 使用说明

```
// JAVA
GetCompanyVerifyUrl          comverify          =          new
GetCompanyVerifyUrl(APP_ID,APP_SECRET,V,HOST);
    CompanyInfoINO companyInfo = new CompanyInfoINO();//企业信息
    companyInfo.setCompany_name("");//企业名称
    companyInfo.setCredit_no("");//统一社会信用代码
    companyInfo.setCredit_image_path("");//统一社会信用代码证件照路径

    BankInfoINO bankInfo = new BankInfoINO();//对公账号信息
    bankInfo.setBank_name("");//银行名称
    bankInfo.setBank_id("");//银行帐号
    bankInfo.setSubbranch_name("");//开户支行名称

    LegallInfoINO legallInfo = new LegallInfoINO();//法人信息
    legallInfo.setLegal_name("");//法人姓名
    legallInfo.setLegal_id("");//法人证件号(身份证)
    legallInfo.setLegal_mobile("");//法人手机号(仅支持国内运营商)
    legallInfo.setLegal_id_front_path("");//法人证件正面照下载地址
    AgentInfoINO agentInfo = new AgentInfoINO();//代理人信息
    agentInfo.setAgent_name("");//代理人姓名
    agentInfo.setAgent_id("");//代理人身份证号码
    agentInfo.setAgent_mobile("");//代理人手机号
    agentInfo.setAgent_id_front_path("");//代理人证件正面照下载地址
    String customer_id = "";//客户编号 必填
```

/\*\*



```

* 实名认证套餐类型
* 0: 标准方案 (对公打款+纸质审核) 默认 0;
* 1: 对公打款;
* 2: 纸质审核;
*/
String verified_way = "";实名认证套餐类型
String page_modify = "";//是否允许用户页面修改
String company_principal_type = "";//1.法人, 2 代理人
String notify_url = "";//回调地址
String return_url = "";//同步通知 url
String result = comverify.invokeCompanyVerifyUrl(companyInfo,bankInfo,legallInfo
,agentInfo, customer_id,verified_way,page_modify,
company_principal_type,return_url,notify_url);

```

返回示例:

```

正例:
{"code":1,"data":{"transactionNo":"10ccfd3a7f67488c9c391b7ebd6c61ae","url":"aHR0c***jg5MDk1ODM="},"msg":"success"}
反例: {"code":1004,"data":null,"msg":"异步通知地址不能为空"}

```

## 5.3 获取个人实名认证地址

接口类型: HTTP POST 请求

接口说明: 此接口返回法大大实名认证页面 url。接入方拿到 url 后在浏览器中打开认证页面

调用方向: 接入平台->法大大

接口地址: [https://FDDServer:Port/api/get\\_person\\_verify\\_url.api](https://FDDServer:Port/api/get_person_verify_url.api)

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secre tbank_card_no+cert_flag+c ert_type+customer_id+cust	



			<p>omer_ident_no+customer_id dent_type+customer_name +ident_back_path+ident_fr ont_path+mobile+notify_u rl+page_modify+result_typ e+return_url+verified_way) ) )</p> <p>标记红色部分规则: 对所有 业务参数进行 ascll 码排序 后拼接</p>	
业务参数				
参数	类型	是否必填	描述	备注
customer_id	String	是	客户编号	注册账号时返回
verified_way	String	是	实名认证套餐类型 0:三要素标准方案; 1:三要素补充方案; 2:四要素标准方案; 3:四要素补充方案,	0, 2 标准方案[运营 商三要素或者 银行卡四要素是 否通过都要继续 人脸识别认证]; 1, 3 补充方案[运 营商三要素/银 行卡四要素通过 的情况下认证成功, 不通过的情况 下进行人脸识别 认证]
page_modify	String	是	是否允许用户页面修改	1 允许, 2 不允许
notify_url	String	是	回调地址	异步通知认证结 果
return_url	String	否	同步通知 url	认证结果返回地 址
customer_name	String	否	姓名	
customer_ident_type	String	否	是否支持其他证件类型	身份证-0 其他 1
customer_ident_no	String	否	证件号码 cert_type=0 身份证号, cert_type=1 护照号, cert_type=B 港澳居民 来往内地通行证号, cert_type=C 台湾居民 来往大陆通行证号	
mobile	String	否	手机号码	



ident_front_path	String	否	证件正面照下载地址 cert_type=0:身份证正面 cert_type=1:护照带人像图片 cert_type=B:港澳居民来往内地通行证带人像图片 cert_type=C:台湾居民来往大陆通行证带人像图片	
ident_back_path	String	否	证件反面照下载地址 cert_type=0:身份证反面 cert_type=1:护照封图片 cert_type=B:港澳居民来往内地通行证封图图片 cert_type=C:台湾居民来往大陆通行证封图图片	
result_type	String	否	刷脸是否显示结果页面 参数值为“1”: 直接跳转到 return_url 或法大大指定页面, 参数值为“2”: 需要用户点击确认后跳转到 return_url 或法大大指定页面	
cert_flag	String	否	是否认证成功自动申请实名认证 参数值为“0”: 不申请, 参数值为“1”: 自动申请	
option	String	否	add (新增)、modify (修改)	不传默认 add
cert_type	String	否	证件类型: 0: 身份证; 1: 护照; B: 港澳居民来往内地通行证, C: 台湾居民来往大陆通行证 (默认为 0, 仅当支持其他证件时, 证件类型 1/B/C 接口允许同步传	



			参)	
bank_card_no	String	否	个人银行卡	

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
code	String	是	状态码	
msg	String	是	状态描述	
data	String	否	认证序列号和编码后的认证 url 地址	

data

参数	类型	是否必填	描述	备注
transactionNo	String	是	交易号 (需要保存, 用于证书申请和实名认证查询)	
url	String	是	地址 (建议接入方保存)	需要 Base64 解码

