

---

# OCR 文档

---

2017-07-03



**百度云**  
cloud.baidu.com



# 目录

|       |                |    |
|-------|----------------|----|
| 1     | API参考          | 1  |
| 1.1   | 简介             | 1  |
| 1.1.1 | 接口能力           | 1  |
| 1.1.2 | 请求格式           | 2  |
| 1.1.3 | 返回格式           | 2  |
| 1.1.4 | 请求限制           | 2  |
| 1.2   | 调用方式           | 3  |
| 1.2.1 | 调用方式一          | 3  |
| 1.2.2 | 调用方式二          | 3  |
| 1.3   | 通用文字识别         | 4  |
| 1.3.1 | 接口描述           | 4  |
| 1.3.2 | 请求说明           | 4  |
| 1.3.3 | 返回说明           | 6  |
| 1.4   | 通用文字识别（含位置信息版） | 7  |
| 1.4.1 | 接口描述           | 7  |
| 1.4.2 | 请求说明           | 7  |
| 1.4.3 | 返回说明           | 9  |
| 1.5   | 通用文字识别（含生僻字版）  | 12 |
| 1.5.1 | 接口描述           | 12 |
| 1.5.2 | 请求说明           | 12 |
| 1.5.3 | 返回说明           | 14 |

|        |                |    |
|--------|----------------|----|
| 1.6    | 网络图片文字识别       | 15 |
| 1.6.1  | 接口描述           | 15 |
| 1.6.2  | 请求说明           | 15 |
| 1.6.3  | 返回说明           | 17 |
| 1.7    | 身份证识别          | 18 |
| 1.7.1  | 接口描述           | 18 |
| 1.7.2  | 请求说明           | 18 |
| 1.7.3  | 返回说明           | 19 |
| 1.8    | 银行卡识别          | 21 |
| 1.8.1  | 接口描述           | 21 |
| 1.8.2  | 请求说明           | 21 |
| 1.8.3  | 返回说明           | 22 |
| 1.9    | 驾驶证识别          | 23 |
| 1.9.1  | 接口描述           | 23 |
| 1.9.2  | 请求说明           | 23 |
| 1.9.3  | 返回说明           | 24 |
| 1.10   | 行驶证识别          | 26 |
| 1.10.1 | 接口描述           | 26 |
| 1.10.2 | 请求说明           | 26 |
| 1.10.3 | 返回说明           | 27 |
| 1.11   | 表格文字识别         | 29 |
| 1.11.1 | 接口描述           | 29 |
| 1.11.2 | 表格文字识别——提交请求接口 | 29 |
|        | 请求说明           | 29 |
|        | 返回说明           | 30 |
| 1.11.3 | 表格文字识别——获取结果接口 | 30 |
|        | 请求说明           | 30 |
|        | 返回说明           | 31 |

|                                |    |
|--------------------------------|----|
| 1.12 错误码 . . . . .             | 34 |
| 2 计费参考 . . . . .               | 39 |
| 2.1 免费额度 . . . . .             | 39 |
| 2.2 免费/付费配置 . . . . .          | 39 |
| 2.2.1 含有免费配额的接口服务 . . . . .    | 39 |
| 2.2.2 不含有免费配额的接口服务 . . . . .   | 39 |
| 2.3 计费与付费 . . . . .            | 40 |
| 2.4 价目表 . . . . .              | 40 |
| 2.4.1 通用文字识别 . . . . .         | 40 |
| 2.4.2 身份证识别 . . . . .          | 40 |
| 2.4.3 银行卡识别 . . . . .          | 41 |
| 2.4.4 通用文字识别（含位置信息版） . . . . . | 41 |
| 2.4.5 通用文字识别（含生僻字版） . . . . .  | 41 |
| 2.4.6 网络图片文字识别 . . . . .       | 42 |
| 2.5 调用失败计费策略 . . . . .         | 42 |
| 2.6 费用举例 . . . . .             | 42 |
| 2.7 余额不足提醒与欠费处理 . . . . .      | 43 |
| 2.7.1 余额不足提醒 . . . . .         | 43 |
| 2.7.2 欠费处理 . . . . .           | 43 |
| 3 C# SDK 文档 . . . . .          | 45 |
| 3.1 快速入门 . . . . .             | 45 |
| 3.1.1 安装SDK . . . . .          | 45 |
| 3.1.2 使用SDK . . . . .          | 46 |
| 3.2 错误信息格式 . . . . .           | 46 |
| 3.3 通用文字识别 . . . . .           | 47 |
| 3.4 通用文字识别（含位置信息版） . . . . .   | 50 |
| 3.5 通用文字识别（含生僻字版） . . . . .    | 53 |

|       |                |     |
|-------|----------------|-----|
| 3.6   | 网络图片文字识别       | 55  |
| 3.7   | 银行卡识别          | 58  |
| 3.8   | 身份证识别          | 59  |
| 3.9   | 行驶证识别          | 61  |
| 3.10  | 驾驶证识别          | 64  |
| 3.11  | 表格识别请求         | 67  |
| 3.12  | 表格识别结果获取       | 68  |
| 3.13  | 表格识别接口         | 71  |
| 3.14  | 版本更新记录         | 72  |
| 4     | Java SDK文档     | 73  |
| 4.1   | 安装OCR Java SDK | 73  |
| 4.2   | 快速入门           | 74  |
| 4.2.1 | 新建AipOcrClient | 74  |
| 4.2.2 | 配置AipOcrClient | 75  |
| 4.3   | 错误信息格式         | 75  |
| 4.4   | 通用文字识别         | 76  |
| 4.5   | 通用文字识别（含位置信息版） | 79  |
| 4.6   | 通用文字识别（含生僻字版）  | 83  |
| 4.7   | 网络图片文字识别       | 86  |
| 4.8   | 银行卡识别          | 88  |
| 4.9   | 身份证识别          | 89  |
| 4.10  | 行驶证识别          | 92  |
| 4.11  | 驾驶证识别          | 95  |
| 4.12  | 表格识别请求         | 99  |
| 4.13  | 表格识别结果获取       | 100 |
| 4.14  | 表格识别轮询接口       | 102 |
| 4.15  | 版本更新记录         | 103 |

|       |                |     |
|-------|----------------|-----|
| 5     | Node SDK文档     | 105 |
| 5.1   | 简介             | 105 |
| 5.2   | 安装OCR Node SDK | 105 |
| 5.3   | 快速入门           | 106 |
| 5.3.1 | 新建AipOcrClient | 106 |
| 5.4   | 通用文字识别         | 106 |
| 5.5   | 通用文字识别（含位置信息版） | 109 |
| 5.6   | 通用文字识别（含生僻字版）  | 113 |
| 5.7   | 网络图片文字识别       | 116 |
| 5.8   | 银行卡识别          | 118 |
| 5.9   | 身份证识别          | 119 |
| 5.10  | 行驶证识别          | 121 |
| 5.11  | 驾驶证识别          | 125 |
| 5.12  | 表格识别请求         | 128 |
| 5.13  | 表格识别结果         | 129 |
| 5.14  | 表格识别接口         | 132 |
| 5.15  | 错误信息格式         | 132 |
| 5.16  | 版本更新记录         | 134 |
| 6     | PHP SDK文档      | 135 |
| 6.1   | 安装OCR PHP SDK  | 135 |
| 6.2   | 快速入门           | 136 |
| 6.2.1 | 初始化一个AipOcr对象  | 136 |
| 6.2.2 | 配置AipOcrClient | 136 |
| 6.3   | 错误信息格式         | 136 |
| 6.4   | 通用文字识别         | 138 |
| 6.5   | 通用文字识别（含位置信息版） | 141 |
| 6.6   | 通用文字识别（含生僻字版）  | 145 |

|       |                   |     |
|-------|-------------------|-----|
| 6.7   | 网络图片文字识别          | 148 |
| 6.8   | 银行卡识别             | 151 |
| 6.9   | 身份证识别             | 152 |
| 6.10  | 行驶证识别             | 154 |
| 6.11  | 驾驶证识别             | 157 |
| 6.12  | 表格识别请求            | 161 |
| 6.13  | 表格识别结果获取          | 162 |
| 6.14  | 表格识别轮询接口          | 164 |
| 7     | Python SDK文档      | 165 |
| 7.1   | 安装Python SDK      | 165 |
| 7.2   | 快速入门              | 166 |
| 7.2.1 | 初始化一个AipOcr对象     | 166 |
| 7.2.2 | 配置AipOcr          | 166 |
| 7.3   | 错误信息格式            | 166 |
| 7.4   | 通用文字识别            | 168 |
| 7.5   | 通用文字识别（含位置信息版）    | 172 |
| 7.6   | 通用文字识别（含生僻字版）     | 176 |
| 7.7   | 网络图片文字识别          | 179 |
| 7.8   | 银行卡识别             | 182 |
| 7.9   | 身份证识别             | 183 |
| 7.10  | 行驶证识别             | 187 |
| 7.11  | 驾驶证识别             | 191 |
| 7.12  | 表格识别请求            | 194 |
| 7.13  | 表格识别结果获取          | 195 |
| 7.14  | 表格识别轮询接口          | 198 |
| 8     | OCR iOS SDK 开发者文档 | 199 |
| 8.1   | 简介                | 199 |



|       |                       |     |
|-------|-----------------------|-----|
| 8.2   | 快速入门                  | 199 |
| 8.2.1 | SDK工程结构               | 199 |
| 8.2.2 | Demo工程                | 200 |
| 8.2.3 | 身份验证与安全               | 201 |
|       | API Key / Secret Key  | 201 |
|       | 授权文件（安全模式）            | 201 |
|       | 自助AccessToken管理       | 202 |
| 8.3   | SDK集成与接口调用            | 202 |
| 8.3.1 | 相机接口                  | 202 |
| 8.3.2 | 数据接口                  | 203 |
|       | 通用文字识别                | 204 |
|       | 通用文字识别（不带位置信息版）       | 208 |
|       | 通用文字识别（带生僻字版）         | 211 |
|       | 网图识别                  | 211 |
|       | 银行卡识别                 | 212 |
|       | 身份证识别                 | 213 |
|       | 身份证质量控制               | 216 |
| 8.3.3 | 特殊配置                  | 217 |
|       | 截图分辨率系数               | 217 |
|       | 图片放大/缩小系数             | 217 |
|       | 自动重试                  | 217 |
| 8.4   | 错误码表                  | 217 |
| 8.5   | 版本更新记录                | 220 |
| 9     | OCR Android SDK 开发者文档 | 221 |
| 9.1   | 简介                    | 221 |
| 9.2   | 快速入门                  | 221 |
| 9.2.1 | 开发包说明                 | 221 |

|       |                                |     |
|-------|--------------------------------|-----|
| 9.2.2 | DEMO使用说明 . . . . .             | 222 |
|       | 为工程添加必要的权限 . . . . .           | 222 |
|       | Proguard配置 . . . . .           | 222 |
| 9.2.3 | 身份验证与安全 . . . . .              | 222 |
|       | API Key / Secret Key . . . . . | 223 |
|       | 授权文件（安全模式） . . . . .           | 223 |
|       | 自助AccessToken管理 . . . . .      | 224 |
| 9.3   | 接口调用说明 . . . . .               | 224 |
| 9.3.1 | OCR-UI模块 . . . . .             | 224 |
|       | OCR-UI模块调用示例 . . . . .         | 224 |
| 9.3.2 | 数据接口 . . . . .                 | 225 |
|       | 通用文字识别（含位置信息版） . . . . .       | 225 |
|       | 通用文字识别 . . . . .               | 229 |
|       | 通用文字识别(含生僻字版) . . . . .        | 232 |
|       | 网络图片文字识别 . . . . .             | 235 |
|       | 银行卡识别 . . . . .                | 238 |
|       | 身份证识别 . . . . .                | 239 |
| 9.4   | 错误码表 . . . . .                 | 242 |
| 9.5   | 版本更新记录 . . . . .               | 244 |
| 10    | 常见问题 . . . . .                 | 247 |

# 第1章

## API参考

### 1.1 简介

Hi, 您好, 欢迎使用百度OCR文字识别API服务。

本文档主要针对API开发者, 描述百度OCR文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问, 可以通过以下几种方式联系我们:

- 在百度云控制台内[提交工单](#), 咨询问题类型请选择人工智能服务;
- 加入开发者QQ群: 224994340;
- 具有免费调用额度的接口, 超过每天的免费额度后会返回错误码: 17, 错误信息: Open api daily request limit reached (每天流量超限额);

#### 1.1.1 接口能力

| 接口名称           | 接口能力简要描述                   |
|----------------|----------------------------|
| 通用文字识别         | 识别图片中的文字信息                 |
| 通用文字识别 (含位置版)  | 识别图片中的文字信息 (包含文字区域的坐标信息)   |
| 通用文字识别 (含生僻字版) | 识别图片中的文字信息 (包含对常见字和生僻字的识别) |
| 网络图片文字识别       | 识别一些网络上背景复杂, 特殊字体的文字       |
| 身份证识别          | 识别身份证正反面的文字信息              |
| 银行卡识别          | 识别银行卡的卡号并返回发卡行和卡片性质信息      |
| 驾驶证识别          | 识别机动车驾驶证所有关键字段             |

| 接口名称   | 接口能力简要描述                           |
|--------|------------------------------------|
| 行驶证识别  | 识别机动车行驶证所有关键字段                     |
| 行驶证识别  | 识别机动车行驶证所有关键字段                     |
| 表格文字识别 | 自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容 |

1.1.2 请求格式

POST方式调用

注意：Content-Type为[application/x-www-form-urlencoded](#)，然后通过[urlencode](#)格式化请求体。

1.1.3 返回格式

JSON格式

1.1.4 请求限制

请求图片需经过[base64编码](#)：图片的base64编码指将一副图片数据编码成一串字符串，使用该字符串代替图像地址。您可以首先得到图片的二进制，然后用Base64格式编码即可。

注意：图片的base64编码是不包含图片头的，如（[data:image/jpg;base64,](#)）

请求格式支持：PNG、JPG、JPEG、BMP、GIF\*\*

| 接口名称          | 图片编码后大小限额                              |
|---------------|--|
| 通用文字识别        | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |
| 通用文字识别（含位置版）  | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |
| 通用文字识别（含生僻字版） | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |
| 网络图片文字识别      | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |
| 身份证识别         | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |
| 银行卡识别         | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |

| 接口名称   | 图片编码后大小限额                              |
|--------|--|
| 驾驶证识别  | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |
| 行驶证识别  | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |
| 表格文字识别 | base64编码后大小不超过4M，最短边至少15px，最长边最大4096px |

## 1.2 调用方式

调用AI服务相关的API接口有两种调用方式，两种不同的调用方式采用相同的接口URL。

区别在于请求方式和鉴权方法不一样，请求参数和返回结果一致。

### 1.2.1 调用方式一

#### 请求URL数据格式

向API服务地址使用POST发送请求，必须在URL中带上参数：

access\_token：必须参数，参考“[Access Token获取](#)”。

注意：access\_token的有效期为30天，需要每30天进行定期更换；

POST中参数按照API接口说明调用即可。

例如文字识别API，使用HTTPS POST发送：

```
https://aip.baidubce.com/rest/2.0/ocr/v1/general?access_token=24.f9ba9c5241b67688bb4adbed8bc91dec.2592000.1485570332.282335-8574074
```

