ni

汉王智能输入云服务

（B2C服务）

API-V1.0

汉王科技-互联网

2016/2/19

变更记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 修改点说明 | 变更人 | 变更日期 | 审批人 | 审批日期 |
| V 1.0 | 创建 | 赵苏 | 2016.2.19 |  |  |

目录

[前言 2](#_Toc447271455)

[1. 客户端与服务器端的接口定义约定 3](#_Toc447271456)

[1.1. 基本规则 3](#_Toc447271457)

[1.2. 请求安全认证 3](#_Toc447271458)

[1.3. 结果码定义 3](#_Toc447271459)

[1.4. 返回消息体响应（code节点） 4](#_Toc447271460)

[1.5. 常见字段说明 4](#_Toc447271461)

[2. 业务接口概览及使用说明 4](#_Toc447271462)

[2.1. 接口名称及对应的请求URL定义 4](#_Toc447271463)

[2.2. 使用说明及代码样例 6](#_Toc447271464)

[2.3. 请求类型 8](#_Toc447271465)

[3. *流水线接口*（http://） 10](#_Toc447271466)

[3.1. 新任务通知（/notice） 10](#_Toc447271474)

[*4.* *标准服务接口*（http://cloud.hwyun.com/dws-cloud/rt/ap/v1/pub/std） 11](#_Toc447271475)

[4.1. 问题反馈（热力图）（/heatmap/send） 11](#_Toc447271483)

[4.2. 发短信（/sms/send） 11](#_Toc447271484)

[4.3. 发邮件（/mail/send） 11](#_Toc447271485)

[*5.* *用户接口*（http://cloud.hwyun.com/dws-cloud/rt/ap/v1/user） 11](#_Toc447271486)

[5.1. 检查用户名重名（/chkname） 11](#_Toc447271494)

[5.2. 注册（/register） 12](#_Toc447271495)

[5.3. 登录（/login） 12](#_Toc447271496)

[5.4. 第三方登录 12](#_Toc447271497)

[5.5. 密码修改（/changepassword） 12](#_Toc447271498)

[5.6. 密码找回 12](#_Toc447271499)

[5.7. 账户信息修改（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/user/info/edit） 12](#_Toc447271500)

[*6.* *文件管理接口*（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/file） 13](#_Toc447271508)

[6.1. 快速识别（/quick/recog） 13](#_Toc447271516)

[6.2. 上传文件（/upload） 28](#_Toc447271517)

[6.3. 文件重命名(/rename) 31](#_Toc447271518)

[6.4. 文件列表（/list） 33](#_Toc447271519)

[6.5. 查看文件（/show） 34](#_Toc447271520)

[6.6. 下载文件（/download） 36](#_Toc447271521)

[6.7. 删除文件（/delete） 39](#_Toc447271522)

[6.8. 查询文件（/search） 41](#_Toc447271523)

[6.9. 文件校验(/checknum) 43](#_Toc447271524)

[*7.* *订单管理接口*（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/order） 44](#_Toc447271525)

[7.1. 价格评估（/evaluate） 44](#_Toc447271533)

[7.2. 添加订单（/add） 46](#_Toc447271534)

[7.3. 订单支付（/pay） 48](#_Toc447271535)

[7.4. 支付结果轮询（/pay/result/polling） 49](#_Toc447271536)

[7.5. 订单列表（/list） 50](#_Toc447271537)

[7.6. 查看订单（/show） 53](#_Toc447271538)

[7.7. 取消订单（/cancel） 55](#_Toc447271539)

[7.8. 删除订单（/delete） 57](#_Toc447271540)

[7.9. 查询订单（/search） 58](#_Toc447271541)

[*8.* *数字加工接口*（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/digital/processing） 61](#_Toc447271542)

[8.1. 任务轮询（/task/polling） 61](#_Toc447271550)

[8.2. 下载文件（/file/download） 63](#_Toc447271551)

[8.3. 修改任务状态（/task/status/edit） 65](#_Toc447271552)

[8.4. 上传文件（/file/upload） 66](#_Toc447271553)

[8.5. 完成（/task/finish） 70](#_Toc447271554)

# 前言

[汉王云](http://baike.baidu.com/view/12367260.htm)是汉王科技模式识别技术与互联网技术、云计算技术相结合的在线云服务产品，汉王云致力于成为移动互联时代模式识别云服务专业提供商。为了更好、更全面的为移动互联提供云识别技术，汉王云提供了公有云方案和私有云方案。汉王公有云分为B2B业务和B2C业务，B2B业务是对外广大的移动开发者提供服务的，B2C业务是对汉王科技内部的软件应用及硬件设备提供云API接入服务的。

详细可参考如下两个网站，并可以进行百度百科进行查询。

汉王公有云B2B服务是通过[汉王开发者中心](http://developer.hanvon.com/)来实现的，<http://developer.hanvon.com/>。可关注访问，已经正式对外提供服务。

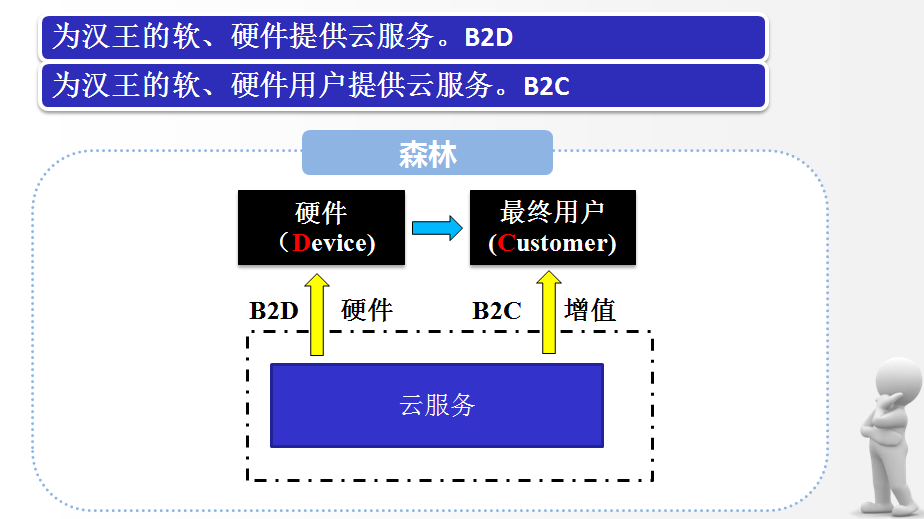
汉王公有云B2C服务是通过[汉王云网站](http://cloud.hanvon.com/)：<http://cloud.hanvon.com/> 来实现的，正在进行测试阶段。

汉王B2C云服务完全采用Restfull WebService，供汉王云网站、汉王云移动应用、汉王设备提供Web服务，其战略意义如下：

1. 让汉王的设备与移动智能终端互联互通。

2. 致力于将汉王的模式识别技术通过移动互联网转化为多种多样的有价值应用，为公司盈利。

3. 统一规范汉王所有的研发产品的互联网化，布局移动互联网时代战略。



本文档主要进行汉王公有云B2C业务的API接入说明。

若文档内容与实际执行效果有差异，请以产品为准进行开发。

# 客户端与服务器端的接口定义约定

## 基本规则

* 汉王云采用WebService方式对调用方提供API，目前此接口提供Rest实现方式。
* API实行标准REST(Representational State Transfer)接口规范，请参考REST相关文档。
* API入口地址：[http://](http://api.hanvoncloud.com/) 。以下所有的描述都是以此地址为基础。

注：调试中需访问测试地址：[http://](http://cloud.hwyun.com/aic-cloud/rt/ws/v1)rc.hwyun.com:9090/rws-cloud/ rt/ws/v1

* 正式环境中，我们的请求认证方式将全部使用https，以保证传输信息的安全性。
* 所有的请求和响应数据编码皆为UTF8格式，URL里的所有参数值最好请做urlencode编码。
* 请求消息体和应答消息体采用JSON编码方式，由header中format参数判定（为空时默认为json），如format="json"。响应消息体包含value，message节点分别表示响应码及响应说明，如 {**"**value**":"**11**","**message**":"**识别错误**"**}
* 所有来自客户端请求，需要在Http协议信息头 (Http Header)中增加以下字段信息。

|  |  |  |
| --- | --- | --- |
| 请求消息体 | | |
| 字段名称 | 是否必选 | 描述 |
| token | 否 | 用户登录后，服务器端自动分配一个token,在之后的所有请求时，都将使用此token. |

## 请求安全认证

采用标准https 方式进行所有请求的发送。

## 结果码定义

请求响应结果具体体现为两处：标准HTTP消息码响应与返回消息体响应（value节点），这两种响应具体如下：

后台返回标准HTTP消息码响应如下：

|  |  |
| --- | --- |
| 响应结果码 | 描述 |
| 100-190 | 信息提示，已定义100-101 |
| 200-299 | 成功，已定义200-206 |
| 300-399 | 服务器发生了重定向，已定义300-305 |
| 400-499 | 客户端错误，已定义400-415 |
| 500-599 | 服务器端错误，已定义500-505 |

## 返回消息体响应（code节点）

|  |  |
| --- | --- |
| 返回错误码 | 描述 |
| 0 | 操作成功 |
| 110 | 错误，具体错误信息见消息体message节点 |

## 常见字段说明

|  |  |
| --- | --- |
| 常见字段 | 解释 |
| oid | 订单唯一标识ID |
| data | 发送的具体数据都使用这一字段。 |

# 业务接口概览及使用说明

## 接口名称及对应的请求URL定义

| 编号 | URL | | | 接口描述 | 备注 |
| --- | --- | --- | --- | --- | --- |
| 1. 流水线接口 [http://](http://api.hanvoncloud.com/) | | | | |  |
|  | /notice | | | 用户有数据加工需求，发通知 | 任务通知接口 |
| 1. 标准服务接口 [http://](http://api.hanvoncloud.com/)cloud.hwyun.com/dws-cloud/rt/ap/v1/pub/std | | | | |  |
|  | /heatmap/send | | | 收集热力图数据 | 问题反馈接口 |
|  | /sms/send | | | 发短信 | 发短信接口 |
|  | /mail/send | | | 发邮件 | 发邮件接口 |
| 1. 用户接口 http://cloud.hwyun.com/dws-cloud/rt/ap/v1/user | | | | |  |
|  | | /chkname | 检查是否用户名重名 | | 检查用户名重名接口 |
|  | | /register | 密保手机或密保邮箱至少输入一个 | | 注册接口 |
|  | | /login | 登录 | | 登录接口 |
|  | | /qqregister | QQ用户登录 | | 第三方登录接口 |
|  | | /wxregister | 微信用户登录 | | 第三方登录接口 |
|  | | /wbregister | 新浪微博用户登录 | | 第三方登录接口 |
|  | | /changepassword | 修改密码 | | 密码重置接口 |
|  | | /getphoneauthcode | 手机获取验证码 | | 手机获取验证码接口 |
|  | | /checkphoneauthcode | 手机验证码校验 | | 手机验证码校验接口 |
|  | | /phoneResetPwd | 重置密码 | | 手机找回密码接口 |
|  | | /findPwdByEmail | 邮箱找回密码 | | 邮箱找回密码接口 |
|  | | http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/user /info/edit | 修改用户信息 | | 账户信息修改接口 |
| 1. 文件管理接口 http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/file | | | | |  |
|  | /quick/recog | | | 上传识别图像文件，根据用户选择的返回类型，立即返回相应结果。 | 快速识别接口  支持断点续传 |
|  | /upload | | | 支持多种文件的传输，一个或打包多个文件上传，存储到汉王云盘 | 识别文件上传接口  支持断点续传 |
|  | /rename | | | 修改文件名称 | 文件重命名接口 |
|  | /list | | | 获取文件列表，可根据订单ID、文件名查询 | 文件列表接口 |
|  | /show | | | 查看单个文件 | 查看文件接口 |
|  | /download | | | 下载一个或打包下载多个文件 | 识别结果文件下载接口  支持断点续传 |
|  | /delete | | | 删除文件 | 删除文件接口 |
|  | /search | | | 查询文件，可全文检索 | 文件搜索接口 |
|  | /checksum | | | 文件校验，采用SHA1算法 | 文件校验接口 |
| 1. 订单管理接口 http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/order | | | | |  |
|  | /evaluate | | | 根据文件全路径和文件总长度下载识别文件，进行价格估算 | 价格评估接口 |
|  | /add | | | 添加新订单 | 添加订单接口 |
|  | /edit/pay/status | | | 修改支付状态 | 修改支付状态接口 |
|  | /pay/result/polling | | | 查询付款结果 | 支付结果轮询接口 |
|  | /list | | | 获取订单列表，可根据订单ID、文件名查询 | 订单列表接口 |
|  | /show | | | 查看订单详情 | 查看订单接口 |
|  | /cancel | | | 修改订单状态为4-取消 | 取消订单接口 |
|  | /search | | | 查询订单，可全文检索 | 订单搜索接口 |
| 1. 数字加工接口http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/digital/processing | | | | |  |
|  | /task/polling | | | 从队列中查询是否有数据加工任务，有就返回订单ID等信息 | 任务轮询接口 |
|  | /file/dowload | | | 下载一个或打包下载多个文件 | 识别文件下载接口  支持断点续传 |
|  | /task/status/edit | | | 下载完识别文件后调用此接口，修改任务、订单状态为3-处理中 | 修改任务状态接口 |
|  | /file/upload | | | 支持多种文件的传输，存储到汉王云盘 | 识别结果文件上传接口  支持断点续传 |
|  | /task/finish | | | 完成数据加工任务 | 任务完成接口 |

