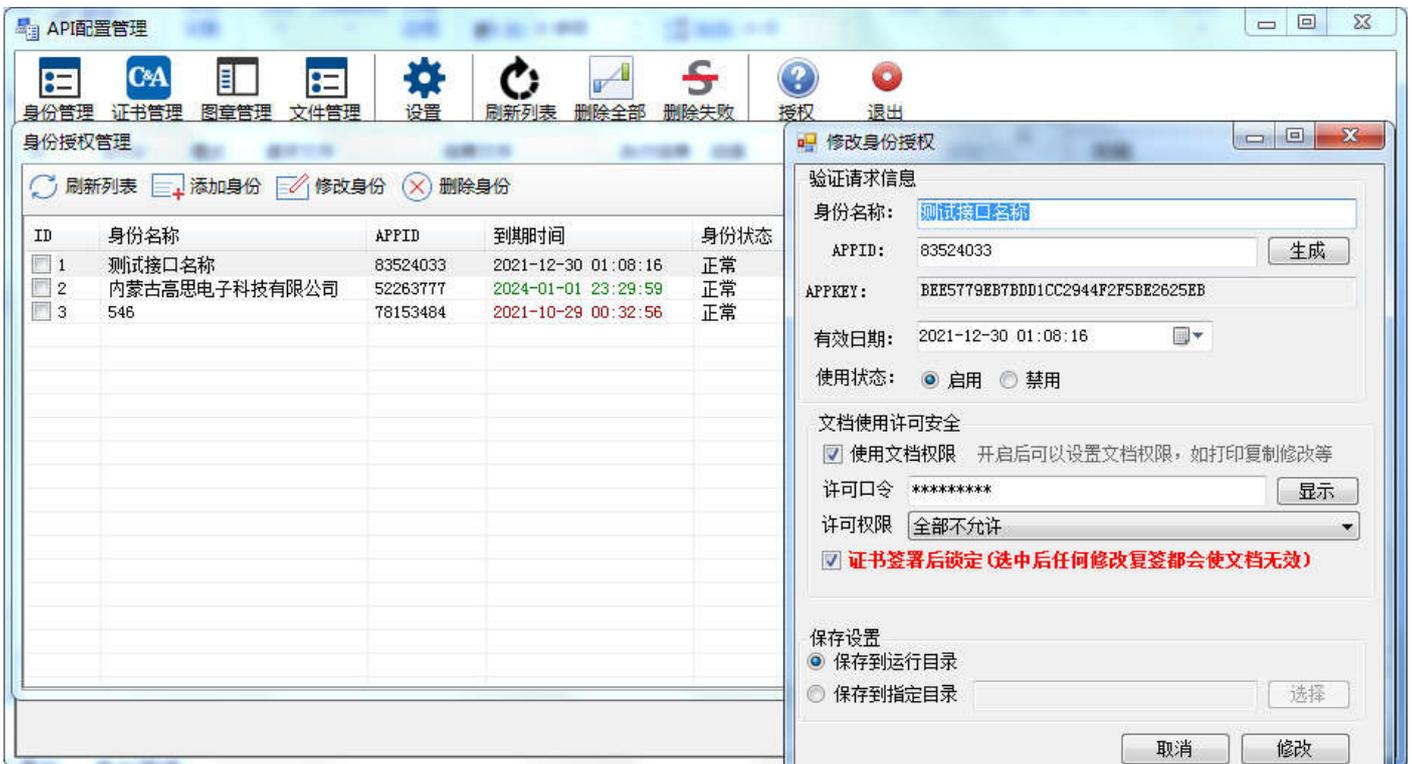


API 配置端使用说明



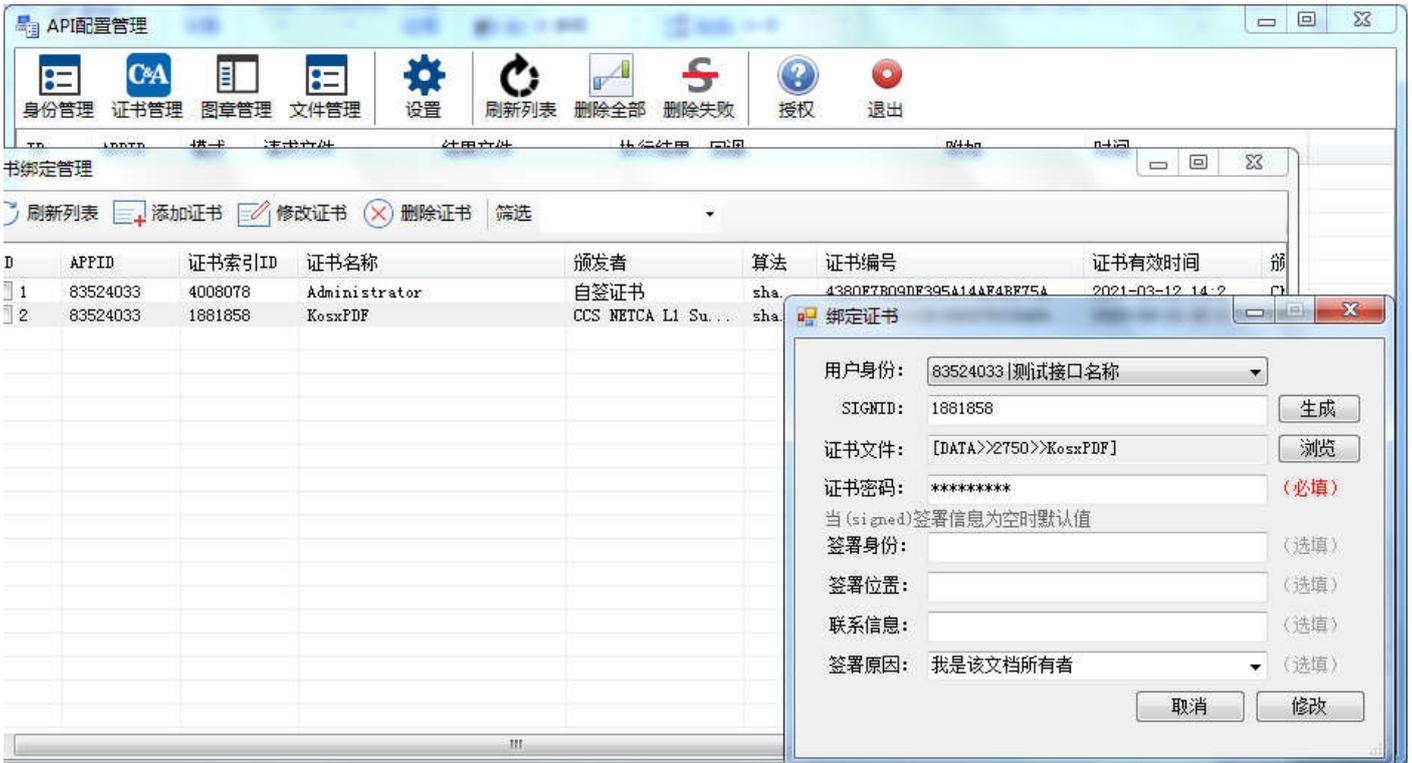
工具栏 - 身份管理

同账号密码注册一样 APPID 用于身份认证 APPKEY 用于签名认证 有效时间 文档权限等管理
通过身份设置 不同身份对文档签署拥有不同的权限 可以设置签章后的文档 打印 修改 等