## SDK 使用说明

```
// JAVA
GetPersonVerifyUrl personverify = new GetPersonVerifyUrl(APP_ID,APP_SECRET,V,HOST);
String customer_id = "";//客户编号
String verified_way = "";//实名认证套餐类型
String page_modify = "";//是否允许用户页面修改
String notify_url = "";//回调地址
String return_url= "";//同步通知 url
String customer_name = "";//姓名
String customer_ident_type = "";//证件类型
String customer_ident_no ="";//证件号码
String mobile ="";//手机号码
String ident_front_path ="";//证件正面照下载地址
String result = personverify.invokePersonVerifyUrl(customer_id,verified_way,
page_modify,notify_url,return_url,customer_name,customer_ident_type,
customer_ident_no,mobile,ident_front_path);
```

返回示例:

正例:

```
{"code":1,"data":{"transactionNo":"9db6660a50c54648bba99ffd1f4902f8","url":"aHR0c***wPTE1NDY5MzAzMDY4ODk="},"msg":"success"}
```

反例: {"code":1004,"data":null,"msg":"异步通知地址不能为空"}



## 5.4 实名认证申请

接口类型：HTTP POST/GET 请求

接口说明：调用此接口可以给相关账号颁发证书。新增平台配置控制是否允许颁发实名认证证书

调用方向：接入平台->法大大

接口地址：https://FDDServer:Port/api/apply\_cert.api

输入参数：

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如：900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret+ customer_id+verified_ serialno) ) ) 标记红色部分规则：对所有业务 参数进行 ascll 码排序后拼接	
业务参数				
参数	类型	是否必填	描述	说明
customer_id	String	是	客户编号	注册账号时返回
verified_serialno	String	是	实名认证序列号	获取实名认证地址时返回

输出参数 (JSON 格式数据)：

参数	类型	是否必填	描述	备注
code	String	是	状态码	
msg	String	是	状态描述	
data	String	否		

## SDK 使用说明

```
// JAVA
ApplyCert applyCert = new ApplyCert(APP_ID,APP_SECRET,V,HOST);
String customer_id = "";//客户编号
String verified_serialno= "";//实名认证序列号
```



```
String result = applyCert.invokeApplyCert(customer_id,verified_serialNo);
```

返回示例:

```
正例: {"code":1,"msg":"success"}
反例: {"code":3201,"msg":"不能重复申请证书"}
```

## 5.5 印章上传

接口类型: HTTP POST 请求

接口说明: 新增用户签章图片

调用方向: 接入平台->法大大

接口地址: [https://FDDServer:Port/api/add\\_signature.api](https://FDDServer:Port/api/add_signature.api)

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号 2.0	
msg_digest	String	是	Base64( SHA1(app_id+ md5(timestamp) +SHA1(app_secret+custo mer_id+signature_img_base64) ) ) 标记红色部分规则: 对所有业务 参数进行 ascll 码排序后拼接	
业务参数				
参数	类型	是否必填	描述	备注
customer_id	String	是	客户编号	
signature_img_base64	String	是	签章图片 base64	

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
code	String	是	状态码	
msg	String	是	状态描述	
data	String	否		

data

参数	类型	要求	描述	备注
signature_id	String	可选	签章图片 ID	
signature_sub_info	String	可选	扩展信息, 目前为空	



## SDK 使用说明

```
// JAVA
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
String customer_id = "";//客户编号
File imgfile = new File("");//签章图片路径 imgUrl 和 imgfile2 选 1
String imgUrl = "";//签章图片公网地址 imgUrl 和 imgfile2 选 1
String result = base.invokeaddSignature(customer_id,imgfile, imgUrl);

FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
String customer_id = "";//客户编号
String signature_img_base64 = "";//签章图片 base64
String result = base.invokeaddSignature(customer_id,signature_img_base64);
```

返回示例:

```
正例: {"code":1,"data":{"signature_id":"4309624","signature_sub_info":null},"msg":"success"}
反例: {"code":1302,"data":null,"msg":"尚未实名认证"}
```

## 5.6 自定义印章

接口类型: HTTP POST 请求

接口说明: 获取用户自定义签章图片

调用方向: 接入平台->法大大

接口地址: [https://FDDServer:Port/api/custom\\_signature.api](https://FDDServer:Port/api/custom_signature.api)

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1(app_id+ md5(timestamp) +SHA1(app_secret+conte nt +customer_id) ) ) 标记红色部分规则: 对所有业务 参数进行 ascll 码排序后拼接	
业务参数				
参数	类型	是否必填	描述	备注



customer_id	String	是	客户编号	注册账号时返回
content	String	是	印章展示的内容	

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
code	String	是	状态码	
msg	String	是	状态描述	
data	String	否		

Data

参数	类型	要求	描述	备注
signature_img_base64	String	必选	印章图片 base64	

## SDK 使用说明

```
// JAVA
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
String customer_id = "";//客户编号
String content = "";//印章展示的内容
String result = base.invokecustomSignature(customer_id,content);
```

返回示例:

```
正例: {"code":1,"data":{"signature_img_base64":"iVBORw0.....IFTkSuQmCC"},"msg":"success"}
反例: {"code":1004,"data":null,"msg":"请求参数非法"}
```

## 5.7 合同上传

接口类型: HTTP POST 请求

接口说明: 接入平台将待签署的 pdf 文档通过此接口将文档传输到法大大, 供签署时使用, 请参考“3.2 上传合同并签署”流程。

调用方向: 接入平台->法大大

接口地址: https://FDDServer:Port/api/uploaddocs.api

接口返回: JSON 报文

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0



msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp ) +SHA1(app_secret+ contract_id) ) ) 标记红色部分规则：对所有业务参数进行 ascll 码排序后拼接	
业务参数				
参数	类型	是否必填	描述	备注
contract_id	String	是	合同编号	只允许长度 <=32 的英文或数字字符
doc_title	String	是	合同标题	如“xx 投资合同”
doc_url	String	是	文档地址	文档地址 doc_url 和 file 两个参数必选一
file	File	是	PDF 文档	File 文件 doc_url 和 file 两个参数必选一
doc_type	String	是	文档类型	.pdf

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
result	String	是	处理结果	success: 成功 error: 失败
code	String	是	状态码	1000: 操作成功 2001: 参数缺失或者不合法 2002: 业务异常, 失败原因见 msg 2003: 其他错误, 请联系法大大
msg	String	是	描述	