## 使用说明及代码样例

* 所有的请求都以Post方式发过来。
* 请求体都是以json方式发过来，并且对请求串要进行base64化处理。
* 接收时进行反base64处理，解析的数据也为json格式。可以直接使用了。

代码参考(java)

/\*\*

\* 向指定的URL的post数据

\* @param urlStr

\* url字符串

\* http://write.hanvoncloud.com/rest/recognize

\* @param requestData

\* 数据

\* requestData = "{\" uid\":\"95841248414524\",\" result\":\"19968\",\" type\":\"1\",\" data\":\"100,101,102,103,99,98,-1,0\"}";

\*/

publicstatic String postDataToCloud(String urlStr, String requestData) {

String result = null;

URL url = null;

HttpURLConnection httpurlconnection = null;

try {

url = new URL(urlStr);

httpurlconnection = (HttpURLConnection) url.openConnection();

httpurlconnection.setDoInput(true);

httpurlconnection.setDoOutput(true);

//1.设备请求类型

httpurlconnection.setRequestMethod("POST");

//2.设备请求头

//

httpurlconnection.setRequestProperty("Token",

"");

httpurlconnection.setRequestProperty("Content-Type",

"application/octet-stream");

//3加密请求数据

requestData = Base64.*encodeBase64String*(requestData.getBytes());

httpurlconnection.getOutputStream().write(requestData.getBytes());

//5发送数据

httpurlconnection.getOutputStream().flush();

httpurlconnection.getOutputStream().close();

//5接收结果数据

InputStream in = httpurlconnection.getInputStream();

BufferedReader r = new BufferedReader(new InputStreamReader(in,

"utf-8"));

String line;

StringBuilder sb = new StringBuilder();

while ((line = r.readLine()) != null) {

sb.append(line);

}

//6解密处理数据

result = new String(Base64.*decodeBase64*(sb.toString().getBytes()));

//7使用数据

System.*out*.println(result);

} catch (Exception e) {

e.printStackTrace();

} finally {

if (httpurlconnection != null) {

httpurlconnection.disconnect();

}

}

return result;

}

## 请求类型

所有的请求分为有状态和无状态两种，对于无状态即为不需要用户登录，可以直接使用。对于有状态，则要求用户是登录的。

用户登录方式有三种，可以来自汉王的硬件设备，可以来自网站，也可以来自移动应用客户端。根据下面业务交互图可以分别调用相应的接口进行接入。返回的token是用户进行访问的关键判断，此token对于每次登录都是不同的。

# *流水线接口*（http://）



## 新任务通知（/notice）



### 接口描述

用户上传图片，云端验证后接单，通知流水线需要进行数据加工。

### 请求方法：

POST /URL

Header 无要求

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| oid | reqeust json | 1 | 否 | String | 订单唯一标识ID |
| ftype | reqeust json | 1 | 否 | String | 输出文档类型  0-txt  1-pdf |
| ptype | reqeust json | 1 | 否 | String | 输入图片类型  0-简体中文 |
| level | reqeust json | 1 | 是 | String | 优先级（数字越小优先级越高）  1-一级  2-二级  3-三级…… |
| size | reqeust json | 1 | 否 | String | 文件总长度（字节长度） |
| iszip | reqeust json | 1 | 否 | String | 是否为zip包（true | false） |
| pamount | reqeust json | 1 | 否 | String | 图片数量 |
| checksum | reqeust json | 1 | 否 | String | 校验值（下载文件整体校验） |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | 无 |

### 请求示例(json)：

{**"o**id**":"32545fds462475sd","ftype":"0","**ptype**":"**0**","**level**":"1","size":"1634","iszip":"false","pamount":"1","checksum":"**06c09ae14ba134a3eaf8063c9903b682**"**}

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |

### 返回值示例(json)：

Success：{**"**code**":"**0**","**result**":"OK"**}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 不重名 | /notice |
| 420 | 输入的json串不合法 | /notice |
| 520 | 服务器有异常发生 | /notice |

# *标准服务接口*（http://cloud.hwyun.com/dws-cloud/rt/ap/v1/pub/std）



## 问题反馈（热力图）（/heatmap/send）

详见《汉王云B2C服务API-V1.1》文档 8.1.1

## 发短信（/sms/send）

详见《汉王云B2C服务API-V1.1》文档 8.4.1

## 发邮件（/mail/send）

详见《汉王云B2C服务API-V1.1》文档 8.5.1

# *用户接口*（http://cloud.hwyun.com/dws-cloud/rt/ap/v1/user）



## 检查用户名重名（/chkname）

详见《汉王云B2C服务API-V1.1》文档 3.1

## 注册（/register）

详见《汉王云B2C服务API-V1.1》文档 3.2

## 登录（/login）

详见《汉王云B2C服务API-V1.1》文档 3.4

## 第三方登录

### QQ用户登录（/qqregister）

详见《汉王云B2C服务API-V1.1》文档 3.6

### 微信用户登录（/wxregister）

详见《汉王云B2C服务API-V1.1》文档 3.7

### 新浪微博用户登录（/wbregister）

详见《汉王云B2C服务API-V1.1》文档 3.8

## 密码修改（/changepassword）

详见《汉王云B2C服务API-V1.1》文档 3.10

## 获取用户信息（/getcontractinfo）

详见《汉王云B2C服务API-V1.1》文档 3.9

## 密码找回

### 手机找回密码

a. 获取验证码（/getphoneauthcode）

详见《汉王云B2C服务API-V1.1》文档 3.14

b. 检查验证码有效性（/checkphoneauthcode）

详见《汉王云B2C服务API-V1.1》文档 3.15

c. 重置密码（/phoneResetPwd）

详见《汉王云B2C服务API-V1.1》文档 3.16

### 邮箱找回密码（/findPwdByEmail）

详见《汉王云B2C服务API-V1.1》文档 3.18

## 后台统计数据收集 （http://cloud.hwyun.com/dws-cloud/rt/ap/v1/statistics/gather）

### 设备信息（/devicestat）

详见《汉王云B2C服务API-V1.1》文档 15.1

### 用户功能行为信息（/functionstat）

详见《汉王云B2C服务API-V1.1》文档 15.2

## 获取订单收货人信息（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/user/get/order/contacts/info）

### 接口描述

获取用户最新修改的一条订单收货人信息。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 否 | String | 用户唯一标识 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不能为空 |

### 请求示例(json)：

{"userid":"test2345"}

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | string | 消息内容 |
| contactId | response json | 1 | string | 联系信息唯一标识ID |
| fullname | response json | 1 | string | 用户姓名 |
| mobile | response json | 1 | string | 联系电话 |
| email | response json | 1 | string | 邮箱（保留字段） |

### 返回值示例(json)：

Success：{"code":"0","result":"success","id":"2","fullname":"张三","mobile":"13511487958"}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /get/order/contacts/info |
| 420 | 输入的json串不合法 | /get/order/contacts/info |
| 421 | 请求存在非法为空项，如：userid | /get/order/contacts/info |
| 426 | 联系信息不存在 | /get/order/contacts/info |
| 520 | | | 服务器有异常发生 | /get/order/contacts/info |

/

## 订单收货人信息修改（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/user/edit/order/contacts/info）

### 接口描述

下单之前，可修改订单收货人信息。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 否 | String | 用户唯一标识 |
| fullname | request | 1 | 是 | String | 姓名 |
| mobile | request | 1 | 是 | String | 手机号 |
| email | request | 1 | 是 | String | 邮箱（保留字段） |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不能为空 |

### 请求示例(json)：

{"userid":"test2345","fullname":"张三","mobile":"13521865024"}

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | string | 消息内容 |
| contactId | response json | 1 | string | 联系信息唯一标识ID |

### 返回值示例(json)：

Success：{"code":"0","result":"success"}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /edit/order/contacts/info |
| 420 | 输入的json串不合法 | /edit/order/contacts/info |
| 421 | 请求存在非法为空项，如：userid | /edit/order/contacts/info |
| 520 | 服务器有异常发生 | /edit/order/contacts/info |



# *文件管理接口*（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/file）



## 快速识别（/quick/recog）

### 接口描述

上传识别图像文件，根据用户选择的返回类型，返回相应结果。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | request header | 1 | 否 | String | 用户唯一标识ID  不登录时，APP端给设备唯一标识ID；WEB端给anonymous；PC端如果有唯一标识ID，就给唯一标识ID，若没有也给anonymous |
| resType | request header | 1 | 否 | String | 结果类型  1-plain text  2-json data  3- 加强版json data  4--txt  5-rtf  6-single pdf  7-double pdf  8-excel  …… |
| platformType | request header | 1 | 否 | String | 平台类型  1-Web  2-PC  3-iOS  4-Android  …… |
| fileName | request header | 1 | 否 | String | 文件全称  带后缀（可带路径），使用 URLEncoder.encode转码 |
| fileFormat | request header | 1 | 否 | String | 输入文件格式  jpg | png 等  多个用逗号（,）分隔 |
| size | request header | 1 | 否 | String | 文件总长度 |
| length | request header | 1 | 否 | String | 每次上传数据的长度（字节长度）  注：最后一次是文件总长度的剩余长度 |
| offset | request header | 1 | 否 | String | 每次写入的offset  第一次为0 |
| checksum | request header | 1 | 是 | String | 校验值（每次上传数据的校验）  SHA1 (Base64(data)) |
| data | request header | 1 | 否 | byte[] | 字节数组 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不可为空 |

### 请求示例(multipart/form-data)：

Java示例代码，如下：

@Test

public void upload(){

String result = null;

// URL

String url = "http://localhost:8080/rws-cloud/rt/ws/v1/file/upload";

File file = new File("E:/160202.txt");

// 上传参数

Map<String, String> params = new HashMap<String, String>();

params.put("userid", userId);

// 结果类型

params.put("resType", "1");

// 平台类型

params.put("platformType", "1");

params.put("fileName", URLEncoder.encode("160202", "UTF-8"));

params.put("fileFormat", "jpg");

// 文件总长度

params.put("size", file.length());

InputStream inputStream = null;

String offset = "0";

try {

inputStream = new FileInputStream(file);

int temLength = 1024\*1024;

byte[] temp = new byte[temLength];

// 上传数据

byte[] data = null;

// 是否为打包文件

params.put("iszip", "false");

for(int len = inputStream.read(temp); len != -1; len = inputStream.read(temp)){

String tempData = "";

if(len < temLength){

byte[] tempArray = new byte[len];

for(int i = 0; i < tempArray.length; i ++){

tempArray[i] = temp[i];

}

tempData = Base64.encode(tempArray);

data = tempArray;

// 每次上传数据的长度

params.put("length", String.valueOf(len));

}else{

tempData = Base64.encode(temp);

data = temp;

// 每次上传数据的长度

params.put("length", String.valueOf(temLength));

}

String checksum = SHA1Util.encodeBySHA(tempData);

// 每次上传数据的偏移量

params.put("offset", offset);

// 每次上传数据的校验和

params.put("checksum", checksum);

result = doPost(params, data, url);

UploadResponse responseBean = JsonUtil.getObjectFromJson(result, UploadResponse.class);

offset = responseBean.getOffset();

}

} catch (Exception e) {

e.printStackTrace();

}

}

private String doPost(Map<String, String> params, byte[] data, String urlStr) throws Exception {

String result = null;

try {

MultipartEntityBuilder mEntityBuilder = MultipartEntityBuilder.create();

// 每次上传的数据

ByteArrayBody dataBody = new ByteArrayBody(data,

ContentType.MULTIPART\_FORM\_DATA, "uploadTemp");

mEntityBuilder.addPart("file", dataBody);

mEntityBuilder.setBoundary("----------ThIs\_Is\_tHe\_bouNdaRY\_$");

mEntityBuilder.setMode(HttpMultipartMode.BROWSER\_COMPATIBLE);

mEntityBuilder.setCharset(Charset.defaultCharset());

// 参数准备

if (params != null) {

for (Entry<String, String> e : params.entrySet()) {

mEntityBuilder.addTextBody(e.getKey(), e.getValue());

}

}

HttpPost httpPost = new HttpPost(urlStr);

httpPost.addHeader("Content-Type", MediaType.MULTIPART\_FORM\_DATA

+ "; boundary=----------ThIs\_Is\_tHe\_bouNdaRY\_$");

// 设置参数

httpPost.setEntity(mEntityBuilder.build());

// 创建默认的httpClient实例.