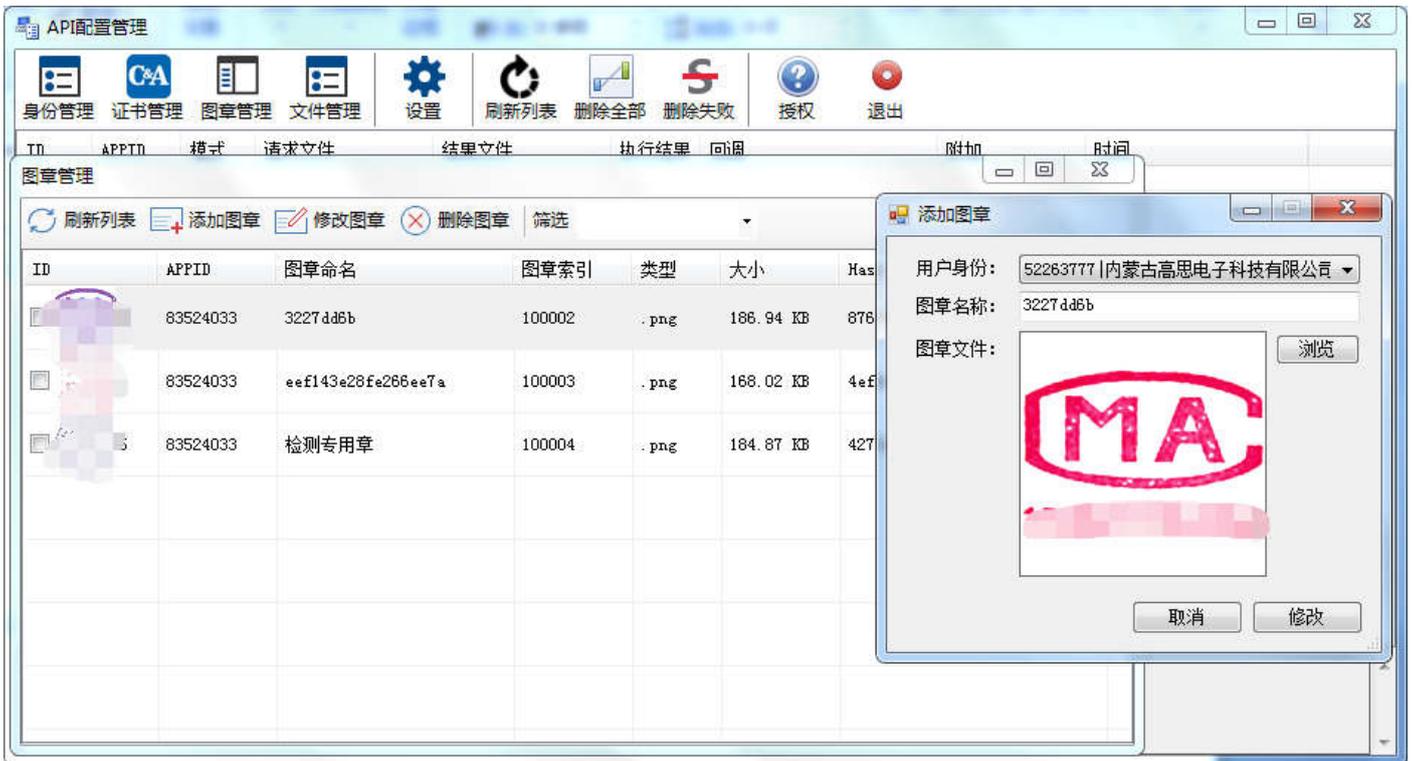
工具栏 - 证书管理

不同的身份对应自己的证书绑定 每一个证书都有单独的索引 ID 签章请求必须的参数
通过证书设置 进行证书上传后 只能删除修改 不能导出 请做好证书备份工作



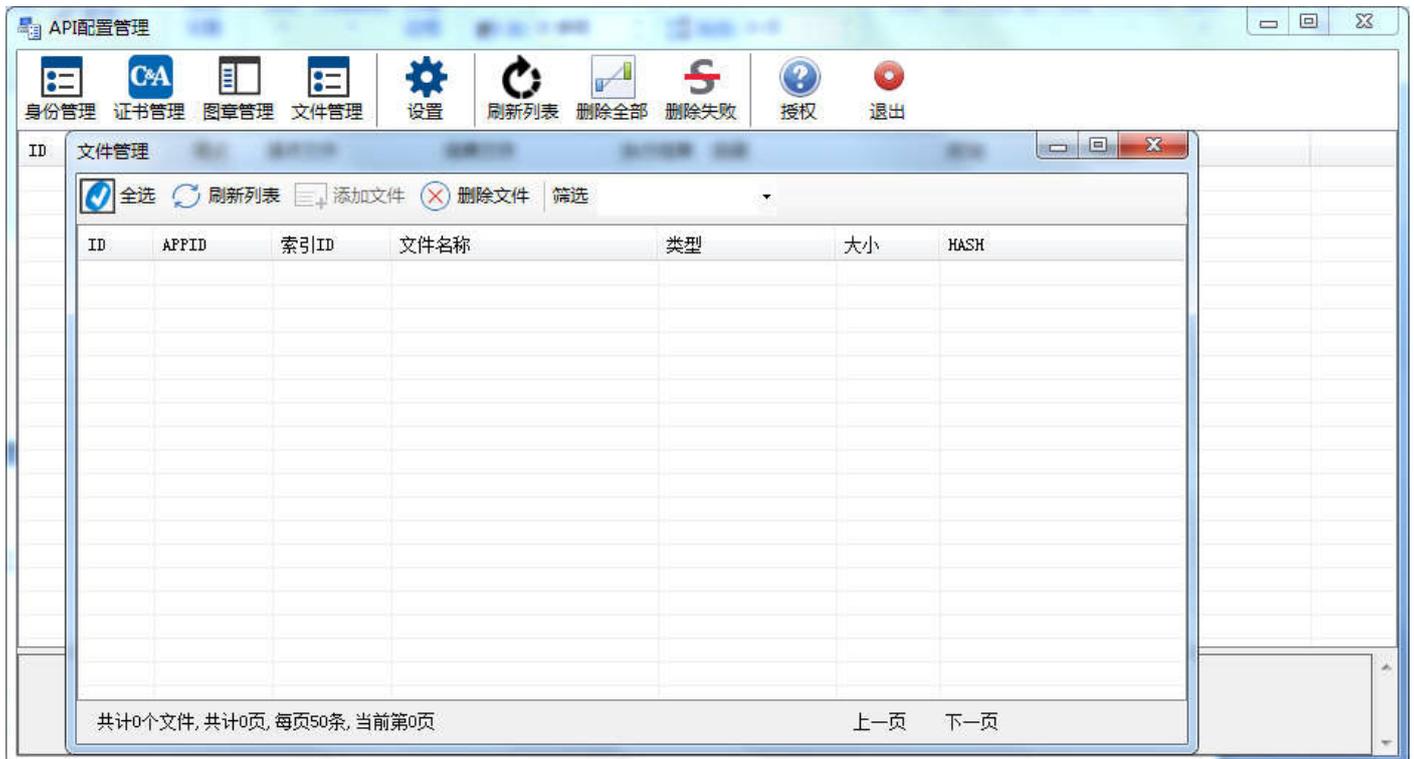
工具栏 - 图章管理

不同身份绑定自己的图章图片或签名图片 每一个图片都有自己的 索引 ID API 签章的必须参数
图章管理支持通过 API 进行管理操作



工具栏 - 文件管理

不同身份绑定自己的上传文件 每一个文件都有自己的 索引 ID API 签章时 fileid 模式的必须参数
文件管理只能通过 API 进行上传管理操作
此功能非必要功能 可以不使用



工具栏 - 设置

设定日志开关 与 其他辅助设置 功能为调试功能 非主要功能



参数请求签名

关于请求说明 所有请求皆为 POST 请求

返回数据分 XML 和 JSON 两种格式 系统级错误返回 XML 程序级返回 JSON

JQ 请尽量使用 \$.post 如果使用 \$.ajax 请设置 ContentType(注意这个要大写,小写无效) ajax 部分设定功能或影响请求

原生请求 注意请求数据形式 必须是 Form Data

接口为了防止爬虫 移除了 Request header 和 Request payload 形式的请求

签名

每一次请求都追加 timestr、 sign 参数

timestr 是时间戳函数 例:1640333261

sign 参数按规则 md5 签名 md5(appid + appkey + md5(datastr))