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

### 1.2.2 调用方式二

#### 请求头域内容

在请求的HTTP头域中包含以下信息：

- host（必填）

- x-bce-date ( 必填 )
- x-bce-request-id ( 选填 )
- authorization ( 必填 )
- content-type ( 必填 )
- content-length ( 选填 )

作为示例，以下是一个标准的人脸识别的请求头域内容:

```
POST /rest/2.0/face/v1/detect HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: 2015-03-24T13:02:00Z
connection: keep-alive
accept: */*
host: aip.baidubce.com
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
content-type: application/x-www-form-urlencoded
authorization: bce-auth-v1/46bd9968a6194b4bbdf0341f2286ccce/2015-03-24T13:02:00Z/
1800/host;x-bce-date/994014d96b0eb26578e039fa053a4f9003425da4bfedf33f4790882fb4c54903
```

说明：方式二鉴权使用的API认证机制authorization必须通过百度云的AK/SK生成。

### 1.3 通用文字识别

#### 1.3.1 接口描述

用户向服务请求识别某张图中的所有文字。

#### 1.3.2 请求说明

请求示例

HTTP 方法: POST

请求URL: [https://aip.baidubce.com/rest/2.0/ocr/v1/general\\_basic](https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic)

URL参数:

| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 “ <a href="#">Access Token获取</a> ” |

Header如下：

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

[请求参数](#)

| 参数            | 是否必选  | 类型     | 可选值范围                                   | 说明  |
|---------------|-------|--------|---|---|
| image         | true  | string | -                                       | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 4M，最短边至少15px，最长边最大4096px, 支持jpg/ png/ bmp 格式   |
| language_type | false | string | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |

| 参数               | 是否必选  | 类型      | 可选值范围      | 说明   |
|------------------|-------|---------|------------|--|
| detect_direction | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |
| detect_language  | FALSE | string  | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）   |

### 1.3.3 返回说明

#### 返回参数

| 字段               | 是否必选 | 类型      | 说明   |
|------------------|------|---------|--|
| direction        | 否    | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是    | uint64  | 唯一的log id，用于问题定位   |
| words_result     | 是    | array() | 识别结果数组   |
| words_result_num | 是    | uint32  | 识别结果数，表示 words_result 的元素个数  |
| +words           | 否    | string  | 识别结果字符串  |

#### 返回示例



```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 2471272194,
  "words_result_num": 2,
  "words_result":
  [
    {"words": " TSINGTAO"},
    {"words": "青島啤酒"}
  ]
}
```

## 1.4 通用文字识别（含位置信息版）

### 1.4.1 接口描述

用户向服务请求识别某张图中的所有文字，并返回文字在图中的位置信息。

### 1.4.2 请求说明

#### 请求示例

HTTP 方法：[POST](#)

请求URL：<https://aip.baidubce.com/rest/2.0/ocr/v1/general>

URL参数：

| 参数           | 值   |
|--------------|---|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考“ <a href="#">Access Token获取</a> ” |

Header如下：

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

#### 请求参数

| 参数                    | 是否必选  | 类型     | 可选值范围                                   | 说明   |
|-----------------------|-------|--------|---|--|
| image                 | true  | string | -                                       | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 4M，最短边至少 15px，最长边最大 4096px, 支持 jpg/ png/ bmp 格式                                   |
| recognize_granularity | true  | string | big、small                               | 是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置   |
| language_type         | false | string | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识别语言类型，默认为 CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |

| 参数                | 是否必选  | 类型      | 可选值范围      | 说明   |
|-------------------|-------|---------|------------|--|
| detect_direction  | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |
| detect_language   | FALSE | string  | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）   |
| vertexes_location | FALSE | string  | true、false | 是否返回文字外接多边形顶点位置，不支持单字位置。默认为false   |

### 1.4.3 返回说明

#### 返回参数

| 字段           | 是否必选 | 类型      | 说明   |
|--------------|------|---------|--|
| direction    | 否    | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id       | 是    | uint64  | 唯一的log id，用于问题定位   |
| words_result | 是    | array() | 定位和识别结果数组  |

| 字段                 | 是否必选 | 类型      | 说明  |
|--------------------|------|---------|---|
| words_result_num   | 是    | uint32  | 识别结果数，表示 words_result 的元素个数                       |
| +vertexes_location | 否    | array() | 当前为四个顶点: 左上，右上，右下，左下。当 vertexes_location=true 时存在 |
| ++x                | 是    | uint32  | 水平坐标（坐标0点为左上角）                                    |
| ++y                | 是    | uint32  | 垂直坐标（坐标0点为左上角）                                    |
| +location          | 是    | array() | 位置数组（坐标0点为左上角）                                    |
| ++left             | 是    | uint32  | 表示定位位置的长方形左上顶点的水平坐标                               |
| ++top              | 是    | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                               |
| ++width            | 是    | uint32  | 表示定位位置的长方形的宽度                                     |
| ++height           | 是    | uint32  | 表示定位位置的长方形的高度                                     |
| +words             | 否    | string  | 识别结果字符串   |
| +chars             | 否    | array() | 单字符结果，recognize_granularity=small 时存在             |
| ++location         | 是    | array() | 位置数组（坐标0点为左上角）                                    |
| +++left            | 是    | uint32  | 表示定位位置的长方形左上顶点的水平坐标                               |
| +++top             | 是    | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                               |

| 字段        | 是否必选 | 类型     | 说明              |
|-----------|------|--------|-----------------|
| +++width  | 是    | uint32 | 表示定位定位位置的长方形的宽度 |
| +++height | 是    | uint32 | 表示位置的长方形的高度     |
| ++char    | 是    | string | 单字符识别结果         |

#### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 3523983603,
  "direction": 0, //detect_direction=true时存在
  "words_result_num": 2,
  "words_result": [
    {
      "location": {
        "left": 35,
        "top": 53,
        "width": 193,
        "height": 109
      },
      "words": "感动",
      "chars": [ //recognize_granularity=small时存在
        {
          "location": {
            "left": 56,
            "top": 65,
            "width": 69,
            "height": 88
          },
          "char": "感"
        },
        {
          "location": {
            "left": 140,
            "top": 65,
```

```
        "width": 70,
        "height": 88
      },
      "char": "动"
    }
  ]
}
...
]
```

## 1.5 通用文字识别（含生僻字版）

### 1.5.1 接口描述

某些场景中，图片中的中文不光有常用字，还包含了生僻字，这时用户需要对该图进行文字识别，应使用通用文字识别（含生僻字版）。

### 1.5.2 请求说明

#### 请求示例

HTTP 方法：[POST](#)

请求URL：[https://aip.baidubce.com/rest/2.0/ocr/v1/general\\_enhanced](https://aip.baidubce.com/rest/2.0/ocr/v1/general_enhanced)

URL参数：

| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 <a href="#">“Access Token获取”</a> |

Header如下：

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

#### 请求参数

| 参数               | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|------------------|-------|---------|---|---|
| image            | true  | string  | -                                       | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式  |
| language_type    | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |
| detect_direction | false | boolean | true、false                              | 是 否 检 测 图 像 朝 向，默 认 不 检 测，即：false。朝 向 是 指 输 入 图 像 是 正 常 方 向、逆 时 针 旋 转 90/180/270 度。可 选 值 包 括：- true：检 测 朝 向；- false：不 检 测 朝 向。                      |

| 参数              | 是否必选  | 类型     | 可选值范围      | 说明                             |
|-----------------|-------|--------|------------|--------------------------------|
| detect_language | FALSE | string | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语） |

1.5.3 返回说明

返回参数

| 字段               | 是否必选 | 类型      | 说明  |
|------------------|------|---------|---|
| direction        | 否    | int32   | 图像方向，当 detect_direction=true 时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id           | 是    | uint64  | 唯一的log id，用于问题定位  |
| words_result     | 是    | array() | 识别结果数组  |
| words_result_num | 是    | uint32  | 识别结果数，表示 words_result的元素个数  |
| +words           | 否    | string  | 识别结果字符串   |

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 2471272194,
  "words_result_num": 2,
  "words_result":
```



```
[  
  {"words": " TSINGTAO"},  
  {"words": "青島啤酒"}  
]
```

## 1.6 网络图片文字识别

### 1.6.1 接口描述

用户向服务请求识别一些网络上背景复杂，特殊字体的文字。

### 1.6.2 请求说明

#### 请求示例

HTTP 方法: **POST**

请求URL: <https://aip.baidubce.com/rest/2.0/ocr/v1/webimage>

URL参数:

| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 <a href="#">“Access Token获取”</a> |

Header如下:

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下:

#### 请求参数

| 参数               | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|------------------|-------|---------|---|---|
| image            | true  | string  | -                                       | 图 像 数 据，base64 编 码，要求base64编码后大小不超过 4M，最短边至少15px，最长边最大4096px，支持jpg/ png/ bmp 格式   |
| language_type    | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |
| detect_direction | false | boolean | true、false                              | 是 否 检 测 图 像 朝 向，默 认 不 检 测，即：false。朝 向 是 指 输 入 图 像 是 正 常 方 向、逆 时 针 旋 转 90/180/270 度。可 选 值 包 括：- true：检 测 朝 向；- false：不 检 测 朝 向。                      |

| 参数              | 是否必选  | 类型     | 可选值范围      | 说明                             |
|-----------------|-------|--------|------------|--------------------------------|
| detect_language | FALSE | string | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语） |

### 1.6.3 返回说明

#### 返回参数

| 字段               | 是否必选 | 类型      | 说明   |
|------------------|------|---------|--|
| direction        | 否    | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是    | uint64  | 唯一的log id，用于问题定位   |
| words_result     | 是    | array() | 识别结果数组   |
| words_result_num | 是    | uint32  | 识别结果数，表示 words_result 的元素个数  |
| +words           | 否    | string  | 识别结果字符串  |

#### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 2471272194,
  "words_result_num": 2,
  "words_result":
```

```
[  
  {"words": " TSINGTAO"},  
  {"words": "青岛啤酒"}  
]
```

## 1.7 身份证识别

### 1.7.1 接口描述

用户向服务请求识别身份证，身份证识别包括正面和背面。

### 1.7.2 请求说明

#### 请求示例

HTTP 方法: **POST**

请求URL: <https://aip.baidubce.com/rest/2.0/ocr/v1/idcard>

URL参数:

| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 <a href="#">“Access Token获取”</a> |

Header如下:

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下:

#### 请求参数

| 参数               | 是否必选  | 类型      | 可选值范围      | 说明   |
|------------------|-------|---------|------------|--|
| detect_direction | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |
| id_card_side     | true  | string  | front、back | front：身份证正面；back：身份证背面   |
| image            | true  | string  | -          | 图像数据，base64 编码，要求base64编码后大小不超过 4M，最短边至少 15px，最长边最大 4096px, 支持 jpg/ png/ bmp 格式          |

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

### 1.7.3 返回说明

[返回参数](#)

| 字段               | 是否必选 | 类型      | 说明   |
|------------------|------|---------|--|
| direction        | 否    | int32   | 图 像 方 向， 当 detect_direction=true 时存在。 - -1: 未定义， - 0: 正向， - 1: 逆时针90度， - 2: 逆时针180度， - 3: 逆时针270度  |
| image_status     | 是    | string  | normal-识 别 正 常<br>reversed_side-未 摆 正 身 份 证<br>non_idcard-上传的图片中不包含身份证<br>blurred-身份证模糊<br>over_exposure-身 份 证 关 键 字 段 反 光 或 过 曝<br>unknown-未知状态 |
| log_id           | 是    | uint64  | 唯一的log id，用于问题定位   |
| words_result     | 是    | array() | 定位和识别结果数组  |
| words_result_num | 是    | uint32  | 识别结果数，表示 words_result 的元素个数  |
| +location        | 是    | array() | 位置数组（坐标0点为左上角）   |
| ++left           | 是    | uint32  | 表示定位位置的长方形左上顶点的水平坐标  |
| ++top            | 是    | uint32  | 表示定位位置的长方形左上顶点的垂直坐标  |
| ++width          | 是    | uint32  | 表示定位位置的长方形的宽度  |
| ++height         | 是    | uint32  | 表示定位位置的长方形的高度  |
| +words           | 否    | string  | 识别结果字符串  |

[返回示例](#)

```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 7037721,
  "direction": 0, //detect_direction=true时存在
  "words_result_num": 2,
  "words_result": {
    "住址": {
      "location": {
        "left": 227,
        "top": 235,
        "width": 229,
        "height": 51
      },
      "words": "湖北省天门市渔薪镇杨咀村一组2号",
    },
    ...
  }
}
```

## 1.8 银行卡识别

### 1.8.1 接口描述

识别银行卡并返回卡号和发卡行。

### 1.8.2 请求说明

#### 请求示例

HTTP 方法: **POST**

请求URL: <https://aip.baidubce.com/rest/2.0/ocr/v1/bankcard>

URL参数:

| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 <a href="#">“Access Token获取”</a> |

Header如下：

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

[请求参数](#)

| 参数    | 类型     | 是否必须 | 说明   |
|-------|--------|------|--|
| image | string | 是    | 图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 |

### 1.8.3 返回说明

[返回参数](#)

| 参数                | 类型     | 是否必须 | 说明                           |
|-------------------|--------|------|------------------------------|
| log_id            | uint64 | 是    | 请求标识码，随机数，唯一。                |
| result            | object | 是    | 返回结果                         |
| +bank_card_number | string | 是    | 银行卡卡号                        |
| +bank_name        | string | 是    | 银行名，不能识别时为空                  |
| +bank_card_type   | uint32 | 是    | 银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡 |

[返回示例](#)



```
{
  "log_id": 1447188951,
  "result": {
    "bank_card_number": "6225000000000000",
    "bank_name": "招商银行",
    "bank_card_type": 1
  }
}
```

## 1.9 驾驶证识别

### 1.9.1 接口描述

对机动车驾驶证所有关键字段进行识别

### 1.9.2 请求说明

[请求示例](#)

HTTP 方法: [POST](#)

请求URL: [https://aip.baidubce.com/rest/2.0/ocr/v1/driving\\_license](https://aip.baidubce.com/rest/2.0/ocr/v1/driving_license)

URL参数:

| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 <a href="#">“Access Token获取”</a> |

Header如下:

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数, 参数详情如下:

[请求参数](#)

| 参数               | 是否必选  | 类型      | 可选值范围       | 说明  |
|------------------|-------|---------|-------------|---|
| image            | true  | string  | -           | 图 像 数 据，base64 编 码，要求base64编码后大小不超过 4M，最 短 边 至 少 15px，最 长 边 最 大 4096px, 支 持 jpg/ png/ bmp 格式                              |
| detect_direction | false | boolean | true、 false | 是 否 检 测 图 像 朝 向，默 认 不 检 测，即：false。朝 向 是 指 输 入 图 像 是 正 常 方 向、逆 时 针 旋 转 90/180/270 度。可 选 值 包 括:- true：检测 朝向； - false：不检测朝向。 |

1.9.3 返回说明

返回参数

| 字段               | 是否必选 | 类型      | 说明                         |
|------------------|------|---------|----------------------------|
| log_id           | 是    | uint64  | 唯一的log id，用于问题定位           |
| words_result_num | 是    | uint32  | 识别结果数，表示 words_result的元素个数 |
| words_result     | 是    | array() | 识别结果数组                     |
| +words           | 否    | string  | 识别结果字符串                    |

返回示例

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