HttpClient httpClient = new DefaultHttpClient();

HttpResponse response = httpClient.execute(httpPost);

int statusCode = response.getStatusLine().getStatusCode();

if (statusCode != 200) {

httpPost.abort();

throw new RuntimeException("HttpClient, response :" + response);

}

HttpEntity entity = response.getEntity();

if (entity != null) { // 返回JSON结果

result = EntityUtils.toString(entity, "utf-8");

httpPost.abort();

}

EntityUtils.consume(entity);

} catch (Exception e) {

e.printStackTrace();

}

return result;

}

### 响应消息体：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| offset | response | 1 | integer | 文件成功上传偏移值。单位：字节 |
| 当resType = 1 时 | | | | |
| result | response | 1 | string | 识别出的文本内容  Base64转码 |
| fid | response | 1 | integer | 识别文件的唯一标识ID |
| 当resType = 2时 | | | | |
| result | response | 1 | string | 消息内容 |
| fid | response | 1 | integer | 识别文件的唯一标识ID |
| block\_count | response | 1 | integer | 文本块总数 |
| table\_count | response | 1 | integer | 表格总数 |
| block\_list | response | 1 | array | 文本块集合 |
| order | block\_list | 1 | string | 在全篇文本的块顺序号 |
| line\_list | block\_list | 1 | array | 一个文本块里行的集合 |
| line\_num | line\_list | 1 | integer | 一个文本块的行号 |
| text | line\_list | 1 | string | 识别出来的文字内容，第一候选 |
| table\_list | response | 1 | array | 表格集合 |
| order | table\_list | 1 | integer | 一个表格在全篇文本的块顺序号 |
| tr\_count | table\_list | 1 | integer | 一个表格行数 |
| td\_count | table\_list | 1 | integer | 一个表格列数 |
| table | table\_list | 1 | array | 表示一个表格 |
| tr | table | 1 | array | 一个表格里的一行 |
| td | tr | 1 | object | 某一行里的一列 |
| colspan | td | 1 | integer | 单元格可横跨的列数 |
| rowspan | td | 1 | integer | 单元格可纵跨的行数 |
| line\_list | td | 1 | array | 一个单元格里行的集合 |
| line\_num | line\_list | 1 | integer | 一个文本块的行号 |
| text | line\_list | 1 | string | 识别出来的文字内容，第一候选 |
| 当resType = 3时 | | | | |
| result | response | 1 | string | 消息内容 |
| fid | response | 1 | integer | 识别文件的唯一标识ID |
| block\_count | response | 1 | integer | 文本块总数 |
| table\_count | response | 1 | integer | 表格总数 |
| block\_list | response | 1 | array | 文本块集合 |
| order | block\_list | 1 | string | 在全篇文本的块顺序号 |
| line\_list | block\_list | 1 | array | 一个文本块里行的集合 |
| line\_num | line\_list | 1 | integer | 一个文本块的行号 |
| text | line\_list | 1 | string | 识别出来的文字内容，第一候选 |
| confidence | line\_list | 1 | string | 每个文字的置信度，从A到E，置信度依次降低 |
| pos | line\_list | 1 | string | 每个文字的位置，文字间用“|”分隔，位置之间用“-”分隔 |
| table\_list | response | 1 | array | 表格集合 |
| order | table\_list | 1 | integer | 一个表格在全篇文本的块顺序号 |
| tr\_count | table\_list | 1 | integer | 一个表格行数 |
| td\_count | table\_list | 1 | integer | 一个表格列数 |
| table | table\_list | 1 | array | 表示一个表格 |
| tr | table | 1 | array | 一个表格里的一行 |
| td | tr | 1 | 对象 | 某一行里的一列 |
| colspan | td | 1 | integer | 单元格可横跨的列数 |
| rowspan | td | 1 | integer | 单元格可纵跨的行数 |
| line\_list | td | 1 | 数组 | 一个单元格里行的集合 |
| line\_num | line\_list | 1 | 数值 | 一个文本块的行号 |
| text | line\_list | 1 | 字符串 | 识别出来的文字内容，第一候选 |
| confidence | line\_list | 1 | 字符串 | 每个文字的置信度，从A到E，置信度依次降低 |
| pos | line\_list | 1 | 字符串 | 每个文字的位置，文字间用“|”分隔，位置之间用“-”分隔 |
| 当resType > 3时 | | | | |
| result | response | 1 | string | 消息内容 |
| fid | response | 1 | 字符串 | 识别结果文件唯一标识ID |

### 返回值示例(json)：

Success：

resType = 1，如下：

{**"**code**":"**0**","**result**":"kfkssdkfks32323","offset":"1432"**}

resType = 2，如下：

{  
 "code": "0",  
 "result": "success",  
 **"offset":"1432",**  
 "block\_count": "2",  
 "table\_count": "1",  
 "block\_list":   
 [  
 {  
 "order": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "表头1表头2"  
 }  
 ]  
 },  
 {  
 "order": "3",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "表尾"  
 }  
 ]  
 }  
 ],  
 "table\_list":   
 [  
 {  
 "order": "2",  
 "tr\_count": "3",  
 "td\_count": "3",  
 "table":   
 [  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第一列"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "笔一列"  
 }  
 ]  
 }  
 }  
 ]  
 },  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第一行"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "11"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "12"  
 }  
 ]  
 }  
 }  
 ]  
 },  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第二行"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "21"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "22"  
 }  
 ]  
 }  
 }  
 ]  
 }  
 ]  
 }  
 ]  
}

resType = 3，如下：

{  
 "code": "0",  
 "result": "success",  
 **"offset":"1432",**  
 "block\_count": "2",  
 "table\_count": "1",  
 "block\_list":   
 [  
 {  
 "order": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "表头1表头2",  
 "confidence": "AAAAAA",  
 "pos": "160-42-192-74|197-42-229-75|237-47-246-71|466-42-498-74|503-42-535-75|540-47-554-71"  
 }  
 ]  
 },  
 {  
 "order": "3",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "表尾",  
 "confidence": "AA",  
 "pos": "323-354-355-385|360-355-392-385"  
 }  
 ]  
 }  
 ],  
 "table\_list":   
 [  
 {  
 "order": "2",  
 "tr\_count": "3",  
 "td\_count": "3",  
 "table":   
 [  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第一列",  
 "confidence": "AAA",  
 "pos": "304-145-335-177|341-157-373-160|378-145-409-177"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "笔一列",  
 "confidence": "DAA",  
 "pos": "507-145-537-171|547-151-572-152|581-145-611-171"  
 }  
 ]  
 }  
 }  
 ]  
 },  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第一行",  
 "confidence": "AAA",  
 "pos": "102-198-133-230|139-210-171-213|176-198-208-230"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "11",  
 "confidence": "AA",  
 "pos": "344-203-353-227|363-203-372-227"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "12",  
 "confidence": "AA",  
 "pos": "546-203-555-227|562-203-576-227"  
 }  
 ]  
 }  
 }  
 ]  
 },  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第二行",  
 "confidence": "AAA",  
 "pos": "102-251-133-283|139-257-171-278|176-251-208-283"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "21",  
 "confidence": "AA",  
 "pos": "341-256-355-280|363-256-372-280"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "22",  
 "confidence": "AA",  
 "pos": "543-256-557-280|562-256-576-280"  
 }  
 ]  
 }  
 }  
 ]  
 }  
 ]  
 }  
 ]  
}

resType > 3，如下：

{**"**code**":"**0**","**result**":"** success**","offset":"1432","fid":"**dsafasfa345135"}

再通过fid，调用下载接口，下载识别结果文档。

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured","offset":""**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /quick/recognize |
| 8001 | 识别服务错误 | /quick/recognize |
| 8002 | 图片样本不够清晰 | /quick/recognize |
| 421 | 请求存在非法为空项，如：userid | /quick/recognize |
| 429 | 上传文件的长度错误 | /quick/recognize |
| 432 | data 为空 | /quick/recognize |
| 441 | 删除文件失败 | /quick/recognize |
| 520 | 服务器有异常发生 | /quick/recognize |
| 524 | Checksum校验失败 | /quick/recognize |

## 快速识别（/web/rapid/recog）

### 接口描述

用户上传识别图像文件后，调用此接口，根据用户选择的返回类型，返回相应结果。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | request json | 1 | 否 | String | 用户唯一标识ID  不登录时，APP端给设备唯一标识ID；WEB端给anonymous；PC端如果有唯一标识ID，就给唯一标识ID，若没有也给anonymous |
| fid | request json | 1 | 否 | String | 文件唯一标识ID  识别文件 |
| resType | request json | 1 | 否 | String | 结果类型  1-plain text  2-json data  3- 加强版json data  4--txt  5-rtf  6-single pdf  7-double pdf  8-excel  …… |
| platformType | request json | 1 | 否 | String | 平台类型  1-Web  2-PC  3-iOS  4-Android  …… |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、 fid、resType、platformType 不能为空 |

### 请求示例(json)：

{"userid":"anonymous","fid":"dfas4faf45235646wer4","resType":"4","platformType":"1"}

### 响应消息体：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| offset | response | 1 | integer | 文件成功上传偏移值。单位：字节 |
| 当resType = 1 时 | | | | |
| result | response | 1 | string | 识别出的文本内容  Base64转码 |
| 当resType = 2时 | | | | |
| result | response | 1 | string | 消息内容 |
| block\_count | response | 1 | integer | 文本块总数 |
| table\_count | response | 1 | integer | 表格总数 |
| block\_list | response | 1 | array | 文本块集合 |
| order | block\_list | 1 | string | 在全篇文本的块顺序号 |
| line\_list | block\_list | 1 | array | 一个文本块里行的集合 |
| line\_num | line\_list | 1 | integer | 一个文本块的行号 |
| text | line\_list | 1 | string | 识别出来的文字内容，第一候选 |
| table\_list | response | 1 | array | 表格集合 |
| order | table\_list | 1 | integer | 一个表格在全篇文本的块顺序号 |
| tr\_count | table\_list | 1 | integer | 一个表格行数 |
| td\_count | table\_list | 1 | integer | 一个表格列数 |
| table | table\_list | 1 | array | 表示一个表格 |
| tr | table | 1 | array | 一个表格里的一行 |
| td | tr | 1 | object | 某一行里的一列 |
| colspan | td | 1 | integer | 单元格可横跨的列数 |
| rowspan | td | 1 | integer | 单元格可纵跨的行数 |
| line\_list | td | 1 | array | 一个单元格里行的集合 |
| line\_num | line\_list | 1 | integer | 一个文本块的行号 |
| text | line\_list | 1 | string | 识别出来的文字内容，第一候选 |
| 当resType = 3时 | | | | |
| result | response | 1 | string | 消息内容 |
| block\_count | response | 1 | integer | 文本块总数 |
| table\_count | response | 1 | integer | 表格总数 |
| block\_list | response | 1 | array | 文本块集合 |
| order | block\_list | 1 | string | 在全篇文本的块顺序号 |
| line\_list | block\_list | 1 | array | 一个文本块里行的集合 |
| line\_num | line\_list | 1 | integer | 一个文本块的行号 |
| text | line\_list | 1 | string | 识别出来的文字内容，第一候选 |
| confidence | line\_list | 1 | string | 每个文字的置信度，从A到E，置信度依次降低 |
| pos | line\_list | 1 | string | 每个文字的位置，文字间用“|”分隔，位置之间用“-”分隔 |
| table\_list | response | 1 | array | 表格集合 |
| order | table\_list | 1 | integer | 一个表格在全篇文本的块顺序号 |
| tr\_count | table\_list | 1 | integer | 一个表格行数 |
| td\_count | table\_list | 1 | integer | 一个表格列数 |
| table | table\_list | 1 | array | 表示一个表格 |
| tr | table | 1 | array | 一个表格里的一行 |
| td | tr | 1 | 对象 | 某一行里的一列 |
| colspan | td | 1 | integer | 单元格可横跨的列数 |
| rowspan | td | 1 | integer | 单元格可纵跨的行数 |
| line\_list | td | 1 | 数组 | 一个单元格里行的集合 |
| line\_num | line\_list | 1 | 数值 | 一个文本块的行号 |
| text | line\_list | 1 | 字符串 | 识别出来的文字内容，第一候选 |
| confidence | line\_list | 1 | 字符串 | 每个文字的置信度，从A到E，置信度依次降低 |
| pos | line\_list | 1 | 字符串 | 每个文字的位置，文字间用“|”分隔，位置之间用“-”分隔 |
| 当resType > 3时 | | | | |
| result | response | 1 | string | 消息内容 |
| fid | response | 1 | 字符串 | 识别结果文件唯一标识ID |