appid 和 appkey 都是由服务器给的 datastr 是请求参数 按字母排序后进行拼接 datastr = 键 + '=' + encodeURI(值) + '&';

```
//objKeySort() 参数排序
var newkey = Object.keys(param).sort();
//先用 Object 内置类的 keys 方法获取要排序对象的属性名,
//再利用 Array 原型上的 sort 方法对获取的属性名进行排序,
//newkey 是一个数组
var newObj = {};//创建一个新的对象,用于存放排好序的键值对
for (var i = 0; i < newkey.length; i++) { //遍历 newkey 数组

//向新创建的对象中按照排好的顺序依次增加键值对
//注意键值转为小写
newObj[newkey[i].toLocaleLowerCase()] = obj[newkey[i]];
}
return newObj;//返回排好序的新对象
```

```
//signRequest(param) 参数签名 签名时不包含 appid 和 appkey
param = objKeySort(param); //请求参数排序
var datastr = "";
for (var k in param) {

//isJSON(param[k]) json 数据主要转为字符串
datastr += k + '=' + encodeURIComponent(param[k]) + '&';
}
if (datastr) {

//去掉参数拼接最后一个&符号
datastr = datastr.substring(datastr, datastr.length - 1);
}
var hash = md5(appid + appkey + md5(datastr));
```

图章签名

上传图章签名 POST: <http://localhost:8089/api/Stamp/Upload>

图章签名文件不再由客户端请求文件签章，由客户端单独进行上传 获得图章 ID 签章 减少每次签章重复请求
图章文件

请求的参数

file：请求的文件 formdata 支持多个文件同时选择上传（附件不参与签名）

name：图章或者签名的备注名称 可以为空 为空默认文件原名

appid

timestr：时间戳

sign：签名（注意的是 这里的签名就是 name 不管 name 是否赋值 都要签名）例：sign = md5(appid + appkey +

md5("name="))

返回结果 {code:1,message:'消息',result:[]} code 为 0 上传失败

图章签名列表 POST: <http://localhost:8089/api/Stamp/Select>

查询 APPID 下的图章 ID name 可以为空 可以指定图章备注名称

请求的参数

name : 图章或者签名的备注名称 可以为空 为空返回所有文件

appid

timestr : 时间戳

sign : 签名 (注意的是 这里的签名就是 name 不管 name 是否赋值 都要签名) 例:sign = md5(appid + appkey + md5("name="))

返回结果 {code:1,message:'消息',result:[]} code 为 0 查询失败

图章签名 删除 POST: <http://localhost:8089/api/Stamp/Delete>

删除 APPID 下的图章文件

请求的参数

fileid : 图章或者签名的索引 ID 必填 (多个文件删除 需要用英文逗号分隔) 例如:10001,10002

appid

timestr : 时间戳

sign : 签名 例:sign = md5(appid + appkey + md5("fileid=10001,10002"))

返回结果 {code:1,message:'消息',result:[]} code 为 0 查询失败

PDF 文件管理

签章文件分为多个模式 文件管理类似 object 文件 先上传文件 后请求签章 可以提升签章速度 (非必须使用的功能, 提供了一种小程序开发的选择)

上传文件 POST: <http://localhost:8089/api/Files/Upload>

请求的参数

file : 请求的文件 formdata 支持多个文件同时选择上传 (附件不参与签名)

appid

timestr : 时间戳

sign : 签名 (注意的是 这里的签名是空签) 例:sign = signRequest({})

返回结果 {code:1,message:'消息',result:[]} code 为 0 上传失败

文件列表 POST: <http://localhost:8089/api/Files/Select>

查询 APPID 下的文件 ID name 可以为空 可以指定文件备注名称

请求的参数

name : 文件的备注名称 可以为空 为空返回所有文件

appid

timestr : 时间戳

sign : 签名 例:sign = signRequest({name:''})

返回结果 {code:1,message:'消息',result:[]} code 为 0 查询失败

文件 删除 POST: <http://localhost:8089/api/Files/Delete>

删除 APPID 下的文件

请求的参数

fileid : 文件索引 ID 必填 (多个文件删除 需要用英文逗号分隔) 例如:10001,10002

appid

timestr : 时间戳

sign : 签名 例:sign = signRequest({fileid:'10001,10002'})