```
{
  "errno": 0,
  "msg": "success",
  "data": {
    "words_result_num": 10,
    "words_result": {
      "证号": {
        "words": "3208231999053090"
      },
      "有效期限": {
        "words": "6年"
      },
      "准驾车型": {
        "words": "B2"
      },
      "有效起始日期": {
        "words": "20101125"
      },
      "住址": {
        "words": "江苏省南通市海门镇秀山新城"
      },
      "姓名": {
        "words": "小欧欧"
      },
      "国籍": {
        "words": "中国"
      },
      "出生日期": {
        "words": "19990530"
      },
      "性别": {
        "words": "男"
      },
      "初次领证日期": {
        "words": "20100125"
      }
    }
  }
}
```

## 1.10 行驶证识别

### 1.10.1 接口描述

对机动车行驶证正本所有关键字段进行识别

### 1.10.2 请求说明

#### 请求示例

HTTP 方法: **POST**

请求URL: [https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle\\_license](https://aip.baidubce.com/rest/2.0/ocr/v1/vehicle_license)

URL参数:

| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 <a href="#">“Access Token获取”</a> |

Header如下:

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数, 参数详情如下:

#### 请求参数

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明  |
|-------|------|--------|-------|---|
| image | true | string | -     | 图 像 数 据, base64 编 码, 要求 base64 编码后大小不超过 4M, 最短边至少 15px, 最长边最大 4096px, 支持 jpg/ png/ bmp 格式 |

| 参数               | 是否必选  | 类型      | 可选值范围      | 说明   |
|------------------|-------|---------|------------|--|
| detect_direction | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |
| accuracy         | true  | string  | normal，缺省  | normal 使用快速服务，1200ms 左右时延；缺省或其它值使用高精度服务，1600ms左右时延                                       |

1.10.3 返回说明

返回参数

| 字段               | 是否必选 | 类型      | 说明                         |
|------------------|------|---------|----------------------------|
| log_id           | 是    | uint64  | 唯一的log id，用于问题定位           |
| words_result_num | 是    | uint32  | 识别结果数，表示 words_result的元素个数 |
| words_result     | 是    | array() | 识别结果数组                     |
| +words           | 否    | string  | 识别结果字符串                    |

返回示例

HTTP/1.1 200 OK  
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e  
Cache-Control: no-cache

Server: BWS

Date: Tue, 18 Oct 2016 02:21:01 GMT

Content-Type: application/json;charset=UTF-8

```
{
  "errno": 0,
  "msg": "success",
  "data": {
    "words_result_num": 10,
    "words_result": {
      "品牌型号": {
        "words": "保时捷GT37182RUCRE"
      },
      "发证日期": {
        "words": "20160104"
      },
      "使用性质": {
        "words": "非营运"
      },
      "发动机号码": {
        "words": "20832"
      },
      "号牌号码": {
        "words": "苏A001"
      },
      "所有人": {
        "words": "圆圆"
      },
      "住址": {
        "words": "南京市江宁区弘景大道"
      },
      "注册日期": {
        "words": "20160104"
      },
      "车辆识别代号": {
        "words": "HCE58"
      },
      "车辆类型": {
        "words": "小型轿车"
      }
    }
  }
}
```

## 1.11 表格文字识别

### 1.11.1 接口描述

自动识别表格线及表格内容，结构化输出表头、表尾及每个单元格的文字内容。

表格文字识别接口为异步接口，分为两个API：提交请求接口、获取结果接口。下面分别描述两个接口的使用方法。

### 1.11.2 表格文字识别—提交请求接口

[请求说明](#) [请求示例](#)

HTTP 方法：**POST**

请求URL：[https://aip.baidubce.com/api/v1/solution/form\\_ocr/request](https://aip.baidubce.com/api/v1/solution/form_ocr/request)

URL参数：

| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 <a href="#">“Access Token获取”</a> |

Header如下：

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

[请求参数](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明   |
|-------|------|--------|-------|--|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求 base64 编码后大小不超过 4M，最短边至少 15px，最长边最大 4096px，支持 jpg/ png/ bmp 格式 |

返回说明 返回参数

| 字段          | 是否必选 | 类型     | 说明                                      |
|-------------|------|--------|---|
| log_id      | 是    | long   | 唯一的log id，用于问题定位                        |
| result      | 是    | list   | 返回的结果列表                                 |
| +request_id | 是    | string | 该请求生成的 request_id，后续使用该request_id获取识别结果 |

返回示例

成功应答示例：

```
{
  "result" : [
    {
      "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg":"internal error"
}
```

1.11.3 表格文字识别—获取结果接口

请求说明 请求示例

HTTP 方法：[POST](#)

请求URL：[https://aip.baidubce.com/api/v1/solution/form\\_ocr/get\\_request\\_result](https://aip.baidubce.com/api/v1/solution/form_ocr/get_request_result)

URL参数：



| 参数           | 值  |
|--------------|--|
| access_token | 通过 API Key 和 Secret Key 获取的 access_token,参考 <a href="#">“Access Token获取”</a> |

Header如下：

| 参数           | 值                                 |
|--------------|-----------------------------------|
| Content-Type | application/x-www-form-urlencoded |

Body中放置请求参数，参数详情如下：

[请求参数](#)

| 参数          | 是否必选 | 类型     | 可选值范围 | 说明  |
|-------------|------|--------|-------|---|
| request_id  | 是    | string | -     | 发送表格文字识别请求时返回的request id  |
| result_type | 否    | string | -     | 期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为“excel” |

[返回说明](#) [返回参数](#)

| 字段           | 是否必选 | 类型     | 说明   |
|--------------|------|--------|--|
| log_id       | 是    | long   | 唯一的log id，用于问题定位   |
| result       | 是    | object | 返回的结果  |
| +result_data | 是    | string | 识别结果字符串，如果 request_type 是 excel，则返回 excel 的文件下载地址，如果 request_type是json，则返回json格式的字符串 |

| 字段          | 是否必选 | 类型     | 说明                       |
|-------------|------|--------|--------------------------|
| +percent    | 是    | int    | 表格识别进度（百分比）              |
| +request_id | 是    | string | 该图片对应请求的request_id       |
| +ret_code   | 是    | int    | 识别状态，1：任务未开始，2：进行中，3:已完成 |
| +ret_msg    | 是    | string | 识别状态信息，任务未开始，进行中，已完成     |

返回示例

成功应答示例：

```
{
  "result" : {
    "result_data" : "",
    "persent":100,
    "request_id": "149691317905102",
    "ret_code": 3
    "ret_msg": "已完成",
  },
  "log_id":149689853984104
}
```

当request\_type为excel时，result\_data格式样例为：

```
{
  "file_url":"https://ai.baidu.com/file/xxxxffddd"
}
```

当request\_type为json时，result\_data格式样例为：

```
{
  "form_num": 1,
  "forms": [
    {
```

```
    "header": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表头信息1",
      }
    ],
    "footer": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "表尾信息1",
      }
    ],
    "body": [
      {
        "row": [
          1
        ],
        "column": [
          1,
          2
        ],
        "word": "单元格文字",
      }
    ]
  }
}
```

其中各个参数的说明(json方式返回结果时):

| 字段       | 是否必选 | 类型     | 说明                  |
|----------|------|--------|---------------------|
| form_num | 是    | int    | 表格数量（可能一张图片中包含多个表格） |
| forms    | 是    | list   | 表格内容信息的列表           |
| +header  | 是    | list   | 每个表格中，表头数据的相关信息     |
| +footer  | 是    | list   | 表尾的相关信息             |
| +body    | 是    | list   | 表格主体部分的数据           |
| ++row    | 是    | list   | 该单元格占据的行号           |
| ++column | 是    | list   | 该单元格占据的列号           |
| ++word   | 是    | string | 该单元格中的文字信息          |

失败应答示例（详细的错误码说明见本文档底部）：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg": "internal error"
}
```

## 1.12 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- error\_code：错误码。
- error\_msg：错误描述信息，帮助理解 and 解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
```

```
"error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

| 错误码    | 错误信息                                    | 描述   |
|--------|---|--|
| 1      | Unknown error                           | 服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（224994340）或工单联系技术支持团队。 |
| 2      | Service temporarily unavailable         | 服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（224994340）或工单联系技术支持团队。  |
| 3      | Unsupported openapi method              | 调用的API不存在，请检查后重新尝试                                     |
| 4      | Open api request limit reached          | 集群超额   |
| 6      | No permission to access data            | 无权限访问该用户数据   |
| 17     | Open api daily request limit reached    | 每天请求量超额  |
| 18     | Open api qps request limit reached      | QPS超额  |
| 19     | Open api total request limit reached    | 请求总量超额   |
| 100    | Invalid parameter                       | 无效的access_token参数，请检查后重新尝试                             |
| 110    | Access token invalid or no longer valid | access_token无效   |
| 111    | Access token expired                    | access token过期   |
| 282000 | internal error                          | 服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。 |
| 216100 | invalid param                           | 请求中包含非法参数，请检查后重新尝试                                     |

| 错误码    | 错误信息                      | 描述  |
|--------|---------------------------|---|
| 216101 | not enough param          | 缺少必须的参数，请检查参数是否有遗漏  |
| 216102 | service not support       | 请求了不支持的服务，请检查调用的url   |
| 216103 | param too long            | 请求中某些参数过长，请检查后重新尝试  |
| 216110 | appid not exist           | appid不存在，请重新核对信息是否为后台应用列表中的appid                              |
| 216200 | empty image               | 图片为空，请检查后重新尝试   |
| 216201 | image format error        | 上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片           |
| 216202 | image size error          | 上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片 |
| 216630 | recognize error           | 识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。           |
| 216631 | recognize bank card error | 识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整      |
| 216633 | recognize idcard error    | 识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整                   |
| 216634 | detect error              | 检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（631977213）或工单联系技术支持团队。           |
| 282003 | missing parameters: {参数名} | 请求参数缺失  |

| 错误码    | 错误信息                        | 描述                            |
|--------|-----------------------------|-------------------------------|
| 282005 | batch processing error      | 处理批量任务时发生部分或全部错误，请根据具体错误码排查   |
| 282006 | batch task limit reached    | 批量任务处理数量超出限制，请将任务数量减少到10或10以下 |
| 282808 | request id: xxxxx not exist | request id xxxxx 不存在          |
| 282809 | result type error           | 返回结果请求错误（不属于excel或json）       |
| 282810 | image recognize error       | 图像识别错误                        |





## 第2章

# 计费参考

### 2.1 免费额度

对于通用文字识别、身份证识别、银行卡识别、通用文字识别（含位置信息版）、网络图片文字识别接口服务，各接口每日拥有500次免费调用额度。成功调用与失败调用均算作一次调用，消耗免费配额。未用完的免费额度不流转到下一日。

通用文字识别（含生僻字版）没有免费额度，开通付费后才可使用。

### 2.2 免费/付费配置

#### 2.2.1 含有免费配额的接口服务

对于通用文字识别、身份证识别、银行卡识别、通用文字识别（含位置信息版）、网络图片文字识别接口服务，免费使用和开通付费后使用的配置有较大差异，具体对比如下：

| 状态   | 500次/日免费配额 | 超过免费配额 | QPS限制   |
|------|------------|--------|---------|
| 免费状态 | 拥有         | 不响应请求  | 不保证并发   |
| 付费状态 | 拥有         | 响应请求   | 保证10次并发 |

#### 2.2.2 不含有免费配额的接口服务

对于通用文字识别（含生僻字版）接口服务，开通付费后才可以使用，相关配置如下：

| 状态   | 500次/日免费配额 | 每日调用量限额  | QPS限制   |
|------|------------|----------|---------|
| 付费状态 | 无          | 不限制，按量付费 | 保证10次并发 |

## 2.3 计费与付费

- 对于有免费配额的接口服务，开通付费后，先使用每日免费调用额度，超过额度才根据接口调用量进行计费。
- 产品采用后付费的方式，每小时对您的百度云账户进行扣费。余额不足或欠费，会给您发送短信与邮件提醒。
- 用户购买前需保证账户无欠款，且保证账户余额和可用代金券总和大于或等于100元。

## 2.4 价目表

产品采用分段阶梯定价方式，调用单价依据自然月累积调用量所落阶梯区间不同而不同。月初，上月累积的调用量清零，重新开始累积本月调用量。您可在[文字识别价格计算器](#)中，预设调用量并查看相应的价格。

### 2.4.1 通用文字识别

每日500次免费调用量，免费额度用尽后开始计费，价格如下：

| 月调用量（万次）    | 通用文字识别（元/次） |
|-------------|-------------|
| 0<调用次数≤5    | 0.0050      |
| 5<调用次数≤10   | 0.0045      |
| 10<调用次数≤20  | 0.0040      |
| 20<调用次数≤50  | 0.0035      |
| 50<调用次数≤100 | 0.0030      |
| 100<调用次数    | 0.0025      |

说明：“调用次数”只包括成功调用，调用失败不计费

### 2.4.2 身份证识别

每日500次免费调用量，免费额度用尽后开始计费，价格如下：

| 月调用量（万次）    | 身份证识别（元/次） |
|-------------|------------|
| 0<调用次数≤5    | 0.0200     |
| 5<调用次数≤10   | 0.0160     |
| 10<调用次数≤20  | 0.0130     |
| 20<调用次数≤50  | 0.0110     |
| 50<调用次数≤100 | 0.0100     |
| 100<调用次数    | 0.0090     |

说明：“调用次数”包含成功调用与少部分需计费的失败调用

### 2.4.3 银行卡识别

每日500次免费调用量，免费额度用尽后开始计费，价格如下：

月调用量（万次） | 银行卡识别（元/次） |

——— | —— |

0<调用次数≤5 | 0.0200 |

5<调用次数≤10 | 0.0160 |

10<调用次数≤20 | 0.0130 |

20<调用次数≤50 | 0.0110 |

50<调用次数≤100 | 0.0100 |

100<调用次数 | 0.0090 |

说明：“调用次数”包含成功调用与少部分需计费的失败调用

### 2.4.4 通用文字识别（含位置信息版）

每日500次免费调用量，免费额度用尽后开始计费，价格如下：

月调用量（万次） | 通用文字识别（含位置信息版）（元/次） |

——— | —— |

0<调用次数≤5 | 0.0100 |

5<调用次数≤10 | 0.0080 |

10<调用次数≤20 | 0.0065 |

20<调用次数≤50 | 0.0055 |

50<调用次数≤100 | 0.0050 |

100<调用次数 | 0.0047 |

说明：“调用次数”只包括成功调用，调用失败不计费

### 2.4.5 通用文字识别（含生僻字版）

月调用量（万次） | 通用文字识别（含生僻字版）（元/次） |

——— | —— |

0<调用次数≤5 | 0.0100 |

5<调用次数≤10 | 0.0080 |

10<调用次数≤20 | 0.0065 |

20<调用次数≤50 | 0.0055 |

50<调用次数≤100 | 0.0050 |

100<调用次数 | 0.0047 |

说明：“调用次数”只包括成功调用，调用失败不计费

## 2.4.6 网络图片文字识别

每日500次免费调用量，免费额度用尽后开始计费，价格如下：

月调用量（万次） | 网络图片文字识别（元/次） |

——— | ——— |

0<调用次数≤5 | 0.0100 |

5<调用次数≤10 | 0.0080 |

10<调用次数≤20 | 0.0065 |

20<调用次数≤50 | 0.0055 |

50<调用次数≤100 | 0.0050 |

100<调用次数 | 0.0047 |

说明：“调用次数”只包括成功调用，调用失败不计费

## 2.5 调用失败计费策略

对于文字识别下的产品，只有极少部分的失败调用算作需计费的调用，大部分失败调用均不计费。需计费的失败调用错误码如下：

| 错误码    | 错误信息                      | 描述      |
|--------|---------------------------|---------|
| 216631 | recognize bank card error | 识别银行卡错误 |
| 216633 | recognize id card error   | 识别身份证错误 |