### 返回值示例(json)：

Success：

resType = 1，如下：

{**"**code**":"**0**","**result**":"**=safjewotupjfalsfj**"**}

resType = 2，如下：

{  
 "code": "0",  
 "result": "success"**,**  
 "block\_count": "2",  
 "table\_count": "1",  
 "block\_list":   
 [  
 {  
 "order": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "表头1表头2"  
 }  
 ]  
 },  
 {  
 "order": "3",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "表尾"  
 }  
 ]  
 }  
 ],  
 "table\_list":   
 [  
 {  
 "order": "2",  
 "tr\_count": "3",  
 "td\_count": "3",  
 "table":   
 [  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第一列"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "笔一列"  
 }  
 ]  
 }  
 }  
 ]  
 },  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第一行"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "11"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "12"  
 }  
 ]  
 }  
 }  
 ]  
 },  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第二行"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "21"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "22"  
 }  
 ]  
 }  
 }  
 ]  
 }  
 ]  
 }  
 ]  
}

resType = 3，如下：

{  
 "code": "0",  
 "result": "success"  
 "block\_count": "2",  
 "table\_count": "1",  
 "block\_list":   
 [  
 {  
 "order": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "表头1表头2",  
 "confidence": "AAAAAA",  
 "pos": "160-42-192-74|197-42-229-75|237-47-246-71|466-42-498-74|503-42-535-75|540-47-554-71"  
 }  
 ]  
 },  
 {  
 "order": "3",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "表尾",  
 "confidence": "AA",  
 "pos": "323-354-355-385|360-355-392-385"  
 }  
 ]  
 }  
 ],  
 "table\_list":   
 [  
 {  
 "order": "2",  
 "tr\_count": "3",  
 "td\_count": "3",  
 "table":   
 [  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第一列",  
 "confidence": "AAA",  
 "pos": "304-145-335-177|341-157-373-160|378-145-409-177"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "笔一列",  
 "confidence": "DAA",  
 "pos": "507-145-537-171|547-151-572-152|581-145-611-171"  
 }  
 ]  
 }  
 }  
 ]  
 },  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第一行",  
 "confidence": "AAA",  
 "pos": "102-198-133-230|139-210-171-213|176-198-208-230"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "11",  
 "confidence": "AA",  
 "pos": "344-203-353-227|363-203-372-227"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "12",  
 "confidence": "AA",  
 "pos": "546-203-555-227|562-203-576-227"  
 }  
 ]  
 }  
 }  
 ]  
 },  
 {  
 "tr":   
 [  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "第二行",  
 "confidence": "AAA",  
 "pos": "102-251-133-283|139-257-171-278|176-251-208-283"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "21",  
 "confidence": "AA",  
 "pos": "341-256-355-280|363-256-372-280"  
 }  
 ]  
 }  
 },  
 {  
 "td":   
 {  
 "colspan": "1",  
 "rowspan": "1",  
 "line\_list":   
 [  
 {  
 "line\_num": "0",  
 "text": "22",  
 "confidence": "AA",  
 "pos": "543-256-557-280|562-256-576-280"  
 }  
 ]  
 }  
 }  
 ]  
 }  
 ]  
 }  
 ]  
}

resType > 3，如下：

{**"**code**":"**0**","**result**":"** success**","fid":"**dsafasfa345135"}

再通过fid，调用下载接口，下载识别结果文档。

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /rapid/recog |
| 8001 | 识别服务错误 | /rapid/recog |
| 8002 | 图片样本不够清晰 | /rapid/recog |
| 8003 | 识别错误，请稍后再试 | /rapid/recog |
| 420 | 输入的json串不合法 | /rapid/recog |
| 421 | 请求存在非法为空项，如：userid | /rapid/recog |
| 431 | 识别文件不存在 | /rapid/recog |
| 520 | 服务器有异常发生 | /rapid/recog |

## 上传文件（/upload）

### 接口描述

支持多种文件的传输，一个或打包多个文件上传，存储到汉王云盘

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | request header | 1 | 否 | String | 用户唯一标识ID  不登录时，APP端给设备唯一标识ID；WEB端给anonymous；PC端如果有唯一标识ID，就给唯一标识ID，若没有也给anonymous |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准  不能为空 |
| fileType | request header | 1 | 否 | String | 文件类型  1-识别文件  2-识别结果文件 |
| fid | request header | 1 | 是 | String | 文件唯一标识ID  首次上传为空值；修改后上传，需传值，覆盖已存在文件 |
| fileName | request header | 1 | 否 | String | 文件全称  带后缀（可带路径），使用 URLEncoder.encode转码 |
| fileFormat | request header | 1 | 否 | String | 文件格式  识别文件：jpg | png 等  结果文件：txt | pdf 等  多个用逗号（,）分隔 |
| fileAmount | request header | 1 | 否 | String | 文件数量 |
| size | request header | 1 | 否 | String | 文件总长度 |
| length | request header | 1 | 否 | String | 每次上传数据的长度（字节长度）  注：最后一次是文件总长度的剩余长度 |
| offset | request header | 1 | 否 | String | 每次写入的offset  第一次为0 |
| checksum | request header | 1 | 是 | String | 校验值（每次上传数据的校验）  SHA1 (Base64(data)) |
| iszip | request header | 1 | 否 | String | 是否为zip包（true | false） |
| data | request header | 1 | 否 | byte[] | 字节数组 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、recogType不为空 |

### 请求示例(multipart/form-data)：

Java模拟调用rest 上传API示例代码，如下：

@Test

public void upload(){

String result = null;

// URL

String url = "http://localhost:8080/rws-cloud/rt/ws/v1/file/upload";

File file = new File("E:/测试图像.jpg");

// 上传参数

Map<String, String> params = new HashMap<String, String>();

params.put("userid", userid);

params.put("recogType", "1");

params.put("fileType", "1"); // 识别文件

params.put("fileAmount", "1");

params.put("fileName", URLEncoder.encode("160202", "UTF-8"));

params.put("fileFormat", "jpg");

// 文件总长度

params.put("size", file.length());

InputStream inputStream = null;

String offset = "0";

try {

inputStream = new FileInputStream(file);

int temLength = 1024\*1024;

byte[] temp = new byte[temLength];

// 上传数据

byte[] data = null;

// 是否为打包文件

params.put("iszip", "false");

for(int len = inputStream.read(temp); len != -1; len = inputStream.read(temp)){

String tempData = "";

if(len < temLength){

byte[] tempArray = new byte[len];

for(int i = 0; i < tempArray.length; i ++){

tempArray[i] = temp[i];

}

tempData = Base64.encode(tempArray);

data = tempArray;

// 每次上传数据的长度

params.put("length", String.valueOf(len));

}else{

tempData = Base64.encode(temp);

data = temp;

// 每次上传数据的长度

params.put("length", String.valueOf(temLength));

}

String checksum = SHA1Util.encodeBySHA(tempData);

// 每次上传数据的偏移量

params.put("offset", offset);

// 每次上传数据的校验和

params.put("checksum", checksum);

result = doPost(params, data, url);

UploadResponse responseBean = JsonUtil.getObjectFromJson(result, UploadResponse.class);

offset = responseBean.getOffset();

}

} catch (Exception e) {

e.printStackTrace();

}

}

private String doPost(Map<String, String> params, byte[] data, String urlStr) throws Exception {

String result = null;

try {

MultipartEntityBuilder mEntityBuilder = MultipartEntityBuilder.create();

// 每次上传的数据

ByteArrayBody dataBody = new ByteArrayBody(data,

ContentType.MULTIPART\_FORM\_DATA, "uploadTemp");

mEntityBuilder.addPart("file", dataBody);

mEntityBuilder.setBoundary("----------ThIs\_Is\_tHe\_bouNdaRY\_$");

mEntityBuilder.setMode(HttpMultipartMode.BROWSER\_COMPATIBLE);

mEntityBuilder.setCharset(Charset.defaultCharset());

// 参数准备

if (params != null) {

for (Entry<String, String> e : params.entrySet()) {

mEntityBuilder.addTextBody(e.getKey(), e.getValue());

}

}

HttpPost httpPost = new HttpPost(urlStr);

httpPost.addHeader("Content-Type", MediaType.MULTIPART\_FORM\_DATA

+ "; boundary=----------ThIs\_Is\_tHe\_bouNdaRY\_$");

// 设置参数

httpPost.setEntity(mEntityBuilder.build());

// 创建默认的httpClient实例.

HttpClient httpClient = new DefaultHttpClient();

HttpResponse response = httpClient.execute(httpPost);

int statusCode = response.getStatusLine().getStatusCode();

if (statusCode != 200) {

httpPost.abort();

throw new RuntimeException("HttpClient, response :" + response);

}

HttpEntity entity = response.getEntity();

if (entity != null) { // 返回JSON结果

result = EntityUtils.toString(entity, "utf-8");

httpPost.abort();

}

EntityUtils.consume(entity);

} catch (Exception e) {

e.printStackTrace();

}

return result;

}

### 响应消息体：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |
| fid | response | 1 | string | 文件唯一标识ID  需要保存，以便修改后上传时使用。 |
| offset | response | 1 | string | 文件成功上传偏移值。单位：字节 |
| errorFileName | response | 1 | String | 不能识别文件名  多个逗号拼接 |

### 返回值示例(json)：

Success：{**"**code**":"**0**","**result**":"** success**","fid":"sdkfkssdkfks32323","offset":"1024",”**errorFileName**”:”a1.jpg,2副本.bmp”**}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured","oid":"","offset":""**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /upload |
| 421 | 请求存在非法为空项，如：userid、recogType | /upload |
| 429 | 上传文件的长度错误 | /upload |
| 430 | fileType 为空 | /upload |
| 432 | data 为空 | /upload |
| 441 | 删除文件失败 | /upload |
| 520 | 服务器有异常发生 | /upload |
| 524 | Checksum校验失败 | /upload |

## 文件重命名(/rename)

### 接口描述

修改文件名称

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | 无 | 1 | 是 | String | 用户唯一标识ID  不登录时，APP端给设备唯一标识ID；WEB端给anonymous；PC端如果有唯一标识ID，就给唯一标识ID，若没有也给anonymous |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准  不能为空 |
| fileType | request | 1 | 否 | String | 文件类型  1-识别文件  2-识别结果文件 |
| fid | 无 | 1 | 否 | String | 文件唯一标识ID |
| fileName | 无 | 1 | 否 | String | 新文件名（不带后缀，不能包含路径） |

### 请求体特别约束

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | recogType不为空 |

### 请求示例

{"userid":"john","recogType":"2","fileType":"1", "fid":"sdkfajs34239090sssd","fileName":"图像1.jpg "}

### 响应消息体

| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |

### 返回值示例

Success：{**"**code**":"**0**","**result**":"** success **"**}

Failure：{**"**code**":"**1000**","**result**":"Xxexcption occurred"**}

### 响应码列表

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /rename |
| 420 | 输入的json串不合法 | /rename |
| 421 | 请求存在非法为空项，如：userid、recogType | /rename |
| 424 | token非法 | /rename |
| 426 | 客户端提供的源目录不存在 | /rename |
| 520 | 服务器有异常发生 | /rename |

## 文件列表（/list）

### 接口描述

获取文件列表。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 否 | String | 用户唯一标识ID |
| start | request | 1 | 否 | String | 起始位置 |
| pageSize | request | 1 | 否 | String | 每页最大记录数 |
| sort | request | 1 | 是 | String json | 默认按最后创建时间降序排列，获取到最新的。可扩展，  **{"create**\_time**"**:**"**desc**"}**降序  **{"create**\_time**"**:**"**asc**"}**升序 |
| fileType | request | 1 | 否 | String | 文件类型  1-识别文件  2-识别结果文件 |
| recogType | request | 1 | 是 | String | 识别类型  1-快速识别 2-人工精准  为空时，查询所有识别类型的识别结果文件列表。 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不能为空 |

### 请求示例(json)：

{"userid":"test2345","start":"0","pageSize":"10","sort":{"**create**\_time":"desc"},"fileType":"2","recogType":"2"}

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | string | 消息内容 |
| list | response json | 1 | json array | 文件列表 |
| fid | list | 1 | String | 文件唯一标识ID |
| fileName | list | 1 | String | 文件名 |
| fileType | list | 1 | string | 文件类型 |
| fileAmount | list | 1 | string | 文件数量 |
| fileSize | list | 1 | string | 文件大小（字节） |
| downloadFlag | list | 1 | string | 下载标识  0-未下载 1-已下载 |
| createTime | list | 1 | String | 文件创建时间 |

### 返回值示例(json)：

Success：{"code":"0","result":"success","list":[{"fid":"skdfj32412ajfasdfa","fileName":"1160223158203613","fileType":"txt","fileAmount":"1","fileSzie":"414","downloadFlag":"0","createTime":"2016-03-09 15:53"},{"fid":"4532agadg246ajfasdfa","fileName":"1160223158204567","fileType":"zip","fileAmount":"2","fileSzie":"32414","downloadFlag":"1","createTime":"2016-03-09 15:53"}]}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /list |
| 420 | 输入的json串不合法 | /list |
| 421 | 请求存在非法为空项，如：userid | /list |
| 520 | 服务器有异常发生 | /list |

## 查看文件（/show）

### 接口描述

查看单个文件。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | reqeust json | 1 | 是 | String | 用户唯一标识ID  不登录时，APP端给设备唯一标识ID；WEB端给anonymous；PC端如果有唯一标识ID，就给唯一标识ID，若没有也给anonymous |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准  不能为空 |
| fileType | request | 1 | 否 | String | 文件类型  1-识别文件  2-识别结果文件 |
| fid | reqeust json | 1 | 否 | String | 文件唯一标识ID |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | recogType、fileType、fid都不为空 |

### 请求示例(json)：

{**"userid":"**test2345**","recogType":"2","fileType":"1","fid":""}**

### 响应消息体(json)：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 | |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | string | 消息内容 |
| fid | response json | 1 | String | 文件唯一标识ID |
| fileName | response json | 1 | String | 文件名 |
| filePath | response json | 1 | String | 文件全路径 |
| fileFormat | response json | 1 | String | 文件格式  （识别文件jpg png zip……）  （识别结果文件txt pdf docx xls zip……） |
| fileSize | response json | 1 | Stirng | 文件总长度  （字节长度） |
| iszip | response json | 1 | Stirng | 是否打包文件 |
| fileAmount | response json | 1 | Stirng | 文件数量 |
| createTime | response json | 1 | String | 创建时间 |
| summary | response json | 1 | String | 文件摘要（前200个字）  注：识别结果文件返回此字段 |