## SDK 使用说明

```
// JAVA
```

```
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
```



```
String contract_id = ""; //自定义合同 id
String doc_title = ""; //合同标题
String doc_url = ""; //合同公网下载地址
String doc_type = ".pdf"; //合同类型 目前仅支持 pdf 格式
String result = base.invokeUploadDocs(contract_id, doc_title, file, doc_url,
doc_type);
```

返回示例:

```
正例: {"code":"1000","msg":"操作成功","result":"success"}
反例: {"code":"2002","msg":"无法获取文档(file 与 doc_url 皆为空).","result":"error"}
```

## 5.8 模板上传

接口类型: HTTP MULTI-PART POST 请求

接口说明: PDF 模板制作参见[使用 Acrobat 制作 PDF 模板说明.pdf](#)。

此接口与[模板填充接口](#)配合使用, 接入方预先将制作好的 PDF 模板通过此接口上传到法大大, 后续如需要签署合同时只需要将需填充的内容通过[模板填充接口](#)传入, 即可生成合同供签署操作, 请参考“3.3 模板生成合同并签署”流程。

调用方向: 接入平台->法大大

接口地址: https://FDDServer:Port/api/uploadtemplate.api

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64 ( SHA1 ( app_id +MD5(timestamp) +SHA1(app_secret+ <b>template_id</b> ) ) 标记红色部分规则: 对所有业务参数进行 ascll 码排序后拼接	
业务参数				
参数	类型	是否必填	描述	说明
template_id	String	是	模板编号	只允许长度 <=32 的英文或数字字符
doc_url	String	是	文档地址	字段类型: 字符



				串, 须为 URL doc_url 和 file 两个参数必选一
file	File	是	PDF 模板	字段类型: 文件流 doc_url 和 file 两个参数必选一

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
result	String	是	处理结果	success: 成功 error: 失败
code	String	是	状态码	1: 操作成功 2001: 参数缺失或者不合法 2002: 业务异常, 失败原因见 msg 2003: 其他错误, 请联系法大大
msg	String	是	描述	

## SDK 使用说明

```
// JAVA
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
String doc_url = ""; //模板公网下载地址
String template = ""; //自定义模板 id
String result = base.invokeUploadTemplate(template_id, file, doc_url);
```

返回示例:

```
正例: {"code": "1", "msg": "上传成功!", "result": "success"}
反例: {"code": "2002", "msg": "文件类型错误", "result": "error"}
```

## 5.9 模板填充

接口类型: HTTP POST 请求

接口说明: 将需填充的内容通过模板填充接口传入, 即可生成合同供签署操作, 请参考“3.3 模板生成合同并签署”流程。

调用方向: 接入平台->法大大

接口地址: [https://FDDServer:Port/api/generate\\_contract.api](https://FDDServer:Port/api/generate_contract.api)

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注



app_id	String	是	接入方 ID 如：900001	
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号 2.0	
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret+template_id+contract_id) + parameter_map ) ) 标记红色部分规则：对所有业务参数进行 ascii 码排序后拼接	
<b>业务参数</b>				
参数	类型	是否必填	描述	备注
doc_title	String	是	文档标题	如“xx 投资合同”
template_id	String	是	模板编号	
contract_id	String	是	合同编号	只允许长度<=32 的英文或数字字符
font_size	String	是	字体大小	如传入该参数，则按照该参数值设置字体大小；参考 word 字体设置，例如：10,12,12.5,14； 若未传入该参数，则文本域字体大小按照文本域设置时的大小（文本域设置为自动，则由系统计算字体大小）
font_type	String	是	字体类型	0-宋体；1-仿宋；2-黑体；3-楷体；4-微软雅黑
parameter_map	String	是	填充内容	<b>JsonObject 字符串</b> key 为文本域，value 为要填充的值，value 值传字符类型 示例： {"platformName":"The Earth","borrower":"Boss Horse"}
dynamic_tables	String	否	动态表格	<b>JSONArray[dynamic_table]字符串</b> 每个 dynamic_table 元素表示一个动态表单，可以生成多个表格。 示例（包含两个表格）： [{"datas":["1","小网","小易","1000"],["2","小云","小音","2000"],["3","小乐","天马","3000"],"headers":["序号","借款人","贷款人","金额"],"colWidthPercent":[1,4,4,4],



				"cellHeight":30,"cellHorizontalAlignm ent":0,"cellVerticalAlignment":5,"pag eBegin":1,"theFirstHeader":"附一"]}
--	--	--	--	---

dynamic_table				
参数	类型	是否必填	描述	备注
insertWay	Integer	否	动态表格插入方式	0: 新建页面添加table (默认) 1: 在某个关键字后添加 table
keyword	String	否	关键字方式插入动态表格	#1. 当insertWay为1时, 必填 #2. 要求该关键字后 (当前页) 必须不包含内容, 否则会被覆盖 #3. 若关键字为多个, 则取第一个关键字, 在此关键字后插入 table
pageBegin	Integer	否	表格需要插入的页数	#1. 当insertWay为0时, 必填 #2. 表示从第几页开始插入表格, 如要从末尾插入table,则pageBegin为pdf总页数加1 #3. 多个表格指定相同pageBegin, 则多个表格按顺序插入, 一个表格新起一页 #4. pageBegin 为-1 时, 则从 pdf 末尾插入 table
borderFlag	boolean	否	table是否有边框	true: 有 (默认) false: 无边框
cellHeight	Float	否	正文行高 (表头不受影响)	单位: pt, 即 point, 等于 1/72 英寸
cellHorizontalAlignment	Integer	否	Table 中每个单元的水平对齐方式	(0: 居左; 1: 居中; 2: 居右) 默认为 0
cellVerticalAlignment	Integer	否	Table 中每个单元的垂直对齐方式	(4: 居上; 5: 居中; 6: 居下) 默认为 4
theFirstHeader	String	否	表头上方的一级标题	
headers	Array[String]	是	表头信息	
headersAlignment	Integer	否	表头对齐方式	(0 居左;1 居中;2 居右) 默认 0
datas	Array[Array[String]]	否	正文	(外层表示行, 内层表示列)



	ing]]			
colWidthPer cent	Array[integer]	否	各列宽度比例	默认值：各列 1:1
tableHorizontalAlignment	Integer	否	table 的水平对齐方式	(0 居左, 1 居中, 2 居右)默认 1
tableWidthPercentage	Float	否	table 宽度的百分比	(0<tableWidthPercentage<=100)默认为 100.0
tableHorizontalOffset	Float		设置 table 居左居中居右后的水平偏移量	(向右偏移值为正数, 向左偏移值为负数)默认为 0.0, 单位 px(像素)