注1：216631-识别银行卡错误码一般出现在用户上传非银行卡图片进行银行卡识别，上传的银行卡图片不完整，上传的银行卡图片不清晰等情况。

注2：216633-识别身份证错误码一般出现在用户上传非身份证图片进行身份证识别，上传的身份证图片不完整，上传的身份证图片不清晰，对某些身份证复印件进行识别等情况。

## 2.6 费用举例

从2017-2-1至2017-2-28，本月调用通用文字识别（含位置信息版）的总次数中，需计费的调用量为12万次。本月调用该接口的费用如下：前5万次落入0~5w阶梯，单价0.01元，费用为500元

中间5万次落入5~10w阶梯，单价0.008元，费用为400元

后2万次落入10~20w阶梯，单价0.0065元，费用为130元

本月费用共计：1030元

## 2.7 余额不足提醒与欠费处理

### 2.7.1 余额不足提醒

- 根据您最近3天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来3天的费用，若不足以支付，系统发送续费提醒。
- 根据您最近2天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来2天的费用，若不足以支付，系统发送续费提醒。
- 根据您最近1天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来1天的费用，若不足以支付，系统发送续费提醒。

### 2.7.2 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，含有免费配额的接口只能使用每日的免费配额，超过配额的调用量系统将不再响应，且不再保证并发处理；不含有免费配额的接口（例如通用文字识别（含生僻字版））将无法使用。



## 第3章

# C# SDK 文档

Hi, 您好, 欢迎使用百度文字识别服务。

本文档主要描述百度文字识别接口服务的相关技术内容。如果您对文档内容有任何疑问, 可以通过以下几种方式联系我们:

- 在百度云控制台内[提交工单](#), 咨询问题类型请选择人工智能服务;
- 加入开发者QQ群: 224994340;

### 3.1 快速入门

#### 3.1.1 安装SDK

##### C# SDK 结构

```
Baidu.Aip
├─ AipSdk.dll           // windows 动态库
├─ AipSdk.XML           // DLL注释
├─ Demo.cs              // Demo文件
└─ thirdparty           // 第三方依赖
```

支持平台: .Net Framework 3.5 及以上版本

##### 使用步骤

- 1.在[官方网站](#)下载C# SDK压缩工具包。
- 2.解压后, 将 [AipSdk.dll](#) 和 thirdparty 中的dll文件添加为引用
- 3.如需使用demo, 将 [Demo.cs](#) 文件添加至工程即可

3.1.2 使用SDK

[Baidu.Aip.Ocr.Ocr](#)是主要的交互类，基本使用方法如下：

```
public static void GeneralBasic()  
{  
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");  
    var image = File.ReadAllBytes("图片文件路径");  
  
    // 通用文字识别  
    var result = client.GeneralBasic(image, null);  
}
```

在上面代码中，常量[Api Key](#)与[Secret Key](#)是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在[AI服务控制台](#)中的应用列表中查看。

3.2 错误信息格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- error\_code：错误码；关于错误码的详细信息请参考通用错误码和业务相关错误码。
- error\_msg：错误描述信息，帮助理解 and 解决发生的错误。

SDK本地检测参数返回的错误码：

| error_code | error_msg                        | 备注          |
|------------|----------------------------------|-------------|
| SDK100     | image size error                 | 图片大小超限      |
| SDK101     | image length error               | 图片边长不符合要求   |
| SDK102     | read image file error            | 读取图片文件错误    |
| SDK108     | connection or read data time out | 连接超时或读取数据超时 |
| SDK109     | unsupported image format         | 不支持的图片格式    |

[服务端返回的错误码](#)

| 错误码    | 错误信息          | 描述   |
|--------|---------------|------|
| 216015 | module closed | 模块关闭 |



| 错误码    | 错误信息                         | 描述            |
|--------|------------------------------|---------------|
| 216100 | invalid param                | 非法参数          |
| 216101 | not enough param             | 参数数量不够        |
| 216102 | service not support          | 业务不支持         |
| 216103 | param too long               | 参数太长          |
| 216110 | appid not exist              | APP ID不存在     |
| 216111 | invalid userid               | 非法用户ID        |
| 216200 | empty image                  | 空的图片          |
| 216201 | image format error           | 图片格式错误        |
| 216202 | image size error             | 图片大小错误        |
| 216300 | db error                     | DB错误          |
| 216400 | backend error                | 后端系统错误        |
| 216401 | internal error               | 内部错误          |
| 216500 | unknown error                | 未知错误          |
| 216600 | id number format error       | 身份证的ID格式错误    |
| 216601 | id number and name not match | 身份证的ID和名字不匹配  |
| 216630 | recognize error              | 识别错误          |
| 216631 | recognize bank card error    | 识别银行卡错误       |
| 216632 | ocr                          | unknown error |
| 216633 | recognize idcard error       | 识别身份证错误       |
| 216634 | detect error                 | 检测错误          |
| 216635 | get mask error               | 获取mask图片错误    |
| 282000 | logic internal error         | 业务逻辑层内部错误     |
| 282001 | logic backend error          | 业务逻辑层后端服务错误   |
| 282002 | input encoding error         | 请求参数编码错误      |
| 282100 | image transcode error        | 图片压缩转码错误      |

### 3.3 通用文字识别

通用文字识别可以接受任意图片，并识别出图片中的文字以及全部文字串。

举例，要对一张图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中。多余的可选参数可在options中设置

```
public static void GeneralBasic()  
{  
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");  
    var image = File.ReadAllBytes("图片文件路径");  
  
    // 通用文字识别  
    var result = client.GeneralBasic(image, null);  
}
```

通用文字识别 [请求参数详情](#)

| 参数            | 是否必选  | 类型     | 可选值范围                                   | 说明  |
|---------------|-------|--------|---|---|
| image         | true  | string | -                                       | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 4M，最短边至少 15px，最长边最大 4096px, 支持 jpg/ png/ bmp 格式  |
| mask          | false | string | -                                       | 表示mask区域的黑白灰度图片，白色代表选中, base64编码  |
| language_type | false | string | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |

| 参数                 | 是否必选  | 类型      | 可选值范围      | 说明   |
|--------------------|-------|---------|------------|--|
| detect_direction   | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |
| detect_language    | FALSE | string  | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）   |
| classify_dimension | FALSE | string  | lottery    | 分类维度（根据 OCR 结果进行分类），逗号分隔，当前只支持 lottery。lottery：彩票分类，设置 detect_direction 有助于提升精度          |

#### 通用文字识别 返回数据参数详情

| 字段        | 必选 | 类型     | 说明   |
|-----------|----|--------|--|
| direction | 否  | int32  | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id    | 是  | uint64 | 唯一的log id，用于问题定位   |

| 字段               | 必选 | 类型      | 说明                          |
|------------------|----|---------|-----------------------------|
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数 |
| words_result     | 是  | array() | 定位和识别结果数组                   |
| +words           | 否  | string  | 识别结果字符串                     |

### 3.4 通用文字识别（含位置信息版）

通用文字识别（含位置信息版）可以接受任意图片，并识别出图片中的文字以及全部文字串，以及字符在图片中的位置信息。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张图片进行文字识别，具体的文字的内容和信息在返回的 words\_result 字段中：

```
public static void GeneralWithLocatin()
{
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");
    var image = File.ReadAllBytes("图片文件路径");

    // 带位置版本
    var result = client.GeneralWithLocatin(image, null);
}
```

[通用文字识别（含位置信息版） 请求参数详情](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明   |
|-------|------|--------|-------|--|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求 base64 编 码后大小不超过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |

| 参数                | 是否必选  | 类型      | 可选值范围   | 说明  |
|-------------------|-------|---------|---|---|
| mask              | false | string  | -   | 表示mask区域的黑白灰度图片，白色代表选中, base64编码  |
| language_type     | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction  | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| vertexes_location | false | boolean | true、false  | 是否在返回结果中标识文字位置  |
| detect_language   | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明  |
|--------------------|------|--------|---------|---|
| classify_dimension | 否    | string | lottery | 分类维度（根据 OCR 结果进行分类），逗号分隔，当前只支持 lottery。lottery：彩票分类，设置 detect_direction 有助于提升精度 |

#### 通用文字识别（含位置信息版）返回数据参数详情

| 字段                 | 必选 | 类型      | 说明   |
|--------------------|----|---------|--|
| direction          | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id             | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result       | 是  | array() | 定位和识别结果数组  |
| words_result_num   | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| +vertexes_location | 否  | array() | 当前为四个顶点：左上，右上，右下，左下。当 vertexes_location=true 时存在   |
| ++x                | 是  | uint32  | 水平坐标（坐标0点为左上角）   |
| ++y                | 是  | uint32  | 垂直坐标（坐标0点为左上角）   |
| +location          | 是  | array() | 位置数组（坐标0点为左上角）   |

| 字段         | 必选 | 类型      | 说明                                   |
|------------|----|---------|--------------------------------------|
| ++left     | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| ++top      | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| ++width    | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| ++height   | 是  | uint32  | 表示定位位置的长方形的高度                        |
| +words     | 否  | string  | 识别结果字符串                              |
| +chars     | 否  | array() | 单字符结果，recognize_granularity=small时存在 |
| ++location | 是  | array() | 位置数组（坐标0点为左上角）                       |
| +++left    | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| +++top     | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| +++width   | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| +++height  | 是  | uint32  | 表示位置的长方形的高度                          |
| ++char     | 是  | string  | 单字符识别结果                              |

### 3.5 通用文字识别（含生僻字版）

某些场景中，图片中的中文不光有常用字，还包含了生僻字，这时用户需要对该图进行文字识别，应使用通用文字识别（含生僻字版）。

举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
public static void GeneralEnhanced()  
{  
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");  
    var image = File.ReadAllBytes("图片文件路径");  
  
    // 带生僻字版  
    var result = client.GeneralEnhanced(image, null);  
}
```

通用文字识别（含生僻字版）请求参数详情

| 参数            | 是否必选  | 类型     | 可选值范围                                   | 说明  |
|---------------|-------|--------|---|---|
| image         | true  | string | -                                       | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px, 支 持 jpg/ png/ bmp 格式   |
| language_type | false | string | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |



| 参数               | 是否必选  | 类型      | 可选值范围      | 说明   |
|------------------|-------|---------|------------|--|
| detect_direction | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |

[通用文字识别（含生僻字版）](#) [返回数据参数详情](#)

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| words_result     | 是  | array() | 定位和识别结果数组  |
| +words           | 否  | string  | 识别结果字符串  |

### 3.6 网络图片文字识别

网络图片文字识别用于识别一些网络上背景复杂，特殊字体的文字。

举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
public static void WebImage()  
{  
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");  
    var image = File.ReadAllBytes("图片文件路径");  
  
    // 网图识别  
    var result = client.WebImage(image, null);  
}
```

网络图片文字识别 [请求参数详情](#)

| 参数            | 是否必选  | 类型     | 可选值范围                                   | 说明  |
|---------------|-------|--------|---|---|
| image         | true  | string | -                                       | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式  |
| language_type | false | string | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |

| 参数               | 是否必选  | 类型      | 可选值范围      | 说明   |
|------------------|-------|---------|------------|--|
| detect_direction | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |
| detect_language  | FALSE | string  | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）   |

#### 网络图片文字识别 [返回数据参数详情](#)

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| words_result     | 是  | array() | 定位和识别结果数组  |
| +words           | 否  | string  | 识别结果字符串  |

### 3.7 银行卡识别

银行卡文字识别需要接受银行卡正面带数字的清晰图片，能识别出对应的银行卡号。

举例，要对一张银行卡进行文字识别：

```
public static void BankCard()
{
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");
    var image = File.ReadAllBytes("图片文件路径");

    // 银行卡识别
    var result = client.BankCard(image);
}
```

银行卡识别 [请求参数详情](#)

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|--|------|
| image | String | 图像数据，支持本地图像文件路径，图像文件二进制数组，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 | 是    |

银行卡识别 [返回数据参数详情](#)

| 参数                 | 类型     | 描述                           |
|--------------------|--------|------------------------------|
| log_id             | UInt64 | 唯一的log id，用于问题定位             |
| result             | Object | 定位和识别结果数组                    |
| \+bank_card_number | String | 银行卡识别结果                      |
| +bank_name         | string | 银行名，不能识别时为空                  |
| +bank_card_type    | uint32 | 银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡 |

### 3.8 身份证识别

身份证识别一次只能接受身份证正面或反面的清晰图片，能识别出证件上的文字。

举例，要对一张身份证进行文字识别，可以仅传入图片信息和注明正面/反面：返回的 words\_result记录了信息，正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发日期、失效日期

```
public static void Idcard()
{
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");
    var image = File.ReadAllBytes("图片文件路径");

    // 身份证正面识别
    var result = client.IdCardFront(image);
    // 身份证背面识别
    result = client.IdCardBack(image);
}
```

身份证识别 可选参数详情

| 参数    | 必选   | 范围 | 类型     | 说明  |
|-------|------|----|--------|---|
| image | true |    | String | 图像数据，支持本地图像文件路径，图像文件二进制数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px, 支持jpg/ png/ bmp格式 |

| 参数               | 必选    | 范围               | 类型     | 说明   |
|------------------|-------|------------------|--------|--|
| detect_direction | false | true、false       | string | 是否检测图像朝向，默认不检测，即：false。可选值为：true - 检测图像朝向；false - 不检测图像朝向。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度 |
| accuracy         | false | auto、normal、high | string | 精准度，精度越高，速度越慢。default：auto   |

#### 身份证识别 [返回数据参数详情](#)

| 参数               | 类型     | 描述  |
|------------------|--------|---|
| direction        | Int32  | 图像方向，当 detect_direction=true 时存在。-1:未定义，0:正向，1: 逆时针90度， 2:逆时针180度， 3:逆时针270度      |
| log_id           | Uint64 | 唯一的log id，用于问题定位  |
| words_result     | Array  | 定位和识别结果数组，数组元素的key是身份证的主体字段（正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发日期、失效日期）。只返回识别出的字段。 |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数   |
| \+location       | Array  | 位置数组（坐标0点为左上角）  |

| 参数        | 类型     | 描述                  |
|-----------|--------|---------------------|
| \+\left   | Uint32 | 表示定位位置的长方形左上顶点的水平坐标 |
| \+\top    | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标 |
| \+\width  | Uint32 | 表示定位位置的长方形的宽度       |
| \+\height | Uint32 | 表示定位位置的长方形的高度       |
| \+words   | String | 识别结果字符串             |

### 3.9 行驶证识别

代码示例

```
public static void VehicleLicense()  
{  
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");  
    var image = File.ReadAllBytes("图片文件路径");  
    var result = client.VehicleLicense(image);  
}
```

请求参数

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|------|------|
| image | byte[] | 图像数据 | 是    |

返回参数

| 参数               | 类型     | 描述                          |
|------------------|--------|-----------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位            |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数 |
| words_result     | Array  | 定位和识别结果数组                   |

| 参数         | 类型     | 描述                  |
|------------|--------|---------------------|
| \+location | Array  | 位置数组（坐标0点为左上角）      |
| \+\+left   | Uint32 | 表示定位位置的长方形左上顶点的水平坐标 |
| \+\+top    | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标 |
| \+\+width  | Uint32 | 表示定位位置的长方形的宽度       |
| \+\+height | Uint32 | 表示定位位置的长方形的高度       |
| \+words    | String | 识别结果字符串             |