### 返回值示例(json)：

成功：{"code":"0","result":"success","fid":"skdfj32412ajfasdfa","fileName":"1160223158203613","filePath":"/test2345/web/digital/out/1160223158203613.txt","fileFormat":"20","size":"14391","iszip":"false","fileAmount":"1","createTime":"2016年03月09日 15:53","summary":"测试测试测试2343156513461"}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /show |
| 420 | 输入的json串不合法 | /show |
| 421 | 请求存在非法为空项，如：userid、recogType | /show |
| 426 | 文件不存在 | /show |
| 520 | 服务器有异常发生 | /show |

## 下载文件（/download）

### 接口描述

下载一个或打包下载多个文件，针对所有文件有效。调用此接口之后，再调用文件校验接口，获取文件整体的校验值，以便验证下载文件的正确性。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | reqeust json | 1 | 是 | String | 用户唯一标识ID |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准  不能为空，为1时userid可为空 |
| fileType | request header | 1 | 否 | String | 文件类型  1-识别文件  2-识别结果文件 |
| fid | request json | 1 | 是 | String | 文件唯一标识ID  为空时，根据文件全路径下载文件 |
| filePath | request json | 1 | 是 | String | 文件全路径  为空时，根据文件ID下载文件  有值时，请使用URLEncoder.encode转码 |
| offset | reqeust json | 1 | 否 | String | 每次从offset开始读取  首次为0 |
| length | reqeust json | 1 | 否 | String | 每次读取下载文件的长度（字节长度） |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | recogType不为空 |
| 2 | fid和filePath至少一个不为空 |

### 请求示例(json)：

{"userid":"test2345","recogType":"2","fileType":"2","fid":"",**"**filePath**":"**/digital/output/测试图像\_result.txt**","offset":"580","length":"1024"}**

### 响应消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | | 描述 |
| content | response | 1 | | InputStream | 数据流 |

### 返回值示例(json)：

成功：获取数据流。

java示例代码，如下：

String input = "{\"userid\":\"zoe8626\",\"recogType\":\"2\",\"fileType\":\"2\",\"fid\":\"\"," +

"\"filePath\":\""+URLEncoder.encode(filePath, "UTF-8") +"\",\"offset\":\""+offset+"\"," +

"\"length\":\""+(long)perLen+"\"}";

// 创建默认的httpClient实例.

HttpClient httpClient = new DefaultHttpClient();

HttpResponse response;

try { // 模拟调用rest API下载文件接口

StringEntity entity = new StringEntity(input);

// 以post方式请求URL

HttpPost httpPost = new HttpPost(urlStr);

// 参数为json串形式

httpPost.addHeader("Content-Type", MediaType.APPLICATION\_JSON);

httpPost.setEntity(entity);

// 执行后获得响应数据

response = httpClient.execute(httpPost);

int statusCode = response.getStatusLine().getStatusCode();

if(statusCode != 200){

httpPost.abort();

throw new RuntimeException("HttpClient, response :" + response);

}

String contentDisposition = response.getHeaders("Content-Disposition")[0].toString();

String attachmentFileName = contentDisposition.substring(

contentDisposition.lastIndexOf("=")+2,

contentDisposition.lastIndexOf("#"));

// 解决中文文件名乱码问题

attachmentFileName = URLDecoder.decode(attachmentFileName, "UTF-8");

System.out.println("------attachmentFileName:"+attachmentFileName);

String size = contentDisposition.substring(

contentDisposition.lastIndexOf("#")+1, contentDisposition.length());

System.out.println("------size:"+size);

String downloadPath = "/temp/"+attachmentFileName;

if(offset == 0){ // 首次删除已存在文件

NfsUtil.deleteFile(downloadPath);

}

HttpEntity responseEntity = response.getEntity();

if(responseEntity != null){

if(responseEntity.isStreaming()){

//下载文件，断点续传，每次的内容追加写入文件

responseEntity.writeTo(new FileOutputStream(

new File("E:/data/temp/"+attachmentFileName), true));

}else{

String result = EntityUtils.toString(responseEntity, "utf-8");

EntityUtils.consume(responseEntity);

System.out.println("------result:" + result);

}

httpPost.abort();

}

//下次取的起始位置

offset += (long)perLen;

} catch (UnsupportedEncodingException e1) {

e1.printStackTrace();

} catch (ClientProtocolException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

失败：{" code":"520"," result":"System excption occured"}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /download |
| 420 | 输入的json串不合法 | /download |
| 421 | 请求存在非法为空项，如：userid、recogType | /download |
| 520 | 服务器有异常发生 | /download |

## 删除文件（/delete）

### 接口描述

删除一个或多个文件，针对所有文件有效。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=john

### 请求消息体：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | | 可为空 | 数据类型 | 描述 |
| userid | request json | 1 | | 否 | String | 用户名，注册时用的用户名 |
| fileType | request | 1 | 否 | | String | 文件类型  1-识别文件  2-识别结果文件 |
| fid | 无 | 1 | | 否 | string | 文件唯一标识ID  多个文件fid的集合，中间用逗号(,)分开，如下：  {12dsf11,as214df,as868d} |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、fileType、fid都不为空 |

### 请求示例(json)：

{**"**userid**":"**john**","** fileType **":"**2**","fid":"**12dsf11,as214df,as868d**"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |

### 返回值示例(json)：

成功：{**"** code**":"**0**","** result**":"OK"}**

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /delete |
| 420 | 输入的json串不合法 | /delete |
| 421 | 请求存在非法为空项，如：userid、fid | /delete |
| 520 | 服务器有异常发生 | /delete |

## 查询文件（/search）

### 接口描述

根据用户查询条件，进行全文检索。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=john

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 是 | String | 用户唯一标识ID |
| start | request | 1 | 否 | String | 起始位置 |
| pageSize | request | 1 | 否 | String | 每页最大记录数 |
| sort | request | 1 | 是 | String json | 默认按最后创建时间降序排列，获取到最新的。可扩展，  **{"create**\_time**"**:**"**desc**"}**降序  **{"create**\_time**"**:**"**asc**"}**升序 |
| fileType | request | 1 | 否 | String | 文件类型  1-识别文件  2-识别结果文件 |
| recogType | request | 1 | 是 | String | 识别类型  1-快速识别 2-人工精准  为空时，查询所有识别类型的识别结果文件列表。 |
| queryMap | request | 1 | 是 | String json | {“oid”:”0503”,”fid”:”113465768741”,”beginDate”:”2013-05-27”,”endDate”:”2013-05-27”,”month”:”2013-05”,”ilename”:”我的图片”}各查询条件之间为“与”的关系。  条件可不都给，beginDate和endDate可单独使用，也可配合使用，但不能和month一起使用。文件名和摘要为模糊查询 |
| oid | queryMap | 1 | 是 | string | 订单唯一标识ID |
| fid | queryMap | 1 | 是 | string | 文件唯一标识ID |
| beginDate | queryMap | 1 | 是 | string | 开始日期  （2015-02-12） |
| endDate | queryMap | 1 | 是 | string | 结束日期  （2015-12-01） |
| month | queryMap | 1 | 是 | string | 月份  （年月：2015-02） |
| fileName | queryMap | 1 | 是 | string | 文件名 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不为空 |
| 2 | 各查询条件之间为“与”的关系。 |

### 请求示例(json)：

{"userid":"tong3","start":"0","pageSize":"10","sort":{"create\_time":"desc"},"fileType":"2","recogType":"2","queryMap":{"beginDate":"","endDate":"","month":"201305","fileName":"我的图片"}}

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |
| list | response | 1 | string | 文件列表 |
| fid | list | 1 | String | 文件唯一标识ID |
| fileName | list | 1 | String | 文件名 |
| createTime | list | 1 | String | 创建时间 |

### 返回值示例(json)：

成功：{**"** code**":"**0**","** result**":"OK","list":[{"fid":"121asdf","fileName":"this is a test excerpt","createTime":"2013-03-24:12:12:12"}{"fid":"121asdf2","fileName":"this is a test excerpt2","createTime":"2013-03-24:12:12:12"}]**}

失败：{**"** code**":"**520**","** result**":"System excption occured","list":""**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /search |
| 420 | 输入的json串不合法 | /search |
| 421 | 请求存在非法为空项，如：userid | /search |
| 520 | 服务器有异常发生 | /search |

## 文件校验(/checknum)

### 接口描述

下载文件之后，调用此接口获取文件整体的校验值，和已下载文件的整体校验值比较，验证下载文件的正确性。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | 无 | 1 | 否 | String | 用户唯一标识ID |
| filePath | 无 | 1 | 否 | String | 文件全路径 |

### 请求体特别约束

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、filePath都不为空 |

### 请求示例

{"userid":"john","filePath":"/scanfile/testnew123.txt"}

### 响应消息体

| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容  成功为文件整体的校验值 |

### 返回值示例

Success：{**"**code**":"**0**","**result**":"**jsdfjalsdjfjsdfjalsdfas**"**}

Failure：{**"**code**":"**1000**","**result**":"Xxexcption occurred"**}

### 响应码列表

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /checksum |
| 420 | 输入的json串不合法 | /checksum |
| 421 | 请求存在非法为空项，如：userid | /checksum |
| 526 | 文件的校验值为空 | /checksum |

## 识别结果下载查询（/result/download/search）

### 接口描述

查看单个文件。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | reqeust json | 1 | 是 | String | 用户唯一标识ID  不登录时，APP端给设备唯一标识ID；WEB端给anonymous；PC端如果有唯一标识ID，就给唯一标识ID，若没有也给anonymous |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准  不能为空 |
| fileType | request | 1 | 否 | String | 文件类型  1-识别文件  2-识别结果文件 |
| downloadFlag | request | 1 | 否 | String | 是否下载  0-未下载 1-已下载 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | recogType、fileType都不为空 |

### 请求示例(json)：

{**"userid":"**test2345**","recogType":"2","fileType":"2"}**

### 响应消息体(json)：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 | |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | string | 消息内容 |
| fileAmount | response json | 1 | Integer | 未下载的结果文件数量 |

### 返回值示例(json)：

成功：{"code":"0","result":"success","fileAmount":"5"}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /result/download/search |
| 420 | 输入的json串不合法 | /result/download/search |
| 421 | 请求存在非法为空项，如：userid | /result/download/search |

## 获取识别结果类型（/get/result/type）

### 接口描述

获取识别结果的文件类型。

### 请求方法：

POST /URL

Header: 无

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| 无 | | | | | |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 无 | |

### 请求示例(json)：

无

### 响应消息体(json)：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | | 父字段 | 出现次数 | 数据类型 | 描述 | |
| code | | response json | 1 | Integer | 返回码 |
| result | | response json | 1 | string | 消息内容 |
| list | | response json | 1 | string | 识别结果类型列表 |
| resTypeName | | list | 1 | String | 识别结果类型名称  txt  pdf  docx  …… |
| resTypeValue | | list | 1 | integer | 识别结果类型值  0 – txt  1 – pdf  2 – docx  ……  （订单保存此值） |
| desc | | list | 1 | String | 识别结果类型描述  文本文档  可复制的PDF文档  Microsoft Word 文档  …… |
| isAvailable | | list | 1 | integer | 是否可用  0 – 不可用  1 – 可用 |
| resultFileType | | response json | | 1 | | string | txt |

多个以逗号分隔[,pdf……]62

### 返回值示例(json)：

成功：{"code":"0","result":"success","list":[{“resTypeName”:”txt”,"resTypeValue":”0”,”desc”:”文本文档”,”isAvailable”:”1”},{“resTypeName”:”pdf”,"resTypeValue":”1”,”desc”:”可复制的PDF文档”,”isAvailable”:”0”},{“resTypeName”:”docx”,"resTypeValue":”2”,”desc”:”Microsoft Word 文档”,”isAvailable”:”0”}]}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /result/download/search |

# *订单管理接口*（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/order）



## 价格评估（/evaluate）

### 接口描述

根据文件路径，下载识别文件进行价格估算。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | reqeust json | 1 | 否 | String | 用户唯一标识ID |
| fid | reqeust json | 1 | 否 | String | 文件唯一标识ID |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、fid不为空 |

### 请求示例(json)：

{**"**userid**":"**test2345**","fid":"25458dsferw23sf"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | String | 消息内容 |
| oid | response | 1 | String | 订单唯一标识ID |
| fileAmount | response | 1 | String | 文件数量 |
| wordsRange | response | 1 | String | 总字数 |
| accurateWords | response | 1 | String | 置信度为A的字数 |
| recogRate | response | 1 | String | 识别率 |
| recogAngle | response | 1 | String | 识别角度 |
| zoom | response | 1 | String | 放大倍数 |
| imgLevel | response | 1 | String | 图片质量级别  A、B、C、D、E |
| level | response | 1 | String | 优先级（数字越小优先级越高）  1-一级  2-二级  3-三级  …… |
| price | response | 1 | String | 价格 |
| finishTime | response | 1 | String | 预计完成时间 |
| waitTime | response | 1 | String | 等待时间 |
| contactId | response | 1 | String | 联系信息唯一标识ID |
| fullname | response | 1 | String | 联系人 |
| mobile | response | 1 | String | 联系电话 |
| errorFileName | response | 1 | String | 不能识别文件名  多个逗号拼接 |