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
result	String	是	处理结果	success: 成功 error: 失败
code	String	是	状态码	1000: 操作成功 2001: 参数缺失或者不合法 2002: 业务异常, 失败原因见 msg 2003: 其他错误, 请联系法大大
msg	String	是	描述	
download_url	String	否	合同下载地址	填充成功时有值, 填充失败无值
viewpdf_url	String	否	合同查看地址	填充成功时有值, 填充失败无值

## SDK 使用说明

```
// JAVA
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
String font_size = "";//字体大小
String font_type = "";//字体类型
String paramter = "";//填充内容
paramter = getparamter();
String dynamic_tables = "";//动态表格
dynamic_tables = getdynamic_tables();
String result = base.invokeGenerateContract(template_id, contract_id, "",
font_size, font_type, paramter, dynamic_tables);

private String getparamter() {
```



```
        JSONObject paramter = new JSONObject();
//        paramter.put("homeUrl","√");
        paramter.put("borrower","深圳法大大");
        return paramter.toString();
    }
    private String getdynamic_tables() {
        JSONArray dynamic_tables = new JSONArray();
        JSONObject dynamic2 = new JSONObject();
        dynamic2.put("insertWay", 1);
//        dynamic2.put("keyword", "");
        dynamic2.put("pageBegin", "1");
        dynamic2.put("cellHeight", "16.0");

        dynamic2.put("colWidthPercent", new int[] { 3, 4, 4, 4});
        dynamic2.put("theFirstHeader", "附二");
        dynamic2.put("cellHorizontalAlignment", "1");
        dynamic2.put("cellVerticalAlignment", "5");
        dynamic2.put("headers", new String[] { "序号", "借款人", "贷款人", "金额" });
        String row1[] = new String[] { "1", "小网", "小易", "1000" };
        String row2[] = new String[] { "2", "小云", "小音", "2000" };
        String row3[] = new String[] { "3", "小乐", "天马", "3000" };
        dynamic2.put("datas", new String[][] { row1, row2, row3 });
        dynamic2.put("headersAlignment", "1");
        dynamic2.put("tableWidthPercentage", 80);
        dynamic_tables.add(dynamic2);
        System.out.println(dynamic_tables.toString());
        return dynamic_tables.toString();
    }
}
```

返回示例:

正例:

```
{"code": "1000", "download_url": "http://testapi.fadada.com.....BQg==", "msg": "操作成功", "result": "success", "viewpdf_url": "http://testapi.fadada.com.....BQg=="}
```

反例: {"code": "2002", "download\_url": null, "msg": "对不起, 没有检索到模板文件!", "result": "error", "viewpdf\_url": null}

## 5.10 自动签署

接口类型: HTTP POST 请求

接口说明: 自动签接口不需要用户亲自操作, 当接入平台调用此接口时, 就会在指定的电子合同上签下指定用户的电子章, 省略了用户交互的步骤。因此如果需要使用此接口的能力, 需要接入平台联系法大大商务签署补充协议。



注意事项：因为该接口签署对于用户是无感知的，会存在一定的法律风险，默认只允许接入平台本身调用，如果需要给接口平台的关联公司（接入平台的子公司，或者母公司）使用，需走[报备流程](#)

调用方向：接入平台->法大大

接口地址：https://FDDServer:Port/api/extsign\_auto.api

输入参数：

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如：900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(transaction_id+ti mestamp) +SHA1(app_secret+ customer_id) ) ) 标记红色部分规则：对所有业务参数进行 ascll 码排序后拼接	
业务参数				
参数	类型	是否必填	描述	说明
transaction_id	String	是	交易号	每次请求视为一个交易。 只允许长度<=32 的英文或数字字符。 交易号为接入平台生成，必须保证唯一并自行记录。
contract_id	String	是	合同编号	根据合同编号指定在哪份文档上进行签署。 合同编号在合同上传或模板填充时设定。
customer_id	String	是	客户编号	注册账号时获取。
client_role	String	是	客户角色	1-接入平台；
doc_title	String	是	文档标题	如“xx 投资合同”
position_type	String	否	定位类型	0-关键字（默认） 1-坐标
sign_keyword	String	否	定位关键字	关键字为文档中的文字内容（能被 ctrl+f 查找功能检索到）。 法大大按此关键字进行签章位置的定位，将电子章盖在这个关键字上



				面。凡出现关键字的地方均会盖上指定用户的电子章，建议关键字在合同中保持唯一。 当 <code>position_type</code> 为 0 时，此参数必填
<code>keyword_strategy</code>	String	否	关键字签章策略	关键字签章策略： 0: 所有关键字签章（默认）； 1: 第一个关键字签章； 2: 最后一个关键字签章；
<code>signature_positions</code>	String	否	定位坐标	类型: String 格式: <code>JSONArray&lt;SearchLocation&gt;</code> 当 <code>position_type</code> 为 1 时，此参数必填
<code>notify_url</code>	String	否	签署结果异步通知 URL	如果指定，当签章完成后，法大大将向此 URL 发送签署结果。 参见 <a href="#">签署结果异步通知规范 (notify_url)</a>

### SearchLocation 属性说明

属性名称	类型	描述	备注
<code>pagenum</code>	int	页码	签章页码，从 0 开始。即在第一页签章，传值 0。
<code>x</code>	double	盖章点 x 坐标	盖章点 x 坐标，以左上角为原点 0,0，取鼠标位置为图片中心点。坐标请以图片宽度为 800 为基准。
<code>y</code>	double	盖章点 y 坐标	盖章点 y 坐标，以左上角为原点 0,0，取鼠标位置为图片中心点。坐标请以图片高度为 1131 为基准。