#### 返回示例：

```
{
  "log_id": 2842358861,
  "words_result_num": 10,
  "words_result": {
    "品牌型号": {
      "location": {
        "width": 140,
        "top": 249,
        "height": 25,
        "left": 258
      },
      "words": "兰博基尼NB"
    },
    "发证日期": {
      "location": {
        "width": 77,
        "top": 337,
        "height": 24,
        "left": 342
      },
      "words": "20130701"
    },
    "使用性质": {
      "location": {
        "width": 48,
        "top": 249,
```



```
        "height": 23,
        "left": 130
    },
    "words": "炫富"
},
"发动机号码": {
    "location": {
        "width": 53,
        "top": 309,
        "height": 19,
        "left": 235
    },
    "words": "888888"
},
"号牌号码": {
    "location": {
        "width": 66,
        "top": 163,
        "height": 15,
        "left": 129
    },
    "words": "LBJNDN"
},
"所有人": {
    "location": {
        "width": 49,
        "top": 190,
        "height": 19,
        "left": 130
    },
    "words": "小DU熊"
},
"住址": {
    "location": {
        "width": 280,
        "top": 220,
        "height": 20,
        "left": 130
    },
    "words": "北京市063号"
},
"注册日期": {
    "location": {
        "width": 76,
        "top": 334,
```

```

        "height": 26,
        "left": 212
    },
    "words": "30130701"
},
"车辆识别代号": {
    "location": {
        "width": 196,
        "top": 279,
        "height": 21,
        "left": 248
    },
    "words": "44444444"
},
"车辆类型": {
    "location": {
        "width": 61,
        "top": 159,
        "height": 24,
        "left": 292
    },
    "words": "豪华"
}
}
}

```

### 3.10 驾驶证识别

#### 代码示例

```

public static void DrivingLicense()
{
    var client = new Baidu.Aip.Ocr.Ocr("Api Key", "Secret Key");
    var image = File.ReadAllBytes("图片文件路径");
    var result = client.DrivingLicense(image);
}

```

#### 请求参数

| 参数    | 类型     | 描述   | 是否必须 | 是 | 否 | 返回参数 |
|-------|--------|------|------|---|---|------|
| image | byte[] | 图像数据 | 是    |   |   | 返回参数 |

| 参数               | 类型     | 描述                          |
|------------------|--------|-----------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位            |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数 |
| words_result     | Array  | 定位和识别结果数组                   |
| \+location       | Array  | 位置数组（坐标0点为左上角）              |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标         |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标         |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度               |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度               |
| \+words          | String | 识别结果字符串                     |

返回示例：

```
{
  "log_id": 2785054721,
  "words_result_num": 10,
  "words_result": {
    "证号": {
      "location": {
        "width": 181,
        "top": 168,
        "height": 22,
        "left": 182
      },
      "words": "666666666666"
    },
    "有效期限": {
      "location": {
        "width": 80,
        "top": 357,
        "height": 26,
        "left": 105
      }
    }
  }
}
```

```
    },
    "words": "8888年13月1日"
  },
  "准驾车型": {
    "location": {
      "width": 40,
      "top": 344,
      "height": 13,
      "left": 223
    },
    "words": "A0"
  },
  "住址": {
    "location": {
      "width": 342,
      "top": 223,
      "height": 47,
      "left": 65
    },
    "words": "北京市073号"
  },
  "至": {
    "location": {
      "width": 83,
      "top": 368,
      "height": 21,
      "left": 223
    },
    "words": "1997年7月1日"
  },
  "姓名": {
    "location": {
      "width": 88,
      "top": 194,
      "height": 25,
      "left": 69
    },
    "words": "小\DU熊"
  },
  "国籍": {
    "location": {
      "width": 35,
      "top": 196,
      "height": 26,
      "left": 334
```

```
        },
        "words": "中国"
    },
    "出生日期": {
        "location": {
            "width": 91,
            "top": 280,
            "height": 24,
            "left": 185
        },
        "words": "19851218"
    },
    "性别": {
        "location": {
            "width": 17,
            "top": 196,
            "height": 24,
            "left": 248
        },
        "words": "男"
    },
    "初次领证日期": {
        "location": {
            "width": 83,
            "top": 310,
            "height": 23,
            "left": 205
        },
        "words": "20081206"
    }
}
```

## 3.11 表格识别请求

### 代码示例

```
public static void FormBegin()
{
    var form = new Baidu.Aip.Ocr.Form("Api Key", "Secret Key");
    var image = File.ReadAllBytes("图片文件路径");
    form.DebugLog = false; // 是否开启调试日志
}
```

```
var result = form.BeginRecognition(image);
Console.Write(result);
}
```

请求参数

| 参数    | 类型     | 描述      | 是否必须 |
|-------|--------|---------|------|
| image | byte[] | 图像二进制数据 | 是    |

返回参数

| 参数名称        | 数据类型   | 说明   |
|-------------|--------|--|
| result      | list   | 返回的结果  |
| +request_id | string | 该请求生成的request_id，<br>后续使用该request_id获取<br>识别结果 |
| log_id      | string | 唯一的log id，排错用                                  |

返回示例：

```
{
  "result" : [
    {
      "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

3.12 表格识别结果获取

代码示例

```
public static void FormGetResult()
{
    var form = new Baidu.Aip.Ocr.Form("Api Key", "Secret Key");
    var options = new Dictionary<string,object>()
    {
```

```
        {"result_type", "json"} // 或者为excel
    };
    var result = form.GetRecognitionResult("123344", options);
    Console.Write(result);
}
```

请求参数

| 参数名称        | 数据类型   | 是否必须 | 备注  | 样例                |
|-------------|--------|------|---|-------------------|
| request_id  | string | 是    | 发送图片请求时返回的 request id   | '149691317905102' |
| result_type | string | 否    | 期望获取结果的类型，取值为“excel”时返回xls文件的地址，取值为“json”时返回json格式的字符串,默认为“excel” | “excel”           |

返回参数

| 参数名称         | 数据类型   | 说明   |
|--------------|--------|--|
| result       |        | 返回的结果  |
| +result_data | string | 识别结果字符串，如果request_type是excel，则返回excel的文件下载地址，如果request_type是json，则返回json格式的字符串 |
| +percent     | int    | 图片识别进度（百分比）  |
| +request_id  | string | 该图片对应请求的 request_id  |
| +ret_code    | int    | 识别状态，1：任务未开始，2：进行中,3:已完成   |
| +ret_msg     | string | 图片的状态信息  |
| log_id       | long   | 唯一的log id，排错用  |

当request\_type为json时，result\_data格式样例为：

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "表头信息1",
        }
      ],
      "footer": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "表尾信息1",
        }
      ],
      "body": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "单元格文字",
        }
      ]
    }
  ]
}
```



当request\_type为excel时，result\_data为url字符串：

```
"https://ai.baidu.com/file/xxxxffdd"
```

失败应答示例：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg": "unknown error"
}
```

### 3.13 表格识别接口

#### 代码示例

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口

```
“ ‘C# public static void FormToJson() { var form = new Baidu.Aip.Ocr.Form( “Api Key” ,
“Secret Key” ); var image = File.ReadAllBytes( “图片文件路径” ); form.DebugLog = false;
// 是否开启调试日志
```

```
    // 识别为Json
    var result = form.RecognizeToJson(image);
    Console.Write(result);
}
```

```
public static void FormToExcel() { var form = new Baidu.Aip.Ocr.Form( “Api Key” , “Secret
Key” ); var image = File.ReadAllBytes( “图片文件路径” ); form.DebugLog = false; // 是否
开启调试日志
```

```
    // 识别为Excel
    var result = form.RecognizeToExcel(image);
    Console.Write(result);
}
```

```
“ “
```

#### 请求参数

与表格识别请求接口返回基本相同，增加超时参数

[返回参数](#)

与表格识别结果接口返回相同

### 3.14 版本更新记录

| 上线日期      | 版本号   | 更新内容          |
|-----------|-------|---------------|
| 2017.6.30 | 1.2.0 | OCR加入表格识别     |
| 2017.6.16 | 1.1.0 | 新增驾驶证、行驶证识别接口 |
| 2017.5.4  | 1.0.0 | 第一版！          |

## 第4章

# Java SDK文档

本文档主要介绍OCR Java SDK的安装和使用。在使用本文档前，您需要先了解Optical Character Recognition(OCR)的基础知识，并已经开通了OCR服务。

### 4.1 安装OCR Java SDK

#### OCR Java SDK目录结构

```
com.baidu.aip
├── auth                //签名相关类
├── http                //Http通信相关类
├── client              //公用类
├── exception           //exception类
├── ocr
│   └── AipOcr          //AipOcr类
└── util               //工具类
```

支持 JAVA版本：1.7+

直接使用JAR包步骤如下：

- 1.在[官方网站](#)下载Java SDK压缩工具包。
- 2.将下载的[aip-java-sdk-version.zip](#)解压后，复制到工程文件夹中。
- 3.在Eclipse右键“工程 -> Properties -> Java Build Path -> Add JARs”。
- 4.添加SDK工具包[ocr\\_sdk-version.jar](#)和第三方依赖工具包[third-party/\\*.jar](#)。

其中，[version](#)为版本号，添加完成后，用户就可以在工程中使用OCR Java SDK。

## 4.2 快速入门

AipOcrClient是与Optical Character Recognition(OCR)交互的客户端，所有OCR操作都是通过AipOcrClient完成的。您可以参考新建AipOcrClient，完成初始化客户端的操作。

### 4.2.1 新建AipOcrClient

OcrClient是Optical Character Recognition的Java客户端，为使用Optical Character Recognition的开发人员提供了一系列的交互方法。

用户可以参考如下代码新建一个OcrClient：

```
public class Sample {
    //设置APPID/AK/SK
    public static final String APP_ID = " 你的_App_ID ";
    public static final String API_KEY = " 你的_Api_ID ";
    public static final String SECRET_KEY = " 你的_Secret_Key ";

    public static void main(String[] args) {
        // 初始化一个OcrClient
        AipOcr client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

        // 可选：设置网络连接参数
        client.setConnectionTimeoutInMillis(2000);
        client.setSocketTimeoutInMillis(60000);

        // 调用身份证识别接口
        String idFilePath = " test.jpg ";
        JSONObject idcardRes = client.idcard(idFilePath, true);
        System.out.println(idcardRes.toString(2));

        // 调用银行卡识别接口
        String bankFilePath = " test_bank.jpg ";
        JSONObject bankRes = client.bankcard(bankFilePath);
        System.out.println(bankRes.toString(2));

        // 调用通用识别接口
        String genFilePath = " test_basic_general.jpg ";
        JSONObject genRes = client.basicGeneral(genFilePath, new HashMap<String, String>());
        System.out.println(genRes.toString(2));

        // 调用通用识别（含位置信息）接口
        String genFilePath = " test_general.jpg ";
        JSONObject genRes = client.general(genFilePath, new HashMap<String, String>());
        System.out.println(genRes.toString(2));
    }
}
```

在上面代码中，常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的应用列表中查看。

注意：如您以前是百度云的老用户，其中API\_KEY对应百度云的“Access Key ID”，SECRET\_KEY对应百度云的“Access Key Secret”。

4.2.2 配置AipOcrClient

如果用户需要配置AipOcrClient的一些细节参数，可以在构造AipOcr之后调用接口设置参数，目前只支持以下参数：

| 接口                           | 说明                      |
|------------------------------|-------------------------|
| setConnectionTimeoutInMillis | 建立连接的超时时间（单位：毫秒）        |
| setSocketTimeoutInMillis     | 通过打开的连接传输数据的超时时间（单位：毫秒） |

4.3 错误信息格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- error\_code：错误码；关于错误码的详细信息请参考通用错误码和业务相关错误码。
- error\_msg：错误描述信息，帮助理解 and 解决发生的错误。

SDK本地检测参数返回的错误码：

| error_code | error_msg                        | 备注          |
|------------|----------------------------------|-------------|
| SDK100     | image size error                 | 图片大小超限      |
| SDK101     | image length error               | 图片边长不符合要求   |
| SDK102     | read image file error            | 读取图片文件错误    |
| SDK108     | connection or read data time out | 连接超时或读取数据超时 |
| SDK109     | unsupported image format         | 不支持的图片格式    |

服务端返回的错误码

| 错误码    | 错误信息          | 描述   |
|--------|---------------|------|
| 216015 | module closed | 模块关闭 |

| 错误码    | 错误信息                         | 描述            |
|--------|------------------------------|---------------|
| 216100 | invalid param                | 非法参数          |
| 216101 | not enough param             | 参数数量不够        |
| 216102 | service not support          | 业务不支持         |
| 216103 | param too long               | 参数太长          |
| 216110 | appid not exist              | APP ID不存在     |
| 216111 | invalid userid               | 非法用户ID        |
| 216200 | empty image                  | 空的图片          |
| 216201 | image format error           | 图片格式错误        |
| 216202 | image size error             | 图片大小错误        |
| 216300 | db error                     | DB错误          |
| 216400 | backend error                | 后端系统错误        |
| 216401 | internal error               | 内部错误          |
| 216500 | unknown error                | 未知错误          |
| 216600 | id number format error       | 身份证的ID格式错误    |
| 216601 | id number and name not match | 身份证的ID和名字不匹配  |
| 216630 | recognize error              | 识别错误          |
| 216631 | recognize bank card error    | 识别银行卡错误       |
| 216632 | ocr                          | unknown error |
| 216633 | recognize idcard error       | 识别身份证错误       |
| 216634 | detect error                 | 检测错误          |
| 216635 | get mask error               | 获取mask图片错误    |
| 282000 | logic internal error         | 业务逻辑层内部错误     |
| 282001 | logic backend error          | 业务逻辑层后端服务错误   |
| 282002 | input encoding error         | 请求参数编码错误      |
| 282100 | image transcode error        | 图片压缩转码错误      |

## 4.4 通用文字识别

通用文字识别可以接受任意图片，并识别出图片中的文字以及全部文字串。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
public void generalRecognition(AipOcr client) {
    // 参数为本地图片路径
    String imagePath = "general.jpg" ;
    JSONObject response = client.basicGeneral(imagePath);
    System.out.println(response.toString());

    // 参数为本地图片文件二进制数组
    byte[] file = readImageFile(imagePath);
    JSONObject response = client.basicGeneral(file);
    System.out.println(response.toString());
}
```

传入图片时还想增加一些自定义参数配置：

```
public void generalRecognition(AipOcr client) {
    // 自定义参数定义
    HashMap<String, String> options = new HashMap<String, String>();
    options.put(" detect_direction" , " false" );
    options.put(" language_type" , " CHN_ENG" );

    // 参数为本地图片路径
    String imagePath = "general.jpg" ;
    JSONObject response = client.basicGeneral(imagePath, options);
    System.out.println(response.toString());

    // 参数为本地图片文件二进制数组
    byte[] file = readImageFile(imagePath);
    JSONObject response = client.basicGeneral(file, options);
    System.out.println(response.toString());
}
```

[通用文字识别 请求参数详情](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明  |
|-------|------|--------|-------|---|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求 base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |

| 参数               | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|------------------|-------|---------|---|---|
| mask             | false | string  | -                                       | 表示mask区域的黑白灰度图片，白色代表选中, base64编码  |
| language_type    | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false                              | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | FALSE | string  | true、false                              | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |



| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

#### 通用文字识别 返回数据参数详情

| 字段               | 必选 | 类型      | 说明  |
|------------------|----|---------|---|
| direction        | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result_num | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| words_result     | 是  | array() | 定位和识别结果数组   |
| +words           | 否  | string  | 识别结果字符串   |