### 返回值示例(json)：

成功：{**"**code**":"**0**","**result**":"OK","oid":"aic\_201605040940210014","fileAmount":"3","wordsRange":"200","** accurateWords **":"180","**recogRate**":"0.875426","**recogAngle**":"0","zoom":"1.4","imgLevel":"A","level":"1","price":"10","**finishTime**":"2016-05-18 17:45:25","waitTime":"2小时9分","contactId":"2","fullname":"13845789586","mobile":"13845789586",”**errorFileName**”:”a1.jpg,2副本.bmp”**}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /evaluate |
| 8002 | 图片样本不够清晰 | /evaluate |
| 420 | 输入的json串不合法 | /evaluate |
| 421 | 请求存在非法为空项，如：userid | /evaluate |
| 426 | 文件不存在 | /evaluate |
| 520 | 服务器有异常发生 | /evaluate |

## 评估进度（/evaluate/progress）

### 接口描述

根据文件唯一标识ID，获取评估进度。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | reqeust json | 1 | 否 | String | 用户唯一标识ID |
| fid | reqeust json | 1 | 否 | String | 文件唯一标识ID |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、fid不为空 |

### 请求示例(json)：

{**"**userid**":"**test2345**","fid":"25458dsferw23sf"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | String | 消息内容 |
| fid | response | 1 | String | 识别文件唯一标识ID |
| rateOfProgress | response | 1 | String | 进度（百分比） |

### 返回值示例(json)：

成功：{**"**code**":"**0**","**result**":"OK",”fid”:”25458dsferw23sf”,"**rateOfProgress**":"70%"**}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /evaluate/progress |
| 420 | 输入的json串不合法 | /evaluate/progress |
| 421 | 请求存在非法为空项，如：userid、**fid** | /evaluate/progress |
| 520 | 服务器有异常发生 | /evaluate/progress |

## 评估结果（/evaluate/result）

### 接口描述

根据文件唯一标识ID，获取评估结果。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | reqeust json | 1 | 否 | String | 用户唯一标识ID |
| fid | reqeust json | 1 | 否 | String | 文件唯一标识ID |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、fid不为空 |

### 请求示例(json)：

{**"**userid**":"**test2345**","fid":"25458dsferw23sf"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | String | 消息内容 |
| oid | response | 1 | String | 订单唯一标识ID |
| fileAmount | response | 1 | String | 文件数量 |
| wordsRange | response | 1 | String | 总字数 |
| accurateWords | response | 1 | String | 置信度为A的字数 |
| recogRate | response | 1 | String | 识别率 |
| recogAngle | response | 1 | String | 识别角度 |
| zoom | response | 1 | String | 放大倍数 |
| imgLevel | response | 1 | String | 图片质量级别  A、B、C、D、E |
| level | response | 1 | String | 优先级（数字越小优先级越高）  1-一级  2-二级  3-三级  …… |
| price | response | 1 | String | 价格 |
| finishTime | response | 1 | String | 预计完成时间 |
| waitTime | response | 1 | String | 等待时间 |
| contactId | response | 1 | String | 联系信息唯一标识ID |
| fullname | response | 1 | String | 联系人 |
| mobile | response | 1 | String | 联系电话 |
| errorFileName | response | 1 | String | 不能识别文件名  多个逗号拼接 |

### 返回值示例(json)：

成功：{**"**code**":"**0**","**result**":"OK","oid":"aic\_201605040940210014","fileAmount":"3","wordsRange":"200","** accurateWords **":"180","**recogRate**":"0.875426","**recogAngle**":"0","zoom":"1.4","imgLevel":"A","level":"1","price":"10","**finishTime**":"2016-05-18 17:45:25","waitTime":"2小时9分","contactId":"2","fullname":"13845789586","mobile":"13845789586",”**errorFileName**”:”a1.jpg,2副本.bmp”**}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /evaluate/result |
| 420 | 输入的json串不合法 | /evaluate/result |
| 421 | 请求存在非法为空项，如：userid、**fid** | /evaluate/result |
| 426 | 结果不存在 | /evaluate/result |
| 520 | 服务器有异常发生 | /evaluate/result |

## 添加订单（/add）

### 接口描述

用户登录后，可创建新订单。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | request | 1 | 否 | String | 用户唯一标识ID |
| oid | request | 1 | 是 | String | 订单唯一标识ID  不能为空，调用完7.1价格评估接口后会生成此ID |
| contactId | request | 1 | 是 | String | 联系信息唯一标识ID  不能为空 |
| inFid | request | 1 | 否 | String | 输入的识别文件唯一标识ID |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准 |
| wordsRange | request | 1 | 否 | String | 总字数 |
| accurateWords | response | 1 | 否 | String | 置信度为A的字数 |
| recogRate | response | 1 | 否 | String | 识别率 |
| recogAngle | response | 1 | 否 | String | 识别角度 |
| zoom | response | 1 | 否 | String | 识别放大倍数 |
| payType | request | 1 | 否 | String | 支付类型  1-第三方支付  2-银行（借记/信用）卡支付 |
| payWay | request | 1 | 否 | String | 支付方式  支付类型选择第三方支付，此值为：  11-支付宝  12-微信支付  13-百度钱包  ……  支付类型选择银行卡支付  21-招商银行  22-工商银行  23-建设银行  24-中国农业银行  …… |
| price | request | 1 | 否 | String | 支付金额 |
| waitTime | request | 1 | 否 | String | 等待时长（几小时几分钟）  使用 URLEncoder.encode转码 utf-8 |
| outputType | request | 1 | 否 | String | 输出文档类型  0-txt  1-pdf  2-docx  3-xls  ……  多种输出类型以逗号（,）分隔，如：0,1,2 |
| level | request | 1 | 否 | String | 优先级（数字越小优先级越高）  1-一级  2-二级  3-三级  …… |
| platformType | request | 1 | 否 | String | 平台类型  1-Web  2-PC  3-iOS  4-Android  …… |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不为空 |

### 请求示例(json)：

{"userid":"test2345","oid":"aic\_201605041750201254","inFid":"4654548798797","wordsRange":"5200"**,"** accurateWords **":"5180","**recogRate**":"0.875426","**recogAngle**":"0","zoom":"1.4"**,"recogType":"2","price":"10","payType":"1","payWay":"12","waitTime":"URLEncoder.encode('18分钟','utf-8')","outputType":"1,2","level":"1","platformType":"1"}

### 响应消息体：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |
| oid | response | 1 | string | 订单唯一标识ID |

### 返回值示例(json)：

Success：{**"**code**":"**0**","**result**":"** success**","oid":"3512355732323"**}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured","oid":""**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /add |
| 420 | 输入的json串不合法 | /add |
| 421 | 请求存在非法为空项，如：userid、oid、contactId | /add |
| 520 | 服务器有异常发生 | /add |

## 修改支付状态（/edit/pay/status）

### 接口描述

第三方支付反馈后调用此接口，修改支付状态，及订单状态

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | request | 1 | 否 | String | 用户唯一标识ID |
| oid | request | 1 | 否 | String | 订单唯一标识ID |
| status | request | 1 | 否 | String | 支付状态  1：支付成功  -1：支付失败 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不为空 |

### 请求示例(json)：

{"userid":"test2345","oid":"4654548798797","status":"1"}

### 响应消息体：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |

### 返回值示例(json)：

Success：{**"**code**":"**0**","**result**":"** success**"**}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /edit/pay/status |
| 420 | 输入的json串不合法 | /edit/pay/status |
| 421 | 请求存在非法为空项，如：userid | /edit/pay/status |
| 520 | 服务器有异常发生 | /edit/pay/status |

## 支付结果轮询（/pay/result/polling）

### 接口描述

用户支付后，需要等待第三方通知支付结果，轮询订单支付结果，查看用户是否支付成功。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | request | 1 | 否 | String | 用户唯一标识ID |
| oid | request | 1 | 否 | String | 订单唯一标识ID |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不为空 |

### 请求示例(json)：

{"userid":"test2345","oid":"4654548798797"}

### 响应消息体：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |

### 返回值示例(json)：

Success：{**"**code**":"**0**","**result**":"** success**"**}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /pay/result/polling |
| 420 | 输入的json串不合法 | /pay/result/polling |
| 421 | 请求存在非法为空项，如：userid、oid | /pay/result/polling |
| 433 | 待付款中 | /pay/result/polling |
| 520 | 服务器有异常发生 | /pay/result/polling |

## 订单列表（/list）

### 接口描述

获取文件列表。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 否 | String | 用户唯一标识ID |
| start | request | 1 | 否 | String | 起始位置 |
| pageSize | request | 1 | 否 | String | 每页最大记录数 |
| sort | request | 1 | 是 | String json | 默认按最后创建时间降序排列，获取到最新的。可扩展，  **{"create**\_time**"**:**"**desc**"}**降序  **{"create**\_time**"**:**"**asc**"}**升序 |
| status | request | 1 | 是 | string | 订单状态  1-待支付  2-待加工（已支付，任务队列中等待）  3-处理中（已加工，等待加工结果）  4-取消  5-完成  此值为空，查询全部订单。  （查询条件） |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不能为空 |

### 请求示例(json)：

{"userid":"test2345","start":"0","pageSize":"10","sort":{"**create**\_time":"desc"},"status":"","recogType":"2"}

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | string | 消息内容 |
| list | response json | 1 | json array | 订单列表 |
| oid | list | 1 | String | 订单唯一标识ID |
| fullname | list | 1 | String | 订单联系人 |
| createTime | list | 1 | String | 订单创建时间 |
| status | list | 1 | String | 订单状态  1-待支付  2-待加工（已支付，任务队列中等待）  3-处理中（已加工，等待加工结果）  4-取消  5-完成  不给值，查询全部订单。 |
| fileName | list | 1 | String | 识别文件名称 |
| price | list | 1 | String | 价格 |
| recordCount | response json | 1 | String | 订单总数 |

### 返回值示例(json)：

Success：{"code":"0","result":"success","list":[{"oid":"65894521584774","fullname":"张三","createTime":"2016年03月09日 15:53","status":"","fileName":"ceshi","price":"10.20"},{"oid":"658945215847569","fullname":"张三","createTime":"2016年03月09日 15:53","status":"","fileName":"ceshi","price":"10.20"}],”recordCount”:”10”}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /list |
| 420 | 输入的json串不合法 | /list |
| 421 | 请求存在非法为空项，如：userid | /list |
| 520 | 服务器有异常发生 | /list |

## 查看订单（/show）

### 接口描述

查看订单详情。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | reqeust json | 1 | 否 | String | 用户唯一标识ID |
| oid | reqeust json | 1 | 否 | String | 订单唯一标识ID |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、oid都不为空 |

### 请求示例(json)：

{**"userid":"**test2345**","oid":"6556589845812"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | String | 消息内容 |
| contactId | response json | 1 | String | 联系信息唯一标识ID |
| fullname | response json | 1 | String | 用户姓名 |
| mobile | response json | 1 | String | 手机号 |
| oid | response json | 1 | String | 订单唯一标识ID |
| wordsRange | response json | 1 | String | 总字数 |
| payType | response json | 1 | String | 支付类型  1-第三方支付  2-银行（借记/信用）卡支付  返回中文描述，直接显示即可 |
| payWay | response json | 1 | String | 支付方式  支付类型选择第三方支付，此值为：  11-支付宝  12-微信支付  13-百度钱包  ……  支付类型选择银行卡支付  21-招商银行  22-工商银行  23-建设银行  24-中国农业银行  ……  返回中文描述，直接显示即可 |
| price | response json | 1 | String | 支付金额  返回多少（元），直接显示即可 |
| waitTime | response json | 1 | String | 等待时长  返回几小时几分钟几秒，直接显示即可 |
| finishTime | response json | 1 | String | 完成时间 |
| createTime | response json | 1 | String | 订单创建时间 |
| inFid | response json | 1 | String | 识别文件唯一标识ID  上传的图片文件的ID |
| fileName | response json | 1 | String | 识别文件名称 |
| fileAmount | response json | 1 | String | 识别文件张数 |
| outFid | response json | 1 | String | 结果文件唯一标识ID  可为空 |
| status | response json | 1 | String | 订单状态  1-待支付  2-待加工（已支付，任务队列中等待）  3-处理中（已加工，等待加工结果）  4-取消  5-完成  为空时，查询全部订单。  返回中文描述，直接显示即可 |

### 返回值示例(json)：

成功：{"code":"0","result":"success","contactId":"2","fullname":"张三","mobile":"13111265326","oid":"65894521584774","wordsRange":"5150-5200","payType":"第三方支付","payWay":"支付宝","price":"10","waitTime":"30","finishTime":"2016年03月09日 18:53","createTime":"2016年03月09日 15:53","inFid":"dsalfj324kj","fileName":"测试图片","fileAmount":"1","outFid":"","status":"待支付"}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /show |
| 420 | 输入的json串不合法 | /show |
| 421 | 请求存在非法为空项，如：userid、oid | /show |
| 426 | 订单不存在 | /show |
| 520 | 服务器有异常发生 | /show |

## 取消订单（/cancel）

### 接口描述

修改订单状态为4-取消。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=john

### 请求消息体：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | | 可为空 | 数据类型 | 描述 |
| userid | request json | 1 | | 否 | String | 用户唯一标识ID |
| oid | request json | 1 | 否 | | String | 订单唯一标识ID  多个订单oid的集合，中间用逗号(,)分开，如下：  6556525895,6556525854,6556525214 |
| status | request json | 1 | | 否 | string | 订单状态  4-取消 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、oid都不为空 |