输出参数 (JSON 格式数据):

参数	要求	描述	备注
<code>result</code>	必选	处理结果	success: 成功 error: 失败
<code>code</code>	必选	状态码	1000: 操作成功 2001: 参数缺失或者不合法 2002: 业务异常，失败原因见 msg 2003: 其他错误，请联系法大大
<code>msg</code>	必选	描述	

## SDK 使用说明

```
// JAVA
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
```



```

ExtsignReq req = new ExtsignReq();
req.setCustomer_id(""); //客户编号
req.setTransaction_id(""); //自定义交易号
req.setContract_id(""); //上传/模板填充接口合同编号
req.setClient_role(""); //客户角色
req.setSign_keyword(""); //定位关键字
req.setDoc_title(""); //文档标题
String result = base.invokeExtSignAuto(req);

```

返回示例:

正例:

```

{"code": "1000", "download_url": "http://testapi.fadada.com....", "msg": "文档签署成功", "result": "success", "viewpdf_url": "http://testapi.fadada.com...."}

```

反例:

```

{"code": "2002", "download_url": null, "msg": "对不起，签约关键字不能为空！", "result": "error", "viewpdf_url": null}

```

## 5.11 手动签署接口

接口类型: HTTP GET 请求

接口说明: 该接口为页面接口, 接入方平台可以在合适的业务场景嵌入该接口链接, 引导客户至法大大进行文档签署 (比如可以在接入方平台网站上放置一个按钮, 该按钮触发跳转至法大大 或将法大大签章页面嵌入接入方流程)。法大大根据浏览器 UA 信息, 自动加载签章界面对应的 PC web 版本或移动 HTML5 版本。用户在法大大的签署页面中进行签署操作。

调用方向: 接入平台->法大大

接口地址: <https://FDDServer:Port/api/extsign.api>

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(transaction_id+timestamp) +SHA1(app_secret+customer_id) ) )	
业务参数				



参数	类型	是否必填	描述	说明
transaction_id	String	是	交易号	每次请求视为一个交易。 只允许长度<=32的英文或数字字符。 交易号为接入平台生成, 必须保证唯一并自行记录。
contract_id	String	是	合同编号	根据合同编号指定在哪份文档上进行签署。 合同编号在文档传输或合同生成时设定。
customer_id	String	是	客户编号	CA注册时获取。
doc_title	String	是	文档标题	如“xx 投资合同”。 需 URL encoding 处理, 编码 utf-8。
sign_keyword	String	否	定位关键字	关键字为文档中的文字内容(要能使用ctrl+f 搜索到)。法大大按此关键字进行签章位置的定位, 将电子章盖在这个关键字上面。凡出现关键字的地方均会盖上指定用户的电子章, 建议关键字在合同中保持唯一。
keyword_strategy	String	否	关键字签章策略	关键字签章策略: 0: 所有关键字签章 (默认); 1: 第一个关键字签章; 2: 最后一个关键字签章;
return_url	String	是	页面跳转 URL (签署结果同步通知)	当签章完成后, 在签章页面停留数秒后浏览器会自动跳转到此 URL, 内容由接入平台自行展示; 参见 <a href="#">页面跳转规范 (return_url)</a>
notify_url	String	否	签署结果异步通知 URL	如果指定, 当签章完成后, 法大大将向此 URL 发送签署结果。 参见 <a href="#">签署结果异步通知规范 (notify_url)</a>

## SDK 使用说明

```
// JAVA
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
ExtsignReq req = new ExtsignReq();
req.setCustomer_id("");//客户编号
req.setTransaction_id();//交易号
req.setContract_id("");//合同编号
req.setDoc_title("");//文档标题
req.setReturn_url("");页面跳转 URL (签署结果同步通知)
String sign_url= base.invokeExtSign(req);
// sign_url 是组装好的地址, 请重定向到这个地址呈现签署页面给用户
```



```
// 例如: HttpServletResponse().sendRedirect(sign_url);
// 输出签署页面
```

输出参数 (JSON 格式数据):

此接口时页面接口, 返回的是签署链接, 打开页面即可进入签署页面。

## 5.12 合同查看

接口类型: HTTP GET 请求

接口说明: 页面接口, 返回签署页面, 根据浏览器 UA 信息返回 pc 或 H5 页面.

调用方向: 接入平台->法大大

接口地址: <https://FDDServer:Port/api/viewContract.api>

输入参数:

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret +contract_id) ) ) 标记红色部分规则: 对所有 业务参数进行 ascii 码排序后 拼接	
业务参数				
参数	类型	是否必填	描述	说明
contract_id	String	是	合同编号	

## SDK 使用说明

```
// JAVA
FddClientExtra extra = new FddClientExtra(APP_ID,APP_SECRET,V,HOST);
String contract_id = ""; //合同编号
String view_url= extra.invokeViewPdfURL(contract_id);
```



```
// 此时 view_url 为查看链接，请开发者自行跳转
// 例如：HttpServletResponse().sendRedirect(view_url);
// 输出页面
```

返回说明：

正例：该接口为页面接口，打开链接即可看到合同内容。  
反例：该接口为页面接口，打开链接会有对应的错误提示。

## 5.13 合同下载

接口类型：HTTP GET 请求

接口说明：通过合同编号下载文档（PDF 格式），可通过 Content-Length 取文件大小。

接口地址：<https://FDDServer:Port/api/downloadContract.api>

调用方向：接入平台->法大大

接口返回：正常时直接返回 PDF 文档（MIME:application/pdf），异常时返回 JSON 报文

输入参数：

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID 如：900001	
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号 2.0	
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret + contract_id) ) ) 标记红色部分规则：对所有 业务参数进行 ascll 码排序 后拼接	
业务参数				
参数	类型	是否必填	描述	备注
contract_id	String	是	合同编号	

输出参数（JSON 格式数据）：

参数	类型	是否必填	描述	备注
result	String	是	处理结果	success: 成功 error: 失败
code	String	是	状态码	1000: 操作成功