## 4.5 通用文字识别（含位置信息版）

通用文字识别（含位置信息版）可以接受任意图片，并识别出图片中的文字以及全部文字串，以及字符在图片中的位置信息。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
public void generalRecognition(AipOcr client) {  
    // 参数为本地图片路径  
    String imagePath = "general.jpg";  
    JSONObject response = client.general(imagePath);  
    System.out.println(response.toString());  
  
    // 参数为本地图片文件二进制数组  
    byte[] file = readImageFile(imagePath);  
    JSONObject response = client.general(file);  
    System.out.println(response.toString());  
}
```

传入图片时还想增加一些自定义参数配置：

```
public void generalRecognition(AipOcr client) {  
    // 自定义参数定义  
    HashMap<String, String> options = new HashMap<String, String>();  
    options.put("detect_direction", "false");  
    options.put("language_type", "CHN_ENG");  
  
    // 参数为本地图片路径  
    String imagePath = "general.jpg";  
    JSONObject response = client.general(imagePath, options);  
    System.out.println(response.toString());  
  
    // 参数为本地图片文件二进制数组  
    byte[] file = readImageFile(imagePath);  
    JSONObject response = client.general(file, options);  
    System.out.println(response.toString());  
}
```

[通用文字识别（含位置信息版）](#) 请求参数详情

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明  |
|-------|------|--------|-------|---|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求 base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |

| 参数                | 是否必选  | 类型      | 可选值范围   | 说明  |
|-------------------|-------|---------|---|---|
| mask              | false | string  | -   | 表示mask区域的黑白灰度图片，白色代表选中, base64编码  |
| language_type     | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction  | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| vertexes_location | false | boolean | true、false  | 是否在返回结果中标识文字位置  |
| detect_language   | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

#### 通用文字识别（含位置信息版）返回数据参数详情

| 字段                 | 必选 | 类型      | 说明  |
|--------------------|----|---------|---|
| direction          | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id             | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result       | 是  | array() | 定位和识别结果数组   |
| words_result_num   | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| +vertexes_location | 否  | array() | 当前为四个顶点：左上，右上，右下，左下。当vertexes_location=true时存在                                    |
| ++x                | 是  | uint32  | 水平坐标（坐标0点为左上角）  |
| ++y                | 是  | uint32  | 垂直坐标（坐标0点为左上角）  |
| +location          | 是  | array() | 位置数组（坐标0点为左上角）  |

| 字段         | 必选 | 类型      | 说明                                   |
|------------|----|---------|--------------------------------------|
| ++left     | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| ++top      | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| ++width    | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| ++height   | 是  | uint32  | 表示定位位置的长方形的高度                        |
| +words     | 否  | string  | 识别结果字符串                              |
| +chars     | 否  | array() | 单字符结果，recognize_granularity=small时存在 |
| ++location | 是  | array() | 位置数组（坐标0点为左上角）                       |
| +++left    | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| +++top     | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| +++width   | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| +++height  | 是  | uint32  | 表示位置的长方形的高度                          |
| ++char     | 是  | string  | 单字符识别结果                              |

## 4.6 通用文字识别（含生僻字版）

某些场景中，图片中的中文不光有常用字，还包含了生僻字，这时用户需要对该图进行文字识别，应使用通用文字识别（含生僻字版）。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
public void enhancedGeneral(AipOcr client) {
```

```
// 参数为本地图片路径
String imagePath = "enhance_general.jpg" ;
JSONObject response = client.enhancedGeneral(imagePath);
System.out.println(response.toString());

// 参数为本地图片文件二进制数组
byte[] file = readImageFile(imagePath);
JSONObject response = client.enhancedGeneral(file);
System.out.println(response.toString());
}
```

传入图片时还想增加一些自定义参数配置：

```
public void generalRecognition(AipOcr client) {
    // 自定义参数定义
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("detect_direction" , " false" );
    options.put(" language_type" , " CHN_ENG" );

    // 参数为本地图片路径
    String imagePath = "enhanced_general.jpg" ;
    JSONObject response = client.enhancedGeneral(imagePath, options);
    System.out.println(response.toString());

    // 参数为本地图片文件二进制数组
    byte[] file = readImageFile(imagePath);
    JSONObject response = client.enhancedGeneral(file, options);
    System.out.println(response.toString());
}
```

[通用文字识别（含生僻字版）请求参数详情](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明  |
|-------|------|--------|-------|---|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求 base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括：- true：检测朝向；- false：不检测朝向。                                |

#### 通用文字识别（含生僻字版） 返回数据参数详情

| 字段        | 必选 | 类型     | 说明   |
|-----------|----|--------|--|
| direction | 否  | int32  | 图像方向，当detect_direction=true时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id    | 是  | uint64 | 唯一的log id，用于问题定位   |

| 字段               | 必选 | 类型      | 说明                          |
|------------------|----|---------|-----------------------------|
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数 |
| words_result     | 是  | array() | 定位和识别结果数组                   |
| +words           | 否  | string  | 识别结果字符串                     |

## 4.7 网络图片文字识别

网络图片文字识别用于识别一些网络上背景复杂，特殊字体的文字。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的 words\_result 字段中：

```
public void webImageOCR(AipOcr client) {  
    // 参数为本地图片路径  
    String imagePath = "webimage.jpg";  
    JSONObject response = client.webImage(imagePath);  
    System.out.println(response.toString());  
  
    // 参数为本地图片文件二进制数组  
    byte[] file = readImageFile(imagePath);  
    JSONObject response = client.webImage(file);  
    System.out.println(response.toString());  
}
```

传入图片时还想增加一些自定义参数配置：

```
public void generalRecognition(AipOcr client) {  
    // 自定义参数定义  
    HashMap<String, String> options = new HashMap<String, String>();  
    options.put("detect_direction", "false");  
    options.put("language_type", "CHN_ENG");  
  
    // 参数为本地图片路径  
    String imagePath = "webimage.jpg";  
    JSONObject response = client.general(imagePath, options);  
    System.out.println(response.toString());  
  
    // 参数为本地图片文件二进制数组  
    byte[] file = readImageFile(imagePath);
```



```
JSONObject response = client.general(file, options);
System.out.println(response.toString());
}
```

[网络图片文字识别 请求参数详情](#)

| 参数               | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|------------------|-------|---------|---|---|
| image            | true  | string  | -                                       | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式  |
| language_type    | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |
| detect_direction | false | boolean | true、false                              | 是 否 检 测 图 像 朝 向，默 认 不 检 测，即：false。朝 向 是 指 输 入 图 像 是 正 常 方 向、逆 时 针 旋 转 90/180/270 度。可 选 值 包 括：- true：检 测 朝 向；- false：不 检 测 朝 向。                      |

| 参数              | 是否必选  | 类型     | 可选值范围      | 说明                             |
|-----------------|-------|--------|------------|--------------------------------|
| detect_language | FALSE | string | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语） |

网络图片文字识别 返回数据参数详情

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| words_result     | 是  | array() | 定位和识别结果数组  |
| +words           | 否  | string  | 识别结果字符串  |

### 4.8 银行卡识别

银行卡文字识别需要接受银行卡正面带数字的清晰图片，能识别出对应的银行卡号。

图片接受类型支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张银行卡进行文字识别：

```
public void bankcardRecognition(AipOcr client) {  
    // 参数为本地图片路径  
    String imagePath = "bankcard.jpg" ;  
    JSONObject response = client.bankcard(imagePath);  
    System.out.println(response.toString());  
}
```

```
// 参数为本地图片文件二进制数组
byte[] file = readImageFile(imagePath);
JSONObject response = client.bankcard(file);
System.out.println(response.toString());
}
```

银行卡识别 请求参数详情

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|--|------|
| image | String | 图像数据，支持本地图像文件路径，图像文件二进制数组，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 | 是    |

银行卡识别 返回数据参数详情

| 参数                 | 类型     | 描述                           |
|--------------------|--------|------------------------------|
| log_id             | Uint64 | 唯一的log id，用于问题定位             |
| result             | Object | 定位和识别结果数组                    |
| \+bank_card_number | String | 银行卡识别结果                      |
| +bank_name         | string | 银行名，不能识别时为空                  |
| +bank_card_type    | uint32 | 银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡 |

## 4.9 身份识别

身份识别一次只能接受身份证正面或反面的清晰图片，能识别出证件上的文字。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张身份证进行文字识别，可以仅传入图片信息和注明正面/反面：返回的 words\_result记录了信息，正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发日期、失效日期

```
public void idcardRecognition(AipOcr client) {
```

```
// 设置识别身份证正面参数
boolean isFront = true;
HashMap<String, String> options = new HashMap<String, String>();

// 参数为本地图片路径
String imagePath = "idcard.jpg" ;
JSONObject response = client.idcard(imagePath, isFront, options);
System.out.println(response.toString());

// 参数为本地图片文件二进制数组
byte[] file = readImageFile(imagePath);
JSONObject response = client.idcard(file, isFront, options);
System.out.println(response.toString());
}
```

传入图片时还想增加一些自定义参数配置：

```
public void idcardRecognition(AipOcr client) {
    // 设置识别身份证正面参数
    boolean isFront = true;

    // 自定义参数定义
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("detect_direction", "false");

    // 参数为本地图片路径
    String imagePath = "idcard.jpg" ;
    JSONObject response = client.idcard(imagePath, isFront, options);
    System.out.println(response.toString());

    // 参数为本地图片文件二进制数组
    byte[] file = readImageFile(imagePath);
    JSONObject response = client.idcard(file, isFront, options);
    System.out.println(response.toString());
}
```

[身份证识别 可选参数详情](#)

| 参数               | 必选    | 范围               | 类型      | 说明   |
|------------------|-------|------------------|---------|--|
| image            | true  |                  | String  | 图像数据，支持本地图像文件路径，图像文件二进制数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px, 支持jpg/ png/ bmp格式      |
| isFront          | true  | true、false       | Boolean | true：身份证正面，false：身份证背面   |
| detect_direction | false | true、false       | string  | 是否检测图像朝向，默认不检测，即：false。可选值为：true - 检测图像朝向；false - 不检测图像朝向。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度 |
| accuracy         | false | auto、normal、high | string  | 精准度，精度越高，速度越慢。 default：auto  |

#### 身份识别 返回数据参数详情

| 参数        | 类型    | 描述   |
|-----------|-------|--|
| direction | Int32 | 图像方向，当 detect_direction=true 时存在。-1:未定义，0:正向，1: 逆时针90度， 2:逆时针180度， 3:逆时针270度 |

| 参数               | 类型     | 描述  |
|------------------|--------|---|
| log_id           | Uint64 | 唯一的log id，用于问题定位  |
| words_result     | Array  | 定位和识别结果数组，数组元素的key是身份证的主体字段（正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发日期、失效日期）。只返回识别出的字段。 |
| words_result_num | Uint32 | 识别结果数，表示words_result的元素个数   |
| \+location       | Array  | 位置数组（坐标0点为左上角）  |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标   |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标   |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度   |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度   |
| \+words          | String | 识别结果字符串   |

## 4.10 行驶证识别

### 代码示例

```

public void vehicleLicenseRecognition(AipOcr client) {
    HashMap<String, String> options = new HashMap<String, String>();

    // 参数为本地图片路径
    String imagePath = "vehicle.jpg";
    JSONObject response = client.vehicleLicense(imagePath, options);
    System.out.println(response.toString());

    // 参数为本地图片文件二进制数组
    byte[] file = readImageFile(imagePath);
    JSONObject response = client.vehicleLicense(file, options);
    System.out.println(response.toString());
}

```

```
}

```

请求参数

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|------|------|
| image | String | 图像数据 | 是    |

返回参数

| 参数               | 类型     | 描述                          |
|------------------|--------|-----------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位            |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数 |
| words_result     | Array  | 定位和识别结果数组                   |
| \+location       | Array  | 位置数组（坐标0点为左上角）              |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标         |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标         |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度               |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度               |
| \+words          | String | 识别结果字符串                     |

返回示例：

```
{
  "log_id": 2842358861,
  "words_result_num": 10,
  "words_result": {
    "品牌型号": {
      "location": {
        "width": 140,
        "top": 249,
        "height": 25,
        "left": 258
      }
    }
  }
}
```

```
    },
    "words": "兰博基尼NB"
  },
  "发证日期": {
    "location": {
      "width": 77,
      "top": 337,
      "height": 24,
      "left": 342
    },
    "words": "20130701"
  },
  "使用性质": {
    "location": {
      "width": 48,
      "top": 249,
      "height": 23,
      "left": 130
    },
    "words": "炫富"
  },
  "发动机号码": {
    "location": {
      "width": 53,
      "top": 309,
      "height": 19,
      "left": 235
    },
    "words": "888888"
  },
  "号牌号码": {
    "location": {
      "width": 66,
      "top": 163,
      "height": 15,
      "left": 129
    },
    "words": "LBJNDN"
  },
  "所有人": {
    "location": {
      "width": 49,
      "top": 190,
      "height": 19,
      "left": 130
    }
```



```
    },
    "words": "小DU熊"
  },
  "住址": {
    "location": {
      "width": 280,
      "top": 220,
      "height": 20,
      "left": 130
    },
    "words": "北京市063号"
  },
  "注册日期": {
    "location": {
      "width": 76,
      "top": 334,
      "height": 26,
      "left": 212
    },
    "words": "30130701"
  },
  "车辆识别代号": {
    "location": {
      "width": 196,
      "top": 279,
      "height": 21,
      "left": 248
    },
    "words": "444444444"
  },
  "车辆类型": {
    "location": {
      "width": 61,
      "top": 159,
      "height": 24,
      "left": 292
    },
    "words": "豪华"
  }
}
```

## 4.11 驾驶证识别

代码示例

```
public void driveLicenseRecognition(AipOcr client) {
    HashMap<String, String> options = new HashMap<String, String>();

    // 参数为本地图片路径
    String imagePath = "drivelicense.jpg";
    JSONObject response = client.drivingLicense(imagePath, options);
    System.out.println(response.toString());

    // 参数为本地图片文件二进制数组
    byte[] file = readImageFile(imagePath);
    JSONObject response = client.drivingLicense(file, options);
    System.out.println(response.toString());
}
```

请求参数

| 参数 | 类型 | 描述 | 是否必须 | | :----- | :-- | :----- | :  
----- | | image | String | 图像数据 | 是 | 返回参数

| 参数               | 类型     | 描述                          |
|------------------|--------|-----------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位            |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数 |
| words_result     | Array  | 定位和识别结果数组                   |
| \+location       | Array  | 位置数组（坐标0点为左上角）              |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标         |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标         |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度               |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度               |
| \+words          | String | 识别结果字符串                     |