返回结果 {code:1,message:'消息',result:[]} code 为 0 查询失败

签章请求

支持多个文件多条配置同时签章 (不推荐同时请求多个文件 数字骑缝章会占用大量请求时间 页数越多时间越长)

签章文件 POST: http://localhost:8089/api/Signature/Stamp

请求的参数

mode : 文件请求方式 本地端文件 local 网络文件 url 已上传文件 fileid

file : 请求文件地址或者参数

mode=local 时 lims OA 服务平台 和 签章 api 同时在一个服务器上建议使用

```
file = [{"file": "D:\Users\PDF 文档 1. pdf", "pass": ""}, {"file": "D:\Users\PDF 文档 2. pdf", "pass": ""}]
```

或 file = D:\Users\PDF 文档 2. pdf, D:\Users\PDF 文档 2. pdf

mode=url 网络资源请求方式

```
file = [{"file": "http://localhost/pdf/PDF 文档 1. pdf", "pass": "123"}, {"file": "http://localhost/pdf/PDF 文档 2. pdf", "pass": "123"}]
```

或 file =http://localhost/pdf/PDF 文档 1. pdf

或 file =http://localhost/pdf/PDF 文档 1. pdf, http://localhost/pdf/PDF 文档 2. pdf 多个资源用英文逗号分隔

mode=fileid 请求已上传资源的索引 ID

```
file = [{"file": "200074", "pass": "123"}, {"file": "200075", "pass": "123"}]
```

或 file =200074, 200075, 200076

attr : 附加参数 文本格式 API 不做处理 异步时原样返回

notify : 异步请求地址 为空则 API 做同步处理 (大文件或者超过 50 个以上的数字签名 建议做异步处理)