				2001: 参数缺失或者不合法 4001: 无效交易号 4002: 文档已删除 (已下载) 4003: 其他原因
msg	String	否	描述	

## SDK 使用说明

```
// JAVA
FddClientExtra extra = new FddClientExtra(APP_ID,APP_SECRET,V,HOST);
String contract_id = ""; //合同编号
String download_url= extra.invokeDownloadPdf(contract_id);
// 此时 download_url 为组装好的下载链接，请开发者自行跳转
// 例如：HttpServletResponse().sendRedirect(download_url);
```

返回说明：

正例：该接口为页面接口，打开链接即可下载合同。  
反例：该接口为页面接口，打开链接会有对应的错误提示。

## 5.14 合同归档

接口类型：HTTP POST 请求

调用场景：接入平台更新合同签署状态为-签署完成，法大大将把合同所有相关操作记录进行归档存证。归档后将不能再对文档再进行签署操作。

接口地址：<https://FDDServer:Port/api/contractFiling.api>

调用方向：接入平台->法大大

输入参数：

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如：900001
timestamp	String	是	请求时间 yyyyMMddHHmmss	
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secre t+ contract_id) ) ) 标记红色部分规则： <b>对所有</b>	



			业务参数进行 ascll 码排序 后拼接	
<b>业务参数</b>				
参数	类型	是否必填	描述	说明
contract_id	String	是	合同编号	

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
result	String	是	处理结果	success: 成功 error: 失败
code	String	是	状态码	1000: 操作成功 2001: 参数缺失或者不合法 2002: 业务异常, 失败原因见 msg 2003: 其他错误, 请联系法大大
msg	String	否	描述	

## SDK 使用说明

```
// JAVA
FddClientBase base = new FddClientBase(APP_ID,APP_SECRET,V,HOST);
String contract_id = ""; //合同编号
String result = base.invokeContractFilling(contract_id);
```

返回示例:

```
正例: {"code":"1000","msg":"操作成功","result":"success"}
反例: {"code":"2002","msg":"合同编号不能为空.","result":"error"}
```

### 5.15.1 实名认证异步回调

接口类型: HTTP POST 请求

调用场景: 实名认证之后异步告知接入方认证结果

调用方向: 法大大 -> 接入平台

### 响应信息:

参数	类型	是否必填	描述	备注
appld	String	是	平台方 id	



serialNo	String	是	认证序列号	
customerId	String	是	客户编号	
status	String	是	个人: Status: 0: 未激活; 1: 未认证; 2: 审核通过; 3: 已提交待审核; 4: 审核不通过 企业: status: 0: 未认证; 1: 管理员资料已提交; 2: 企业基本资料(没有申请表)已提交; 3: 已提交待审核; 4: 审核通过; 5: 审核不通过; 6 人工初审通过,	个人: 2, 3, 4 企业: 3, 4, 5 以上状态发生变化就会连续异步通知 3 次, 如果 3 次之后通知还是失败, 建议主动查询个人实名认证信息 (5.17/5.18 接口)
statusDesc	String	否	不通过原因描述	
certStatus	String	否	0: 没有申请证书或者申请证书失败, 1: 成功申请证书	

## 回调示例:

```
{
  "appId": "900001",
  "serialNo": "5ae85dc2ddd948b5a8218de8a8e21cf8",
  "customerId": "3E0A88FA5F8261DD",
  "status": "2",
  "statusDesc": "测试"
}
```

## 5.15.2 实名认证同步回调

接口类型: HTTP GET 请求

调用场景: 实名认证之后同步告知接入方认证结果

调用方向: 法大大->接入平台



## 响应信息：

参数	类型	是否必填	描述	备注
personName (companyName)	String	是	用户姓名 (企业姓名)	
transactionNo	String	是	认证序列号	
authenticationType	String	是	帐号类型	1: 个人 2: 企业
status	String	是	个人: Status: 0: 未激活; 1: 未认证; 2: 审核通过; 3: 已提交待审核; 4: 审核不通过 企业: status: 0: 未认证; 1: 管理员资料已提交; 2: 企业基本资料(没有申请表)已提交; 3: 已提交待审核; 4: 审核通过; 5: 审核不通过; 6 人工初审通过,	个人: 2, 3, 4 企业: 3, 4, 5 以上状态发生变化就会连续异步通知 3 次, 如果 3 次之后通知还是失败, 建议主动查询个人实名认证信息 (5.17/5.18 接口)

## 回调示例：

```
https://www.xxx.com?personName=xxx&transactionNo=xxx&authenticationType=1&status=2
```

## 5.16 查询个人实名认证信息

接口类型：HTTP POST 请求

接口说明：接入方主动发起实名认证信息查询。

调用方向：接入平台->法大大

接口地址：[https://FDDServer:Port/api/find\\_personCertInfo.api](https://FDDServer:Port/api/find_personCertInfo.api)



输入参数

公共请求参数				
参数	类型	是否必填	描述	示例值
app_id	String	是	接入方 ID	900001
timestamp	String	是	请求时间	yyyyMMddHHmmss
v	String	是	版本号	2.0
msg_digest	String	是	Base64 (SHA1 (app_id +MD5(timestamp) +SHA1(app_secret+verified_serialno) ) ) 标记红色部分规则: 对所有业务参数进行 ascll 码排序后拼接	
请求参数				
参数	类型	是否必填	描述	示例值
verified_serialno	String	是	交易号, 获取认证地址时返回	范围: 32

输出参数 (JSON 格式数据):

参数	类型	是否必填	描述	备注
code	int	是	状态码	
Msg	String	是	状态描述	
data	String	否	JSON 对象	

data:

参数	类型	是否必填	描述	备注
transactionNo	String	是	交易号	该参数等同实名认证异步回调 serialNO
type	String	是	1.个人, 2 企业	
passTime	String	否	审核通过时间	yyyy-MM-dd HH:mm:ss
authenticationSubmitTime	String	是	认证提交时间	yyyy-MM-dd HH:mm:ss
person	Person		个人信息	

person

参数	类型	是否必填	描述	备注
personName	String	是	个人姓名	



type	String	是	0: 个人; 1: 法人; 2: 代理人,	
certType	String	是	证件类型, 0: 身份证; 1: 护照; B: 香港居民来往内地通行证; C: 台湾居民来往大陆通行证	
auditorTime	String	否	审核时间	yyyy-MM-dd HH:mm:ss
idCard	String	是	身份证号	
mobile	String	是	手机号	
status	String	是	0:未激活; 1:未认证; 2:审核通过; 3:已提交待审核; 4:审核不通过	
verifyType	String	是	person 认证方式, 0:腾讯云认证; 1:三要素认证; 2:手势照认证;	
auditFailReason	String	否	不通过原因	
headPhotoPath	String	是	身份证正面图片 <b>uuid</b>	当 certType=0 时是身份证正面图片 <b>uuid</b> 、当 certType=1 时是护照带人像图片 <b>uuid</b> 、当 certType=B 时是香港居民来往内地通行证带人像图片 <b>uuid</b> 、当 certType=C 时是台湾居民来往大陆通行证带人像图片 <b>uuid</b>
backgroundIdCardPath	String	是	身份证反面图片 <b>uuid</b>	当 certType=0 时是身份证反面图片 <b>uuid</b> 、当 certType=1 时是护照封面 <b>uuid</b> 、当 certType=B 时是香港居民来往内地通行证封面图片 <b>uuid</b> 、当 certType=C 时是台湾居民来往大陆通行证封面图片 <b>uuid</b>



photoUuid	String	否	微众返回的照片 uuid	只有通过微众人脸识别成功才返回
gesturesPhotoPath	String	否	手势照图片 uuid	手势照图片 uuid
fork	String	否	民族	
birthday	String	否	出生日期	yyyy-MM-dd
sex	String	否	性别	1 男, 2 女
startDate	String	否	证件起始时间	yyyy-MM-dd
expiresDate	String	否	证件到期时间	yyyy-MM-dd
issueAuthority	String	否	颁发机构	
address	String	否	地址	

## SDK 使用说明

```
// JAVA
FindCertInfo personCertInfo = new FindCertInfo(APP_ID,APP_SECRET,V,HOST);
String verified_serialno = ""; //交易号, 获取认证地址时返回
String result = personCertInfo.invokeFindPersonCert(verified_serialno,"1");
注: sdk 封装如果第 2 参数传 1 则是查询个人, 其他比如"2"就是查询企业。
```

返回示例:

正例:

```
{ "code": 1, "data": { "authenticationSubmitTime": "1546939583000", "bankCard": null, "company": null, "manager": null, "passTime": "1546939597000", "person": { "auditFailReason": "", "auditorTime": "", "headPhotoPath": "xxx", "idCard": "xxx", "mobile": "xxx", "personName": "xxx", "status": "2", "type": "0", "verifyType": "1", "transactionNo": "xxx", "type": "1", "msg": "success" } }
```

反例: { "code": 3100, "data": null, "msg": "账号类型非法" }

## 5.17 查询企业实名认证信息

接口类型: HTTP POST 请求

接口说明: 接入方主动发起实名认证信息查询。

调用方向: 接入平台-法大大

接口地址: [https://FDDServer:Port/api/find\\_companyCertInfo.api](https://FDDServer:Port/api/find_companyCertInfo.api)

输入参数

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	如: 900001
timestamp	String	是	请求时间	yyyyMMddHHmmss



v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret + verified_serialNo) ) ) 标记红色部分规则：对所有业务参数进行 ascll 码排序后拼接	
<b>请求参数</b>				
<b>参数</b>	<b>类型</b>	<b>是否必填</b>	<b>描述</b>	<b>备注</b>
verified_serial_no	String	是	交易号，获取认证地址时返回	范围：

输出参数 (JSON 格式数据)：

参数	类型	是否必填	描述	备注
code	int	是	状态码	
Msg	String	是	状态描述	
data	String	否	JSON 对象	

data

参数	类型	是否必填	描述	备注
transactionNo	String	是	交易号	
type	String	是	1.个人, 2.企业	
passTime	String	否		yyyy-MM-dd HH:mm:ss
authenticationSubmitTime	String	是		yyyy-MM-dd HH:mm:ss
company	Company	否		Type= 2 时
manager	manager	否		Type =2 时
bankCard	bankCard	否		Type =2 时



Company

参数	类型	是否必填	描述	备注
relatedTransactionNo	String	是	关联法人/代理人交易号	
verifyType	String	是	认证方式, 0, 银行卡认证; 1, 纸质审核认证	
companyName	String	是	企业名称	
organization	String	是	统一社会信用代码(多合一证件)	
organizationPath	String	是	统一社会信用代码(多合一证件)图片 uuid	
legalName	String	否	法人姓名	
legal	String	否	法人身份证号	
regFormPath	String	是	企业信息登记表图片 uuid	
status	String	是	0: 未认证; 1: 管理员资料已提交; 2: 企业基本资料(没有申请表)已提交; 3: 已提交待审核; 4: 审核通过; 5: 审核不通过; 6 人工初审通过,	
auditFailReason	String	否	不通过原因	
auditorTime	String	否	审核时间	yyyy-MM-dd HH:mm:ss
hasagent	String	是	是否有代理人,	1: 代理人, 0: 法人

BankCard

参数	类型	是否必填	描述	备注
BankName	String	否	银行名称	
bankDetailName	String	否	支行名称	
bankCardNo	String	否	银行卡号	

备注: Manager 跟 person 字段一样



## SDK 使用说明

```
// JAVA
FindCertInfo personCertInfo = new FindCertInfo(APP_ID,APP_SECRET,V,HOST);
String verified_serialno = "";//交易号，获取认证地址时返回
String result = personCertInfo.invokeFindPersonCert(verified_serialno,
    "2");
注： sdk 封装如果第 2 参数传 1 则是查询个人，其他比如"2"就是查询企业。
```

返回示例：

正例：

```
{ "code":1,"data":{"authenticationSubmitTime":"1546938963000","bankCard":null,"company":{"auditFailReason":
":"","auditorTime":"2019-01-08
17:15:05.0","companyName":"xxx","hasagent":"0","legal":"","legalName":"xxx","organization":"xxx","organizati
onPath":"xxx","regFormPath":"xxx","relatedTransactionNo":"xxx","status":"4","verifyType":"1"},"manager":{"au
ditFailReason":"","auditorTime":"","headPhotoPath":"xxx","idCard":"xxx","mobile":"xxx","personName":"xxx","
status":"2","type":"1","verifyType":"0"},"passTime":"1546938905000","person":null,"transactionNo":"xxx","type
":"2"},"msg":"success"}
```