返回示例：

```
{
  "log_id": 2785054721,
  "words_result_num": 10,
  "words_result": {
    "证号": {
      "location": {
        "width": 181,
        "top": 168,
        "height": 22,
        "left": 182
      },
      "words": "666666666666"
    },
    "有效期限": {
      "location": {
        "width": 80,
        "top": 357,
        "height": 26,
        "left": 105
      },
      "words": "8888年13月1日"
    },
    "准驾车型": {
      "location": {
        "width": 40,
        "top": 344,
        "height": 13,
        "left": 223
      },
      "words": "A0"
    },
    "住址": {
      "location": {
        "width": 342,
        "top": 223,
        "height": 47,
        "left": 65
      },
      "words": "北京市073号"
    },
    "至": {
      "location": {
        "width": 83,
        "top": 368,
        "height": 21,
```

```
        "left": 223
      },
      "words": "1997年7月1日"
    },
    "姓名": {
      "location": {
        "width": 88,
        "top": 194,
        "height": 25,
        "left": 69
      },
      "words": "小DU熊"
    },
    "国籍": {
      "location": {
        "width": 35,
        "top": 196,
        "height": 26,
        "left": 334
      },
      "words": "中国"
    },
    "出生日期": {
      "location": {
        "width": 91,
        "top": 280,
        "height": 24,
        "left": 185
      },
      "words": "19851218"
    },
    "性别": {
      "location": {
        "width": 17,
        "top": 196,
        "height": 24,
        "left": 248
      },
      "words": "男"
    },
    "初次领证日期": {
      "location": {
        "width": 83,
        "top": 310,
        "height": 23,
```

```
        "left": 205
      },
      "words": "20081206"
    }
  }
}
```

## 4.12 表格识别请求

### 代码示例

```
public void tableRecognition(AipOcr client) {
    //异步接口
    String file = "table.jpg";
    JSONObject res = client.tableRecognitionAsync(file);
    System.out.println(res.toString(2));
}
```

### 请求参数

| 参数              | 类型            | 描述           | 是否必须 |
|-----------------|---------------|--------------|------|
| imgPath/imgData | String/byte[] | 图像文件路径或二进制数据 | 是    |

### 返回参数

| 参数名称        | 数据类型   | 说明                                      |
|-------------|--------|---|
| result      | list   | 返回的结果                                   |
| +request_id | string | 该请求生成的request_id, 后续使用该request_id获取识别结果 |
| log_id      | string | 唯一的log id, 排错用                          |

### 返回示例:

```
{
  "result" : [
    {
      "request_id" : "1234_6789"
```

```
}
],
"log_id":149689853984104
}
```

4.13 表格识别结果获取

代码示例

```
public void tableRecognition(AipOcr client) {
    //异步接口

    // 为使用接口的返回值restableRecognitionAsyncjson
    String reqId = res.getJSONArray(" result" ).getJSONObject(0).getString(" request_id" );
    // 获取结果json
    System.out.println( client.getTableRecognitionJsonResult(reqId));
    // 获取结果excel
    System.out.println( client.getTableRecognitionExcelResult(reqId));
}
```

请求参数

| 参数名称       | 数据类型   | 是否必须 | 备注                    | 样例                |
|------------|--------|------|-----------------------|-------------------|
| request_id | string | 是    | 发送图片请求时返回的 request id | ‘149691317905102’ |

返回参数

| 参数名称         | 数据类型   | 说明   |
|--------------|--------|--|
| result       |        | 返回的结果  |
| +result_data | string | 识别结果字符串，如果request_type是 excel，则返回 excel的文件下载地址，如果request_type是json，则返回json格式的字符串 |
| +percent     | int    | 图片识别进度（百分比）  |
| +request_id  | string | 该图片对应请求的 request_id  |

| 参数名称      | 数据类型   | 说明                             |
|-----------|--------|--------------------------------|
| +ret_code | int    | 识别状态, 1: 任务未开始, 2: 进行中, 3: 已完成 |
| +ret_msg  | string | 图片的状态信息                        |
| log_id    | long   | 唯一的log id, 排错用                 |

当request\_type为json时, result\_data格式样例为:

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "表头信息1",
        }
      ],
      "footer": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "表尾信息1",
        }
      ],
      "body": [
        {
          "row": [
            1
          ],
          "column": [
```

```
        1,  
        2  
    ],  
    "word": "单元格文字",  
  }  
]  
}  
]  
}
```

当request\_type为excel时，result\_data为url字符串：

```
"https://ai.baidu.com/file/xxxxffdd"
```

失败应答示例：

```
{  
  "log_id": 149319909347709,  
  "error_code": 282000  
  "error_msg": "unknown error"  
}
```

## 4.14 表格识别轮询接口

### 代码示例

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口

```
public void tableRecognition(AipOcr client){  
    //异步接口  
  
    //使用封装的同步轮询接口  
    JSONObject jsonres = client.tableRecognizeToJson(file, 20000);  
    System.out.println(jsonres.toString(2));  
  
    JSONObject excelres = client.tableRecognizeToExcelUrl(file, 20000);  
    System.out.println(excelres.toString(2));  
}
```

### 请求参数



| 参数                  | 类型            | 描述                        | 是否必须 |
|---------------------|---------------|---------------------------|------|
| imgPath/imgData     | String/byte[] | 图像文件路径或二进制数据              | 是    |
| timeoutMilliseconds | long          | 最长等待时间，超时将返回错误，一般任务在20s完成 | 是    |

#### 返回参数

与表格识别结果接口返回相同

## 4.15 版本更新记录

| 上线日期      | 版本号   | 更新内容                        |
|-----------|-------|-----------------------------|
| 2017.6.30 | 1.3.6 | 新增表格识别系列接口                  |
| 2017.6.16 | 1.3.5 | 新增驾驶证、行驶证识别接口               |
| 2017.4.13 | 1.3.2 | 新增通用文字识别（含生僻字版）和网图识别接口      |
| 2017.3.31 | 1.3.1 | 新增通用文字识别（含位置信息版）            |
| 2017.3.23 | 1.3   | 兼容Android环境                 |
| 2017.3.2  | 1.2   | 上线对图片参数要求限制，增加设置超时接口        |
| 2017.1.20 | 1.1   | 对部分云用户调用不成功的错误修复            |
| 2017.1.6  | 1.0   | 初始版本，上线身份证识别、银行卡识别和通用文字识别接口 |



## 第5章

# Node SDK文档

### 5.1 简介

本文档主要介绍OCR Node SDK的安装和使用。在使用本文档前，您需要先了解Optical Character Recognition(OCR)的基础知识，并已经开通了OCR服务。

### 5.2 安装OCR Node SDK

#### OCR Node SDK目录结构

```
├─ src
│   ├── auth                //授权相关类
│   ├── http                //Http通信相关类
│   ├── client              //公用类
│   ├── util                //工具类
│   └─ const                //常量类
├─ AipOcr.js                //人脸识别交互类
├─ index.js                 //入口文件
└─ package.json             //npm包描述文件
```

支持 node 版本 4.0+

直接使用node开发包步骤如下：

- 1.在[官方网站](#)下载node SDK压缩包。
- 2.将下载的[aip-node-sdk-version.zip](#)解压后，复制到工程文件夹中。
- 3.进入目录，运行npm install安装sdk依赖库
- 4.把目录当做模块依赖

其中，[version](#)为版本号，添加完成后，用户就可以在工程中使用OCR Node SDK。

直接使用npm安装依赖：

暂无

## 5.3 快速入门

AipOcrClient是与Optical Character Recognition(OCR)交互的客户端，所有OCR操作都是通过AipOcrClient完成的。您可以参考新建AipOcrClient，完成初始化客户端的操作。

### 5.3.1 新建AipOcrClient

OcrClient是Optical Character Recognition的node客户端，为使用Optical Character Recognition的开发人员提供了一系列的交互方法。

用户可以参考如下代码新建一个OcrClient：

```
var AipOcr = require("baidu-ai").ocr;

// 设置APPID/AK/SK
var APP_ID = "你的 App ID";
var API_KEY = "你的 Api ID";
var SECRET_KEY = "你的 Secret Key";

var client = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
```

在上面代码中，常量[APP\\_ID](#)在百度云控制台中创建，常量[API\\_KEY](#)与[SECRET\\_KEY](#)是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的应用列表中查看。

注意：如您以前是百度云的老用户，其中[API\\_KEY](#)对应百度云的“Access Key ID”，[SECRET\\_KEY](#)对应百度云的“Access Key Secret”。

## 5.4 通用文字识别

通用文字识别可以接受任意图片，并识别出图片中的文字以及全部文字串。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/general.jpg');
var base64Img = new Buffer(image).toString('base64');
client.generalBasic(base64Img).then(function(result) {
  console.log(JSON.stringify(result));
});
```

传入图片时还想增加一些自定义参数配置：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/general.jpg');
var base64Img = new Buffer(image).toString('base64');
client.generalBasic(base64Img, {detect_direction: true}).then(function(result) {
  console.log(JSON.stringify(result));
});
```

[通用文字识别 请求参数详情](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明   |
|-------|------|--------|-------|--|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

#### 通用文字识别 返回数据参数详情

| 字段               | 必选 | 类型      | 说明  |
|------------------|----|---------|---|
| direction        | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result_num | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| words_result     | 是  | array() | 定位和识别结果数组   |
| +words           | 否  | string  | 识别结果字符串   |

## 5.5 通用文字识别（含位置信息版）

通用文字识别（含位置信息版）可以接受任意图片，并识别出图片中的文字以及全部文字串，以及字符在图片中的位置信息。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/general.jpg');
var base64Img = new Buffer(image).toString('base64');
client.general(base64Img).then(function(result) {
  console.log(JSON.stringify(result));
});
```

传入图片时还想增加一些自定义参数配置：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/general.jpg');
var base64Img = new Buffer(image).toString('base64');
client.general(base64Img, {detect_direction: true}).then(function(result) {
  console.log(JSON.stringify(result));
});
```

[通用文字识别（含位置信息版）](#) [请求参数详情](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明  |
|-------|------|--------|-------|---|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求 base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |



| 参数                | 是否必选  | 类型      | 可选值范围   | 说明  |
|-------------------|-------|---------|---|---|
| language_type     | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction  | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| vertexes_location | false | boolean | true、false  | 是否在返回结果中标识文字位置  |
| detect_language   | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

#### 通用文字识别（含位置信息版）返回数据参数详情

| 字段                 | 必选 | 类型      | 说明  |
|--------------------|----|---------|---|
| direction          | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id             | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result       | 是  | array() | 定位和识别结果数组   |
| words_result_num   | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| +vertexes_location | 否  | array() | 当前为四个顶点：左上，右上，右下，左下。当vertexes_location=true时存在                                    |
| ++x                | 是  | uint32  | 水平坐标（坐标0点为左上角）  |
| ++y                | 是  | uint32  | 垂直坐标（坐标0点为左上角）  |
| +location          | 是  | array() | 位置数组（坐标0点为左上角）  |

| 字段         | 必选 | 类型      | 说明                                   |
|------------|----|---------|--------------------------------------|
| ++left     | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| ++top      | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| ++width    | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| ++height   | 是  | uint32  | 表示定位位置的长方形的高度                        |
| +words     | 否  | string  | 识别结果字符串                              |
| +chars     | 否  | array() | 单字符结果，recognize_granularity=small时存在 |
| ++location | 是  | array() | 位置数组（坐标0点为左上角）                       |
| +++left    | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| +++top     | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| +++width   | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| +++height  | 是  | uint32  | 表示位置的长方形的高度                          |
| ++char     | 是  | string  | 单字符识别结果                              |

## 5.6 通用文字识别（含生僻字版）

某些场景中，图片中的中文不光有常用字，还包含了生僻字，这时用户需要对该图进行文字识别，应使用通用文字识别（含生僻字版）。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/general.jpg');
var base64Img = new Buffer(image).toString('base64');
client.generalEnhance(base64Img).then(function(result) {
  console.log(JSON.stringify(result));
});
```

传入图片时还想增加一些自定义参数配置：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/general.jpg');
var base64Img = new Buffer(image).toString('base64');
client.generalEnhance(base64Img, {detect_direction: true}).then(function(result) {
  console.log(JSON.stringify(result));
});
```

通用文字识别（含生僻字版）请求参数详情

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明   |
|-------|------|--------|-------|--|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括：- true：检测朝向；- false：不检测朝向。                                |

#### 通用文字识别（含生僻字版） 返回数据参数详情

| 字段        | 必选 | 类型     | 说明   |
|-----------|----|--------|--|
| direction | 否  | int32  | 图像方向，当detect_direction=true时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id    | 是  | uint64 | 唯一的log id，用于问题定位   |

| 字段               | 必选 | 类型      | 说明                          |
|------------------|----|---------|-----------------------------|
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数 |
| words_result     | 是  | array() | 定位和识别结果数组                   |
| +words           | 否  | string  | 识别结果字符串                     |

### 5.7 网络图片文字识别

网络图片文字识别用于识别一些网络上背景复杂，特殊字体的文字。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的 words\_result 字段中：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/general.jpg');
var base64Img = new Buffer(image).toString('base64');
client.webImage(base64Img).then(function(result) {
  console.log(JSON.stringify(result));
});
```

传入图片时还想增加一些自定义参数配置：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/general.jpg');
var base64Img = new Buffer(image).toString('base64');
client.webImage(base64Img, {detect_direction: true}).then(function(result) {
  console.log(JSON.stringify(result));
});
```

[网络图片文字识别 请求参数详情](#)

| 参数               | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|------------------|-------|---------|---|---|
| image            | true  | string  | -                                       | 图 像 数 据，base64 编 码，要求base64编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式   |
| language_type    | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |
| detect_direction | false | boolean | true、false                              | 是 否 检 测 图 像 朝 向，默 认 不 检 测，即：false。朝 向 是 指 输 入 图 像 是 正 常 方 向、逆 时 针 旋 转 90/180/270 度。可 选 值 包 括：- true：检 测 朝 向；- false：不 检 测 朝 向。                      |

| 参数              | 是否必选  | 类型     | 可选值范围      | 说明                             |
|-----------------|-------|--------|------------|--------------------------------|
| detect_language | FALSE | string | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语） |

网络图片文字识别 返回数据参数详情

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| words_result     | 是  | array() | 定位和识别结果数组  |
| +words           | 否  | string  | 识别结果字符串  |

5.8 银行卡识别

银行卡文字识别需要接受银行卡正面带数字的清晰图片，能识别出对应的银行卡号。

图片接受类型支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张银行卡进行文字识别：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/bankcard');
var base64Img = new Buffer(image).toString('base64');
client.bankcard(base64Img).then(function(result) {
  console.log(JSON.stringify(result));
});
```



```
});
```

银行卡识别 请求参数详情

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|--|------|
| image | String | 图像数据，支持本地图像文件路径，图像文件二进制数组，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 | 是    |

银行卡识别 返回数据参数详情

| 参数                 | 类型     | 描述                           |
|--------------------|--------|------------------------------|
| log_id             | Uint64 | 唯一的log id，用于问题定位             |
| result             | Object | 定位和识别结果数组                    |
| \+bank_card_number | String | 银行卡识别结果                      |
| +bank_name         | string | 银行名，不能识别时为空                  |
| +bank_card_type    | uint32 | 银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡 |

## 5.9 身份证识别

身份证识别一次只能接受身份证正面或反面的清晰图片，能识别出证件上的文字。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张身份证进行文字识别，可以仅传入图片信息和注明正面/反面：返回的 words\_result记录了信息，正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发日期、失效日期

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/bankcard');
var base64Img = new Buffer(image).toString('base64');
```

```
// 识别身份证正面
client.idcard(base64Img, true).then(function(result) {
  console.log(JSON.stringify(result));
});

// 识别身份证背面
client.idcard(base64Img, false).then(function(result) {
  console.log(JSON.stringify(result));
});
```

传入图片时还想增加一些自定义参数配置：

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/idcard.jpg');
var base64Img = new Buffer(image).toString('base64');

// 识别身份证正面，并且带上检查图像方向参数
client.webImage(base64Img, true, {detect_direction: true}).then(function(result) {
  console.log(JSON.stringify(result));
});
```