stamp : 签章配置内容参数 文本格式 JSON 做转字符串处理

stamp 详细参数如下

```
[{
```

```
  "type": "contract", //contract 普通签名图章 stampmark 骑缝章 signtrue 数字签章
```

```
  "page": "1-2, 3to5, 7, 9to-1", //页面范围 全部页面 all 或者为空 单个指定页面 1, 2, -1, -2(文档倒数页面) 连续页面 1to10(可以输入倒数页面)或 1-10 (to 和 - 的区别是 to 支持负页面 -不支持负页面)
```

```
  "keyword": "", //关键词获取坐标, 合同章、数字签章可用-当此参数不为空 page, page_post 无效. x:y 是获取位置的偏移值, 为 0 居中-如获取关键词坐标失败则不执行配置
```

```
  "page_post": "0", //起始位置-默认为 0 或为空 签章或数字签章 0 左下角 1 左上角 2 右上角 3 右下角 骑缝章则 0 右侧 1 左侧
```

```
  "page_odd": "1-2" //骑缝章专有参数 该设置只在 page 页面范围内 默认为 0 或为空 1 是单页加骑缝章 2 是双页加骑缝章
```

```
  "page_x": "0", //横向边距位置 单位:mm 如果 x:y 都为 0 则居中 属于 page_post 位置(1:1)起始进行的偏移属性 与文档尺寸 1:1 关联 例如文档是 A4 尺寸 210mm×297mm 则 x y 不能超过这个范围
```

```
  "page_y": "0", //纵向边距位置 单位:mm 如果 x:y 都为 0 则居中
```

```
  "page_w": "42", //图章显示宽度 单位:mm 实际应用文档中 宽度是参与 dpi 计算的 高度是用于参考定位, 本身不计算 DPI. 图章不能带空白边框 (常用标准图章尺寸 38mm 40mm 42mm)
```

```
  "page_h": "42", //图章显示高度 单位:mm
```

```
  "stamp_id": "100002", //图章索引 ID 必填项 错误的索引或空则不执行该配置
```

```
  "stamp_bottom": "true", //图章是否在文字下面 默认为空或 false 数字签章此参数无效
```

```
"stamp_angle": "1-10", //图章旋转角度 范围 0-360 度 默认空或者 0 范围内随机旋转输入范围值 例如 10-50
```

```
"stamp_opacity": "180-255", //图章透明度 默认 255 即不做设置 范围值则随机范围内的透明值 例如 200-255
```

```
"stamp_offset": "1-2", //位置偏移 默认值 0 范围值随机偏移, 单位 px (此值只应用于文档中内部尺寸 不支持小数范围) 值不宜设置过大 建议 100 以内的整数 固定偏移输入固定数字 列入 5 或 15 如为单个数字 则横向和纵向都偏移此值, //也可以横向或者纵向偏移 例如 5-10, 8-10 标识 横向随机偏移 5-10 之间的像素 纵向随机偏移 8-10 之间的像素 //或者 输入 0, 1-10, 表示横向坐标不变 纵向偏移 1-10 的范围, 或者是 0, 20 横向不变 纵向偏移 20px
```

```
"sign_id": "1881858", //证书索引 ID 通过 API 服务端添加的证书会产生一个 ID (安全原因 不能通过 API 进行证书增删操作) 同一个 APPID 身份可以有多个证书 ID 通过设置身份进行文档权限设置
```

```
//此参数专属 stampmark 骑缝章 signtrue 数字签章 如果 (错误/或者为空) 使用索引 ID 则不执行该配置 骑缝章不建议使用数字签章(实测页面太多的时候或者图片式的 PDF 都会占用大量 IO CPU GDI 对硬件配置要求要最低 2 核 4G 及以上 )
```

```
//此下是多个配置的演示 同上参数是一样的
```

```
{  
  "type": "contract",  
  "page": "1-2, 3to5, 7, 9to-1",  
  "keyword": "",  
  "page_post": "0",  
  "page_x": "0",  
  "page_y": "0",  
  "page_w": "42",  
  "page_h": "42",  
  "stamp_id": "100002",  
  "stamp_bottom": "true",  
  "stamp_angle": "1-10",  
  "stamp_opacity": "180-255",  
  "stamp_offset": "1-2"  
},  
{  
  "type": "contract",  
  "page": "all",  
  "keyword": "白皮书",  
  "page_post": "0",  
  "page_x": "0",  
  "page_y": "0",  
  "page_w": "42",  
  "page_h": "42",  
  "stamp_id": "100003",  
  "stamp_bottom": "true",  
  "stamp_angle": "1-10",  
  "stamp_opacity": "150-250",  
  "stamp_offset": "1-5"  
},  
{  
  "type": "signtrue",  
  "page": "all",  
  "keyword": "数字签名",  
  "page_post": "0",
```

```
    "page_x": "0",
    "page_y": "0",
    "page_w": "42",
    "page_h": "42",
    "stamp_id": "100004",
    "stamp_angle": "5",
    "stamp_opacity": "80",
    "stamp_offset": "1-10, 5-8",
    "sign_id": "1881858"
  },
  {
    "type": "stampmark",
    "page": "all",
    "page_post": "1",
    "page_odd": "",
    "page_x": "0",
    "page_y": "0",
    "page_w": "42",
    "page_h": "42",
    "stamp_id": "100006",
    "stamp_angle": "9.5",
    "stamp_opacity": "120-255",
    "stamp_offset": "1-2, 2-5",
    "stamp_bottom": "true"
  },
  {
    "type": "stampmark",
    "page": "all",
    "page_post": "0",
    "page_odd": "2",
    "page_x": "0",
    "page_y": "-2",
    "page_w": "42",
    "page_h": "42",
    "stamp_id": "100007",
    "stamp_angle": "9.5",
    "stamp_opacity": "170-195",
    "stamp_offset": "1-6, 1-5",
    "sign_id": "3133888"
  }
]
```

appid

timestr: 时间戳

sign: 签名 例:sign = signRequest({mode:mode,file:file,attr:attr,notify:notify,stamp:stamp})

返回结果 {code:1,message:'消息',result:[]} code 为 0 请求失败