反例： { "code":3100,"data":null,"msg":"账号类型非法"}

## 5.18 通过 uuid 下载文件

接口类型： HTTP GET/POST 请求

接口说明： 提供查询个人认证信息功能，请求参数一律用字符串表示。

调用方向： 接入平台 -> 法大大

接口地址： [https://FDDServer:Port/api/get\\_file.api](https://FDDServer:Port/api/get_file.api)

输入参数：

公共请求参数				
参数	类型	是否必填	描述	备注
app_id	String	是	接入方 ID	900001
timestamp	String	是	请求时间	yyyyMMddHHmmss
v	String	是	版本号	2.0
msg_digest	String	是	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret+ uuid) ) )	



			标记红色部分规则： <b>对所有业务参数进行 ascll 码排序后拼接</b>	
<b>请求参数</b>				
参数	类型	是否必填	描述	备注
uuid	String	是	图片 uuid 查询认证结果时返回	范围：32

## SDK 使用说明

```
// JAVA
GetFileForUUID getFileForUUID = new GetFileForUUID(APP_ID,APP_SECRET,V,HOST);
String uuid = "";//图片 uuid, 查询认证结果时返回
String result = getFileForUUID.invokeFileForUUID(uuid);
//result 下载链接, 可保存对应的图片信息。
```

返回说明:

该接口返回一个下载链接, 可保存对应的图片信息。

## 5.19 页面跳转规范 (return\_url)

请求方式: HTTP GET

调用方向: 法大大->接入平台

规范说明: 该 URL 地址由接入方在调用签署接口时指定; 签章结束后法大大将把客户引导至此页面, 该页面由接入方自行决定如何展示。如可以提供客户查看已签署文档、下载已签署文档的按钮。

输入参数:

参数	要求	描述	备注
transaction_id	必选	交易号	
result_code	必选	签章结果代码	3000 (签章成功) 3001(签章失败)
result_desc	必选	签章结果描述	签章结果描述信息
download_url	可选	下载地址	下载签署后的合同
viewpdf_url	可选	查看地址	查看签署后的合同
timestamp	必选	请求时间	yyyyMMddHHmmss
msg_digest	必选	摘要	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret+transaction_id)



			) ) 建议接入平台方进行校验
--	--	--	-----------------------

## 5.20 签署结果异步通知规范 (notify\_url)

请求方式: HTTP POST

调用方向: 法大大->接入平台

规范说明: 该 URL 地址由接入方在调用签署接口时指定;

输入参数:

参数	要求	描述	备注
transaction_id	必选	交易号	
contract_id	必选	合同编号	
result_code	必选	签章结果代码	3000 (签章成功) 3001(签章失败)
result_desc	必选	签章结果描述	签章结果描述信息
download_url	可选	下载地址	下载签署后的合同
viewpdf_url	可选	查看地址	查看签署后的合同
timestamp	必选	请求时间	yyyyMMddHHmmss
msg_digest	必选	摘要	Base64( SHA1( app_id +MD5(timestamp) +SHA1(app_secret+transac tion_id) ) ) 建议接入平台方进行校验

输出参数

HTTP\_STATUS 为 200 即视为通知成功

## 5.21 接口错误码列表:

公共错误码		
code	msg	描述
-1	抱歉, 暂时无法处理您的请求, 请稍后再试。	未知异常信息
0	error	失败 (有时候会带上具体原因)
1	success	成功



2	repeat request	重复请求
1001	公共参数非法	公共参数指 app_id, timestamp, v, msg_digest 为空或者不符合规则
1002	app_id 不存在或未启动	
1003	msg_digest 无效	
1004	请求参数非法	必选参数为空（有时候会带上具体原因）
1005	未绑定 IP 白名单	
1006	请求频繁，请稍后再试	
<b>接口错误码</b>		
1201	签署记录为空	
1202	签署记录为待签署	
1203	签署记录未成功	
1204	签署记录为失败	
1205	签署记录为撤销	
1301	实名认证信息已锁定	
1302	尚未实名认证	
1303	实名认证信息正在审核中	
1304	实名认证信息未通过	
1305	实名认证信息通过	
1401	合同记录为空	
1402	合同已归档	
1403	合同已过期	
1404	合同已撤销	
1405	合同已拒签	
1406	合同状态不允许操作	
1409	已签署	
1501	手机信息为空	
1502	短信认证失败	
1504	人脸认证失败	
1505	地址已失效	
1506	骑缝章客户编号异常	
1507	签章图片获取失败	
1509	交易号重复	
1511	签署成功，请勿重复请求	
1601	客户编号无效	
1602	原章不存在	
1603	下载到本地失败	
1604	图片为空	
1605	图片过大	
1607	图片格式无效	
1608	印章编号无效	



1609	印章数量为空	
1610	企业用户不能手写签章	
1611	印章内容过长	
1801	合同编号非法	
1802	合同编号无效	
1803	合同已删除	
1804	合同名称不合法	
1903	客户编号非法	
1904	客户编号解析失败	
2501	模板编号非法	
2502	无效模板编号	
2503	模板已删除	
2504	无效模板文件	
2505	模板状态不允许修改	
2506	获取模板图片异常，请稍后再试	
2507	对不起，模板号重复	
2508	模板文件上传失败	
2509	模板不存在	
2510	合同 id 重复	
3201	不能重复申请证书	
3202	存证编号非法	
3203	证书申请失败	
3204	实名认证序列号非法	
3205	实名认证未通过	
3206	不允许申请实名证书	

## 报备流程：

平台方关联公司如果需要走自动签接口，需要关联公司提供《[关联公司开通自动签免责声明.docx](#)》，然后后台添加白名单。流程是：关联公司申请 CA，承诺书盖章后扫描件邮件发给 [api@fadada.com](mailto:api@fadada.com), 运营后台配置。纸质件须寄送到法大大总部。