### 请求示例(json)：

{**"**userid**":"**john**","oid":"**6556525895,6556525854,6556525214**","status":"**4**",}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |

### 返回值示例(json)：

成功：{**"** code**":"**0**","** result**":"OK"}**

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /cancel |
| 420 | 输入的json串不合法 | /cancel |
| 421 | 请求存在非法为空项，如：userid、oid | /cancel |
| 520 | 服务器有异常发生 | /cancel |

## 删除订单（/delete）

### 接口描述

将订单逻辑删除。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=john

### 请求消息体：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | | 可为空 | 数据类型 | 描述 |
| userid | request json | 1 | | 否 | String | 用户唯一标识ID |
| oid | request json | 1 | 否 | | String | 订单唯一标识ID  多个订单oid的集合，中间用逗号(,)分开，如下：  6556525895,6556525854,6556525214 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid、oid都不为空 |

### 请求示例(json)：

{**"**userid**":"**john**","oid":"**6556525895,6556525854,6556525214**"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |

### 返回值示例(json)：

成功：{**"** code**":"**0**","** result**":"OK"}**

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /delete |
| 420 | 输入的json串不合法 | /delete |
| 421 | 请求存在非法为空项，如：userid、oid | /delete |
| 520 | 服务器有异常发生 | /delete |

## 查询订单（/search）

### 接口描述

根据用户查询条件，进行全文检索。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=john

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 是 | String | 用户唯一标识ID |
| start | request | 1 | 否 | String | 起始位置 |
| pageSize | request | 1 | 否 | String | 每页最大记录数 |
| sort | request | 1 | 是 | String json | 默认按最后创建时间降序排列，获取到最新的。可扩展，  **{"create**\_time**"**:**"**desc**"}**降序  **{"create**\_time**"**:**"**asc**"}**升序 |
| status | request | 1 | 是 | string | 订单状态  1-待支付  2-待加工（已支付，任务队列中等待）  3-处理中（已加工，等待加工结果）  4-取消  5-完成  为空时，查询全部订单。  （查询条件） |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准 |
| queryMap | request | 1 | 是 | String json | {"oid":"644548494946","beginDate":"2013-05-27","endDate":"2013-05-27","month":"2013-05"} 各查询条件之间为“与”的关系。  条件可不都给，beginDate和endDate可单独使用，也可配合使用，但不能和month一起使用。 |
| oid | queryMap | 1 | 是 | string | 订单唯一标识ID |
| beginDate | queryMap | 1 | 是 | string | 开始日期  （2015-02-12） |
| endDate | queryMap | 1 | 是 | string | 结束日期  （2015-12-01） |
| month | queryMap | 1 | 是 | string | 月份  （年月：2015-02） |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid不为空 |
| 2 | 各查询条件之间为“与”的关系。 |

### 请求示例(json)：

{"userid":"tong3","start":"0","pageSize":"10","sort":{"create\_time":"desc"},"status":"1","recogType":"2","queryMap":{"oid":"25655885668865","beginDate":"","endDate":"","month":"201305"}}

### 响应消息体(json)：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 | |
| code | response json | 1 | Integer | 返回码 | |
| result | response json | 1 | String | 消息内容 | |
| list | response json | 1 | json array | 订单列表 | |
| oid | list | 1 | String | 订单唯一标识ID | |
| createTime | list | 1 | String | 订单创建时间 | |
| status | list | 1 | String | 订单状态  1-待支付  2-待加工（已支付，任务队列中等待）  3-处理中（已加工，等待加工结果）  4-取消  5-完成  不给值，查询全部订单。 | |
| fileName | list | 1 | String | 识别文件名称 |
| price | list | 1 | String | 价格 |

### 返回值示例(json)：

成功：{"code":"0","result":"success","list":[{"oid":"65894521584774","createTime":"2016年03月09日 15:53","status":"","fileName":"ceshi","price":"10.20"},{"oid":"658945215847569","createTime":"2016年03月09日 15:53","status":"","fileName":"ceshi","price":"10.20"}]}

失败：{**"** code**":"**520**","** result**":"System excption occured","list":""**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /search |
| 420 | 输入的json串不合法 | /search |
| 421 | 请求存在非法为空项，如：userid | /search |
| 520 | 服务器有异常发生 | /search |

## 获取收费评估标准（/get/fee/standard）

### 接口描述

获取收费评估标准列表。

### 请求方法：

POST /URL

Header: 无

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| 无 | | | | | |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 无 | |

### 请求示例(json)：

无

### 响应消息体(json)：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | | 父字段 | 出现次数 | 数据类型 | 描述 | |
| code | | response json | 1 | Integer | 返回码 |
| result | | response json | 1 | string | 消息内容 |
| list | | response json | 1 | string | 收费评估标准列表 |
| imgLevel | | list | 1 | String | 图片质量级别  A级  B级  C级  D级  E级 |
| levelRange | | list | 1 | String | 图片质量评估范围  A –【95-100】%  B –【90-95】%  C –【85-90】%  D –【80-85】%  E –【70-80】% |
| txtPrice | | list | 1 | String | TXT-单价/千字  A –25元  B –28元  C –30元  D –35元  E –40元 |
| txtTime | | list | 1 | String | TXT-时间/千字  A – 20分钟  B – 25分钟  C – 30分钟  D – 40分钟  E – 50分钟 |
| resultFileType | | response json | | 1 | | string | txt |

多个以逗号分隔[,pdf……]62

### 返回值示例(json)：

成功：{"code":"0","result":"success","list":[{“imgLevel”:”A”,"levelRange":”【95-100】%”,”txtPrice”:”25元”,”txtTime”:”20分钟”},{“imgLevel”:”B”,"levelRange":”【90-95】%”,”txtPrice”:”28元”,”txtTime”:”25分钟”}]}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /get/fee/standard |

# *支付宝支付-APP端调用接口*（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/alipay）

## 签名接口（/sign）

### 接口描述

根据订单信息，生成签名。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 否 | String | 用户唯一标识 |
| orderInfo | reqeust json | 1 | 否 | String | 订单信息 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | userid 、orderInfo不能为空 |

### 请求示例(json)：

{"userid":"test2345",**"**orderInfo**":"<订单信息拼接字符串>"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | String | 消息内容 |
| sign | response | 1 | String | 签名 |

### 返回值示例(json)：

成功：{**"**code**":"**0**","**result**":"OK","sign":"pqirqmalsfmdlsakfjjfk34"**}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /sign |
| 420 | 输入的json串不合法 | /sign |
| 520 | 服务器有异常发生 | /sign |

## 通知接口（/notify）

### 接口描述

获取支付宝支付结果通知，由支付宝回调此接口。

## 查询接口（/query）

### 接口描述

查询订单支付结果。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 否 | String | 用户唯一标识 |
| oid | reqeust json | 1 | 否 | String | 订单唯一标识 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | 请求字段都不能为空 |

### 请求示例(json)：

{"userid":"test2345",**"oid":"aic\_201604201600230987"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | String | 消息内容 |

### 返回值示例(json)：

成功：{**"**code**":"**0**","**result**":"OK","data":"0"**}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /query |
| 420 | 输入的json串不合法 | /query |
| 421 | 请求存在非法为空项，如：userid | /query |
| 433 | 等待支付 | /query |
| 520 | 服务器有异常发生 | /query |

# *微信支付-APP端调用接口*（http://rc.hwyun.com9090/rws-cloud/rt/ws/v1/wxpay）

## 统一下单接口（/unifiedorder）

### 接口描述

根据订单信息，调用微信统一下单接口。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 否 | String | 用户唯一标识 |
| oid | reqeust json | 1 | 否 | String | 订单唯一标识 |
| productId | reqeust json | 1 | 否 | String | 识别类型  1-快速识别  2-精准人工  目前均为2 |
| body | reqeust json | 1 | 否 | String | 产品名称  暂定：汉王识文-精准人工识别  使用URLEncoder.encode("汉王识文-精准人工识别", "UTF-8") 转码 |
| price | reqeust json | 1 | 否 | String | 价格 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | 请求字段都不能为空 |

### 请求示例(json)：

{"userid":"test2345",**"oid":"aic\_201604201600230987","productId":"2","body":"**URLEncoder.encode("汉王识文-精准人工识别", "UTF-8") 转码后的值**","price":"15"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | String | 消息内容 |
| returnCode | response | 1 | String | 微信返回状态码 |
| returnMsg | response | 1 | String | 微信返回信息 |
| appId | response | 1 | String | 应用ID |
| nonceStr | response | 1 | String | 随机字符串 |
| sign | response | 1 | String | 签名，统一下单的签名 |
| resultCode | response | 1 | String | 业务结果 |
| prepayId | response | 1 | String | 预支付交易会话ID |
| tradeType | response | 1 | String | 交易类型 |
| partnerId | response | 1 | String | 微信支付分配的商户号 |
| timeStamp | response | 1 | String | 时间戳 |
| packageName | response | 1 | String | 扩展字段，暂填写固定值Sign=WXPay |
| reSign | response | 1 | String | 签名，服务端生成的签名 |

### 返回值示例(json)：

成功：**{"code":"0","result":"OK","returnCode":"SUCCESS","returnMsg":"","appId":"wx021fab5878ea9288","nonceStr":"5K8264ILTKCH16CQ2502SI8ZNMTM67VS","sign":"C380BEC2BFD727A4B6845133519F3AD6","resultCode":"SUCCESS","prepayId":"","tradeType":"APP","partnerId":"1330963101","timeStamp":"1412000000","packageName":"**Sign=WXPay**","reSign":"CDKJWOE34234DSJFS"}**

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /unifiedorder |
| 420 | 输入的json串不合法 | /unifiedorder |
| 520 | 服务器有异常发生 | /unifiedorder |

## 通知接口（/notify）

### 接口描述

获取微信支付结果通知，由微信回调此接口。

## 查询接口（/query）

### 接口描述

根据订单信息，调用微信统一下单接口。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| userid | request | 1 | 否 | String | 用户唯一标识 |
| oid | reqeust json | 1 | 否 | String | 订单唯一标识 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | 请求字段都不能为空 |

### 请求示例(json)：

{"userid":"test2345",**"oid":"aic\_201604201600230987"}**

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | String | 消息内容 |
| data | response | 1 | String | 0 支付成功 |

### 返回值示例(json)：

成功：{**"**code**":"**0**","**result**":"OK","data":"0"**}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /query |
| 2 | 已经支付 | /query |
| 420 | 输入的json串不合法 | /query |
| 520 | 服务器有异常发生 | /query |

# *支付-PC端调用接口*（http://rc.hwyun.com:9090/aic-cloud/pcorder）

## 支付接口（/pay）

### 接口描述

根据订单信息，调用Web支付页面。

### 请求方法：

POST /URL

Header: token=ksdkmzxmcvqwuier&9124kjsadf

userid=johe

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| userid | request | 1 | 否 | String | 用户唯一标识ID |
| oid | request | 1 | 是 | String | 订单唯一标识ID  不能为空，调用完7.1价格评估接口后会生成此ID |
| contactId | request | 1 | 是 | String | 联系信息唯一标识ID  不能为空 |
| inFid | request | 1 | 否 | String | 输入的识别文件唯一标识ID |
| recogType | request | 1 | 否 | String | 识别类型  1-快速识别 2-人工精准 |
| wordsRange | request | 1 | 否 | String | 总字数 |
| accurateWords | response | 1 | 否 | String | 置信度为A的字数 |
| recogRate | response | 1 | 否 | String | 识别率 |
| recogAngle | response | 1 | 否 | String | 识别角度 |
| zoom | response | 1 | 否 | String | 识别放大倍数 |
| price | request | 1 | 否 | String | 支付金额 |
| waitTime | request | 1 | 否 | String | 等待时长（几小时几分钟）  使用 URLEncoder.encode转码 utf-8 |
| outputType | request | 1 | 否 | String | 输出文档类型  0-txt  1-pdf  2-docx  3-xls  ……  多种输出类型以逗号（,）分隔，如：0,1,2 |
| level | request | 1 | 否 | String | 优先级（数字越小优先级越高）  1-一级  2-二级  3-三级  …… |
| platformType | request | 1 | 否 | String | 平台类型  1-Web  2-PC  3-iOS  4-Android  …… |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | 请求字段都不能为空 |

### 请求示例(multipart/form-data)：

Java示例代码，如下：

@Test

public void upload(){

String result = null;

// URL

String url = "http://localhost:8080/aic-cloud/pcorder/pay";

// 上传参数

Map<String, String> params = new HashMap<String, String>();

params.put("userId", userId);

params.put("oid", "aic\_201605201143251258");

params.put("price", "10");

……

try {

result = doPost(params, url);

} catch (Exception e) {

e.printStackTrace();

}

}

private String doPost(Map<String, String> params, String urlStr) throws Exception {

String result = null;

try {

MultipartEntityBuilder mEntityBuilder = MultipartEntityBuilder.create();

mEntityBuilder.setBoundary("----------ThIs\_Is\_tHe\_bouNdaRY\_$");

mEntityBuilder.setMode(HttpMultipartMode.BROWSER\_COMPATIBLE);

mEntityBuilder.setCharset(Charset.defaultCharset());

// 参数准备

if (params != null) {

for (Entry<String, String> e : params.entrySet()) {

mEntityBuilder.addTextBody(e.getKey(), e.getValue());

}

}

HttpPost httpPost = new HttpPost(urlStr);

httpPost.addHeader("Content-Type", MediaType.MULTIPART\_FORM\_DATA

+ "; boundary=----------ThIs\_Is\_tHe\_bouNdaRY\_$");

// 设置参数

httpPost.setEntity(mEntityBuilder.build());

// 创建默认的httpClient实例.

HttpClient httpClient = new DefaultHttpClient();

HttpResponse response = httpClient.execute(httpPost);

int statusCode = response.getStatusLine().getStatusCode();

if (statusCode != 200) {

httpPost.abort();

throw new RuntimeException("HttpClient, response :" + response);

}

HttpEntity entity = response.getEntity();

if (entity != null) { // 返回JSON结果

result = EntityUtils.toString(entity, "utf-8");

httpPost.abort();

}

EntityUtils.consume(entity);

} catch (Exception e) {

e.printStackTrace();

}

return result;

}

### 响应消息体：

跳转页面

### 返回值示例：

成功：跳转到Web端支付页面

失败：跳转到Web端错误页面

### 响应码列表：

无

# *数字加工接口*（http://rc.hwyun.com:9090/rws-cloud/rt/ws/v1/digital/processing）



## 任务轮询（/task/polling）

### 接口描述

用户上传图片，验证后加入任务等待队列中。调用此接口从等待队列中查询任务，如果有任务返回任务信息。

### 请求方法：

POST /URL

Header 无要求

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| 无 | | | | | |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | 无 |

### 请求示例(json)：

无

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | string | 消息内容 |
| oid | response json | 1 | String | 订单唯一标识ID |
| outputMode | response json | 1 | String | 输出方式  1-一图一文  2-多图一文 |
| fileFormat | response json | 1 | String | 输出文档类型  0-txt （先做）  1-pdf （先做）  2-docx  3-xls  ……  多种输出类型以逗号（,）分隔，如：0,1,2 |
| language | response json | 1 | String | 识别语言  0-简体中文 （先做）  1-繁体中文  2-古文  …… |
| level | response json | 1 | String | 优先级（数字越小优先级越高）  1-一级  2-二级  3-三级  …… |
| size | response json | 1 | String | 文件总长度（字节长度） |
| iszip | response json | 1 | String | 是否为zip包（true | false） |
| fileAmount | response json | 1 | String | 图片数量 |
| checksum | response json | 1 | String | 校验值（下载文件整体校验）  SHA1 (Base64(data)) |

### 返回值示例(json)：

Success：{"code":"0","result":"success","oid":"95841248414524","outputMode":"1","fileFormat":"0","language":"0","level":"1","size":"437178","checksum":"16BA2E74B4CF823F58371B3A1A5EBD49","iszip":"false","fileAmount":"1"}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | **/**task/polling |
| 520 | 服务器有异常发生 | **/**task/polling |

## 下载文件（/file/download）

### 接口描述

下载一个或打包下载多个文件，针对所有文件有效。

### 请求方法：

POST /URL

Header: 无要求

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| oid | reqeust json | 1 | 是 | string | 订单唯一标识ID |
| filePath | request json | 1 | 是 | String | 文件全路径 |
| offset | reqeust json | 1 | 否 | string | 每次从offset开始读取  首次为0 |
| length | reqeust json | 1 | 否 | string | 每次读取下载文件的长度（字节长度） |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | oid不为空 |

### 请求示例(json)：

{**"oid":"","**filePath**":"**/test23456/8542156325985/digital/input/655625358958.jpg**","offset":"580","length":"1024"}**

### 响应消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | | 描述 |
| content | response | 1 | | InputStream | 数据流 |

### 返回值示例：

成功：获取数据流。

java示例代码，如下：

String input = "{\"filePath\":\""+path+"\",\"offset\":\""+offset+"\"," +

"\"length\":\""+(long)perLen+"\"}";

// 创建默认的httpClient实例.

HttpClient httpClient = new DefaultHttpClient();

HttpResponse response;

try { // 模拟调用rest API下载文件接口

StringEntity entity = new StringEntity(input);

// 以post方式请求URL

HttpPost httpPost = new HttpPost(urlStr);

// 参数为json串形式

httpPost.addHeader("Content-Type", MediaType.APPLICATION\_JSON);

httpPost.setEntity(entity);

// 执行后获得响应数据

response = httpClient.execute(httpPost);

int statusCode = response.getStatusLine().getStatusCode();

if(statusCode != 200){

httpPost.abort();

throw new RuntimeException("HttpClient, response :" + response);

}

String contentDisposition = response.getHeaders("Content-Disposition")[0].toString();

// 获得附件文件名称

String attachmentFileName = contentDisposition.substring(

contentDisposition.lastIndexOf("=")+2, contentDisposition.length()-1);

// 解决中文文件名乱码问题

attachmentFileName = URLDecoder.decode(attachmentFileName, "UTF-8");

String downloadPath = "/temp/"+attachmentFileName;

if(offset == 0){ // 首次删除已存在文件

NfsUtil.deleteFile(downloadPath);

}

HttpEntity responseEntity = response.getEntity();

if(responseEntity != null){

if(responseEntity.isStreaming()){

//下载文件，断点续传，每次的内容追加写入文件

responseEntity.writeTo(new FileOutputStream(

new File("E:/data/temp/"+attachmentFileName), true));

}else{ //失败JSON结果

String result = EntityUtils.toString(responseEntity, "utf-8");

EntityUtils.consume(responseEntity);

System.out.println("------result:" + result);

}

httpPost.abort();

}

//下次取的起始位置

offset += (long)perLen;

} catch (UnsupportedEncodingException e1) {

e1.printStackTrace();

} catch (ClientProtocolException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

失败：{**"** code**":"**520**","** result**":"System excption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /file/download |
| 420 | 输入的json串不合法 | /file/download |
| 520 | 服务器有异常发生 | /file/download |

## 修改任务状态（/task/status/edit）

### 接口描述

下载文件成功后，调用此接口，修改任务和订单的状态为处理中。

### 请求方法：

POST /URL

Header 无要求

### 请求消息体：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| oid | reqeust json | 1 | 否 | string | 订单唯一标识ID |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | oid不能为空 |

### 请求示例(json)：

{"oid":"231654647987"}

### 响应消息体(json)：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response json | 1 | Integer | 返回码 |
| result | response json | 1 | string | 消息内容 |

### 返回值示例(json)：

Success：{"code":"0","result":"success"}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | **/**task/status/edit |
| 520 | 服务器有异常发生 | **/**task/status/edit |

## 上传文件（/file/upload）

### 接口描述

支持多种文件的传输，存储到汉王云盘

### 请求方法：

POST /URL

Header: 无要求

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| oid | request header | 1 | 否 | String | 订单唯一标识ID |
| size | request header | 1 | 否 | String | 文件总长度 |
| length | request header | 1 | 否 | String | 每次上传数据的长度（字节长度）  注：最后一次是文件总长度的剩余长度 |
| offset | request header | 1 | 否 | String | 每次写入的offset  第一次为0 |
| checksum | request header | 1 | 否 | String | 校验值（每次上传数据的校验）  SHA1 (Base64(data)) |
| fileFormat | request header | 1 | 否 | String | 文件格式  txt  pdf  docx  xls  ……  每次一种格式 |
| iszip | request header | 1 | 否 | String | 是否为zip包（true | false） |
| fileAmount | request header | 1 | 否 | String | 文件数量 |
| fileName | request header | 1 | 否 | String | 文件名称  不带后缀，使用 URLEncoder.encode转码 |
| data | request header | 1 | 否 | byte[] | 字节数组 |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | oid不为空 |

### 请求示例(multipart/form-data)：

Java模拟调用rest 上传API示例代码，如下：

@Test

public void upload(){

String result = null;

// URL

String url = "http://localhost:8080/rws-cloud/rt/ws/v1/file/upload";

File file = new File("E:/160202.txt");

// 上传参数

Map<String, String> params = new HashMap<String, String>();

// 订单ID

params.put("oid", "95841248414524");

// 文件总长度

params.put("size", file.length() + "");

params.put("fileName", URLEncoder.encode("160202", "UTF-8"));

params.put("fileFormat", "txt");

params.put("fileAmount", "1");

fileName

InputStream inputStream = null;

String offset = "0";

try {

inputStream = new FileInputStream(file);

int temLength = 1024;

byte[] temp = new byte[temLength];

// 上传数据

byte[] data = null;

// 是否为打包文件

params.put("iszip", "false");

for(int len = inputStream.read(temp); len != -1; len = inputStream.read(temp)){

String tempData = "";

if(len < temLength){

byte[] tempArray = new byte[len];

for(int i = 0; i < tempArray.length; i ++){

tempArray[i] = temp[i];

}

tempData = Base64.encode(tempArray);

data = tempArray;

// 每次上传数据的长度

params.put("length", String.valueOf(len));

}else{

tempData = Base64.encode(temp);

data = temp;

// 每次上传数据的长度

params.put("length", String.valueOf(temLength));

}

String checksum = SHA1Util.encodeBySHA(tempData);

// 每次上传数据的偏移量

params.put("offset", offset);

// 每次上传数据的校验和

params.put("checksum", checksum);

result = doPost(params, data, url);

UploadResponse responseBean = JsonUtil.getObjectFromJson(result, UploadResponse.class);

offset = responseBean.getOffset();

}

} catch (Exception e) {

e.printStackTrace();

}

}

private String doPost(Map<String, String> params, byte[] data, String urlStr) throws Exception {

String result = null;

try {

MultipartEntityBuilder mEntityBuilder = MultipartEntityBuilder.create();

// 每次上传的数据

ByteArrayBody dataBody = new ByteArrayBody(data,

ContentType.MULTIPART\_FORM\_DATA, "uploadTemp");

mEntityBuilder.addPart("file", dataBody);

mEntityBuilder.setBoundary("----------ThIs\_Is\_tHe\_bouNdaRY\_$");

mEntityBuilder.setMode(HttpMultipartMode.BROWSER\_COMPATIBLE);

mEntityBuilder.setCharset(Charset.defaultCharset());

// 参数准备

if (params != null) {

for (Entry<String, String> e : params.entrySet()) {

mEntityBuilder.addTextBody(e.getKey(), e.getValue());

}

}

HttpPost httpPost = new HttpPost(urlStr);

httpPost.addHeader("Content-Type", MediaType.MULTIPART\_FORM\_DATA

+ "; boundary=----------ThIs\_Is\_tHe\_bouNdaRY\_$");

// 设置参数

httpPost.setEntity(mEntityBuilder.build());

// 创建默认的httpClient实例.

HttpClient httpClient = new DefaultHttpClient();

HttpResponse response = httpClient.execute(httpPost);

int statusCode = response.getStatusLine().getStatusCode();

if (statusCode != 200) {

httpPost.abort();

throw new RuntimeException("HttpClient, response :" + response);

}

HttpEntity entity = response.getEntity();

if (entity != null) { // 返回JSON结果

result = EntityUtils.toString(entity, "utf-8");

httpPost.abort();

}

EntityUtils.consume(entity);

} catch (Exception e) {

e.printStackTrace();

}

return result;

}

### 响应消息体：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |
| fid | response | 1 | string | 文件唯一标识ID |
| offset | response | 1 | string | 文件成功上传偏移值。单位：字节 |

### 返回值示例(json)：

Success：{**"**code**":"**0**","**result**":"** success**","fid":"sdkfkssdkfks32323","offset":"1024"**}

Failure：{**"**code**":"**1000**","**result**":"XXexcption occured","oid":"","offset":""**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | /file/upload |
| 520 | 服务器有异常发生 | /file/upload |
| 524 | Checksum校验失败 | /file/upload |

## 完成（/task/finish）

### 接口描述

任务已完成。

### 请求方法：

POST /URL

Header: 无要求

### 请求消息体：

| 字段 | 父字段 | 出现次数 | 可为空 | 数据类型 | 描述 |
| --- | --- | --- | --- | --- | --- |
| oid | request json | 1 | 否 | String | 订单唯一标识ID |
| size | request json | 1 | 否 | String | 文件总长度 |
| checksum | request json | 1 | 是 | String | 校验值（文件整体校验）  SHA1 (Base64(data)) |

### 请求体特别约束：

|  |  |
| --- | --- |
| 编号 | 描述 |
| 1 | oid不为空 |

### 请求示例(json)：

{"oid":"3462sadfa34262","size":"1453","checksum":"06c09ae14ba134a3eaf8063c9903b682"}

### 响应消息体：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 父字段 | 出现次数 | 数据类型 | 描述 |
| code | response | 1 | Integer | 返回码 |
| result | response | 1 | string | 消息内容 |

### 返回值示例(json)：

Success：{**"**code**":"**0**","**result**":"** success**"**}

Failure：{**"**code**":"**524**","**result**":"checksum value are not equal"**}

### 响应码列表：

|  |  |  |
| --- | --- | --- |
| 代码 | 描述 | 模块 |
| 0 | 成功 | **/**task/finish |
| 420 | 输入的json串不合法 | **/**task/finish |
| 520 | 服务器有异常发生 | **/**task/finish |
| 524 | Checksum校验失败 | **/**task/finish |