身份证识别 可选参数详情

| 参数      | 必选   | 范围         | 类型      | 说明  |
|---------|------|------------|---------|---|
| image   | true |            | String  | 图像数据，支持本地图像文件路径，图像文件二进制数据，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px, 支持jpg/ png/ bmp格式 |
| isFront | true | true、false | Boolean | true：身份证正面，false：身份证背面  |

| 参数               | 必选    | 范围               | 类型     | 说明   |
|------------------|-------|------------------|--------|--|
| detect_direction | false | true、false       | string | 是否检测图像朝向，默认不检测，即：false。可选值为：true - 检测图像朝向；false - 不检测图像朝向。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度 |
| accuracy         | false | auto、normal、high | string | 精准度，精度越高，速度越慢。default：auto   |

[身份证识别](#) [返回数据参数详情](#)

参数 | 类型 | 描述 |  
:----- | :-- | :----- |  
direction | Int32 | 图像方向，当detect\_direction=true时存在。-1:未定义，0:正向，1: 逆时针90度， 2:逆时针180度， 3:逆时针270度 |  
log\_id | Uint64 | 唯一的log id，用于问题定位 |  
words\_result | Array | 定位和识别结果数组，数组元素的key是身份证的主体字段（正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发日期、失效日期）。只返回识别出的字段。 |  
words\_result\_num | Uint32 | 识别结果数，表示words\_result的元素个数 |  
\+location | Array | 位置数组（坐标0点为左上角） |  
\+\+left | Uint32 | 表示定位位置的长方形左上顶点的水平坐标 |  
\+\+top | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标 |  
\+\+width | Uint32 | 表示定位位置的长方形的宽度 |  
\+\+height | Uint32 | 表示定位位置的长方形的高度 |  
\+words | String | 识别结果字符串

## 5.10 行驶证识别

[代码示例](#)

```

var fs = require('fs');
var image = fs.readFileSync('assets/OCR/vehicle.jpg');
var base64Img = new Buffer(image).toString('base64');

// 识别行驶证
client.vehicleLicense(base64Img).then(function(result) {
  console.log(JSON.stringify(result));
});

client.vehicleLicense(base64Img).then(function(result) {
  console.log(JSON.stringify(result));
});

```

#### 请求参数

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|------|------|
| image | String | 图像数据 | 是    |

#### 返回参数

| 参数               | 类型     | 描述                          |
|------------------|--------|-----------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位            |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数 |
| words_result     | Array  | 定位和识别结果数组                   |
| \+location       | Array  | 位置数组（坐标0点为左上角）              |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标         |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标         |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度               |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度               |
| \+words          | String | 识别结果字符串                     |

## 返回示例:

```
{
  "log_id": 2842358861,
  "words_result_num": 10,
  "words_result": {
    "品牌型号": {
      "location": {
        "width": 140,
        "top": 249,
        "height": 25,
        "left": 258
      },
      "words": "兰博基尼NB"
    },
    "发证日期": {
      "location": {
        "width": 77,
        "top": 337,
        "height": 24,
        "left": 342
      },
      "words": "20130701"
    },
    "使用性质": {
      "location": {
        "width": 48,
        "top": 249,
        "height": 23,
        "left": 130
      },
      "words": "炫富"
    },
    "发动机号码": {
      "location": {
        "width": 53,
        "top": 309,
        "height": 19,
        "left": 235
      },
      "words": "888888"
    },
    "号牌号码": {
      "location": {
```

```
        "width": 66,
        "top": 163,
        "height": 15,
        "left": 129
      },
      "words": "LBJNDN"
    },
    "所有人": {
      "location": {
        "width": 49,
        "top": 190,
        "height": 19,
        "left": 130
      },
      "words": "小DU熊"
    },
    "住址": {
      "location": {
        "width": 280,
        "top": 220,
        "height": 20,
        "left": 130
      },
      "words": "北京市063号"
    },
    "注册日期": {
      "location": {
        "width": 76,
        "top": 334,
        "height": 26,
        "left": 212
      },
      "words": "30130701"
    },
    "车辆识别代号": {
      "location": {
        "width": 196,
        "top": 279,
        "height": 21,
        "left": 248
      },
      "words": "44444444"
    },
    "车辆类型": {
      "location": {
```

```
        "width": 61,
        "top": 159,
        "height": 24,
        "left": 292
      },
      "words": "奢华"
    }
  }
}
```

## 5.11 驾驶证识别

### 代码示例

```
var fs = require('fs');
var image = fs.readFileSync('assets/OCR/driving.jpg');
var base64Img = new Buffer(image).toString('base64');

// 识别驾驶证
client.drivingLicense(base64Img).then(function(result) {
  console.log(JSON.stringify(result));
});
```

### 请求参数

| 参数    | 类型     | 描述   | 是否必须 | 是否返回 |
|-------|--------|------|------|------|
| image | String | 图像数据 | 是    | 返回参数 |

| 参数               | 类型     | 描述                          |
|------------------|--------|-----------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位            |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数 |
| words_result     | Array  | 定位和识别结果数组                   |
| \+location       | Array  | 位置数组（坐标0点为左上角）              |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标         |

| 参数        | 类型     | 描述                  |
|-----------|--------|---------------------|
| \+\top    | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标 |
| \+\width  | Uint32 | 表示定位位置的长方形的宽度       |
| \+\height | Uint32 | 表示定位位置的长方形的高度       |
| \+words   | String | 识别结果字符串             |

返回示例：

```
{
  "log_id": 2785054721,
  "words_result_num": 10,
  "words_result": {
    "证号": {
      "location": {
        "width": 181,
        "top": 168,
        "height": 22,
        "left": 182
      },
      "words": "666666666666"
    },
    "有效期": {
      "location": {
        "width": 80,
        "top": 357,
        "height": 26,
        "left": 105
      },
      "words": "8888年13月1日"
    },
    "准驾车型": {
      "location": {
        "width": 40,
        "top": 344,
        "height": 13,
        "left": 223
      },
      "words": "A0"
    }
  }
}
```



```
"住址": {
  "location": {
    "width": 342,
    "top": 223,
    "height": 47,
    "left": 65
  },
  "words": "北京市073号"
},
"至": {
  "location": {
    "width": 83,
    "top": 368,
    "height": 21,
    "left": 223
  },
  "words": "1997年7月1日"
},
"姓名": {
  "location": {
    "width": 88,
    "top": 194,
    "height": 25,
    "left": 69
  },
  "words": "小DU熊"
},
"国籍": {
  "location": {
    "width": 35,
    "top": 196,
    "height": 26,
    "left": 334
  },
  "words": "中国"
},
"出生日期": {
  "location": {
    "width": 91,
    "top": 280,
    "height": 24,
    "left": 185
  },
  "words": "19851218"
},
```

```
      "性别": {
        "location": {
          "width": 17,
          "top": 196,
          "height": 24,
          "left": 248
        },
        "words": "男"
      },
      "初次领证日期": {
        "location": {
          "width": 83,
          "top": 310,
          "height": 23,
          "left": 205
        },
        "words": "20081206"
      }
    }
  }
}
```

### 5.12 表格识别请求

代码示例

```
let requestId;
let base64 = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');

client.tableBegin(base64).then(function(result) {
  // 获取并保存requestId
  requestId = result.result[0]['request_id'];
  console.log('<tableBegin>: ' + JSON.stringify(result));
}).catch(function (e) {
  console.log(e);
});
```

请求参数

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|------|------|
| image | String | 图像数据 | 是    |

返回参数

| 参数名称        | 数据类型   | 说明   |
|-------------|--------|--|
| result      | list   | 返回的结果  |
| +request_id | string | 该请求生成的request_id，<br>后续使用该request_id获取<br>识别结果 |
| log_id      | string | 唯一的log id，排错用                                  |

返回示例：

```
{
  "result" : [
    {
      "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

5.13 表格识别结果

代码示例

```
// requestId在表格识别请求成功返回的结果中获取
let requestId;

// 获取json格式表格识别结果
client.tableGetresult(requestId, 'json').then(function(result) {
  console.log('<tableGetresult>: ' + JSON.stringify(result));
});

// 获取excel格式表格识别结果
client.tableGetresult(requestId, 'excel').then(function(result) {
  console.log('<tableGetresult>: ' + JSON.stringify(result));
});
```

请求参数

| 参数名称        | 数据类型   | 是否必须 | 备注   | 样例                |
|-------------|--------|------|--|-------------------|
| request_id  | string | 是    | 发送图片请求时返回的 request id  | '149691317905102' |
| result_type | string | 否    | 期望获取结果的类型, 取值为“excel”时返回xls文件的地址, 取值为“json”时返回json格式的字符串, 默认为“excel” | “excel”           |

### 返回参数

| 参数名称         | 数据类型   | 说明   |
|--------------|--------|--|
| result       |        | 返回的结果  |
| +result_data | string | 识别结果字符串, 如果request_type是excel, 则返回excel的文件下载地址, 如果request_type是json, 则返回json格式的字符串 |
| +percent     | int    | 图片识别进度(百分比)  |
| +request_id  | string | 该图片对应请求的 request_id  |
| +ret_code    | int    | 识别状态, 1: 任务未开始, 2: 进行中, 3: 已完成   |
| +ret_msg     | string | 图片的状态信息  |
| log_id       | long   | 唯一的log id, 排错用   |

当request\_type为json时, result\_data格式样例为:

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
```

```

        "row": [
            1
        ],
        "column": [
            1,
            2
        ],
        "word": "表头信息1",
    }
],
"footer": [
    {
        "row": [
            1
        ],
        "column": [
            1,
            2
        ],
        "word": "表尾信息1",
    }
],
"body": [
    {
        "row": [
            1
        ],
        "column": [
            1,
            2
        ],
        "word": "单元格文字",
    }
]
}
]
}

```

当request\_type为excel时，result\_data为url字符串：

"https://ai.baidu.com/file/xxxxffddd"

失败应答示例：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg": "unknown error"
}
```

## 5.14 表格识别接口

### 代码示例

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口

```
// 图片文件base64编码
let base64 = new Buffer(fs.readFileSync('assets/OCR/table.jpg')).toString('base64');

// 以json格式获取表格识别结果，10秒的超时限制
client.tableRecorgnize(base64, 'json', 10000).then(function(result) {
  console.log('<tableRecorgnize>: ' + JSON.stringify(result));
}).catch(function(e){
  console.log(e)
});
```

### 请求参数

与表格识别请求接口返回基本相同，增加超时参数

### 返回参数

与表格识别结果接口返回相同

## 5.15 错误信息格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- error\_code：错误码；关于错误码的详细信息请参考通用错误码和业务相关错误码。
- error\_msg：错误描述信息，帮助理解 and 解决发生的错误。

### 服务端返回的错误码

| 错误码    | 错误信息                         | 描述                      |
|--------|------------------------------|-------------------------|
| 216015 | module closed                | 模块关闭                    |
| 216100 | invalid param                | 非法参数                    |
| 216101 | not enough param             | 参数数量不够                  |
| 216102 | service not support          | 业务不支持                   |
| 216103 | param too long               | 参数太长                    |
| 216110 | appid not exist              | APP ID不存在               |
| 216111 | invalid userid               | 非法用户ID                  |
| 216200 | empty image                  | 空的图片                    |
| 216201 | image format error           | 图片格式错误                  |
| 216202 | image size error             | 图片大小错误                  |
| 216300 | db error                     | DB错误                    |
| 216400 | backend error                | 后端系统错误                  |
| 216401 | internal error               | 内部错误                    |
| 216500 | unknown error                | 未知错误                    |
| 216600 | id number format error       | 身份证的ID格式错误              |
| 216601 | id number and name not match | 身份证的ID和名字不匹配            |
| 216630 | recognize error              | 识别错误                    |
| 216631 | recognize bank card error    | 识别银行卡错误                 |
| 216632 | ocr                          | unknown error           |
| 216633 | recognize idcard error       | 识别身份证错误                 |
| 216634 | detect error                 | 检测错误                    |
| 216635 | get mask error               | 获取mask图片错误              |
| 282000 | logic internal error         | 业务逻辑层内部错误               |
| 282002 | input encoding error         | 请求参数编码错误                |
| 282100 | image transcode error        | 图片压缩转码错误                |
| 282807 | image too long               | 图片超过4M                  |
| 282808 | request id: xxxxx not exist  | request id xxxxx 不存在    |
| 282809 | result type error            | 返回结果请求错误（不属于excel或json） |
| 282810 | image recognize error        | 图像识别错误                  |

## 5.16 版本更新记录

| 上线日期      | 版本号   | 更新内容      |
|-----------|-------|-----------|
| 2017.6.30 | 1.2.0 | OCR加入表格识别 |
| 2017.6.15 | 1.1.0 | 加入行驶证，驾驶证 |
| 2017.4.13 | 1.0.0 | 初版        |



## 第6章

# PHP SDK文档

本文档主要介绍OCR PHP SDK的安装和使用。在使用本文档前，您需要先了解Optical Character Recognition(OCR)的基础知识，并已经开通了OCR服务。

### 6.1 安装OCR PHP SDK

#### OCR PHP SDK目录结构

```
ocr
├── AipOcr                      //Ocr识别类
├── lib
│   ├── AipHttpClient.php      //内部http请求类
│   ├── AipBCEUtil.php        //内部工具类
│   └── AipBase                //Aip基类
└── demo
    ├── DemoAipOcr.php         //OCR服务示例
    ├── idcard.jpg             //身份证图片示例
    ├── bankcard.jpg           //银行卡图片示例
    └── general.png            //文字图片示例
```

支持 PHP版本：5.3+

使用SDK步骤如下：

- 1.在[官方网站](#)下载PHP SDK压缩工具包。
- 2.将下载的[aip-ocr-php-sdk-version.zip](#)解压后，复制AipOcr.php以及lib/\*到工程文件夹中。
- 3.引入AipOcr.php。

## 6.2 快速入门

### 6.2.1 初始化一个AipOcr对象

AipOcr类是Optical Character Recognition的PHP SDK客户端，为使用Optical Character Recognition的开发人员提供了一系列的交互方法。

用户可以参考如下代码初始化一个AipOcr对象：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的AppID'
const API_KEY = ' 你的APIKey' ;
const SECRET_KEY = ' 你的SecretKey' ;

// 初始化对象AipOcr
\AipOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
```

在上面代码中，常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的应用列表中查看。

注意：如您以前是百度云的老用户，其中API\_KEY对应百度云的“Access Key ID”，SECRET\_KEY对应百度云的“Access Key Secret”。

### 6.2.2 配置AipOcrClient

如果用户需要配置AipOcrClient的一些细节参数，可以在构造AipOcr之后调用接口设置参数，目前只支持以下参数：

| 接口                           | 说明                      |
|------------------------------|-------------------------|
| setConnectionTimeoutInMillis | 建立连接的超时时间（单位：毫秒）        |
| setSocketTimeoutInMillis     | 通过打开的连接传输数据的超时时间（单位：毫秒） |

## 6.3 错误信息格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- error\_code：错误码；关于错误码的详细信息请参考通用错误码和业务相关错误

码。

- error\_msg: 错误描述信息，帮助理解 and 解决发生的错误。

SDK本地检测参数返回的错误码：

| error_code | error_msg                        | 备注          |
|------------|----------------------------------|-------------|
| SDK100     | image size error                 | 图片大小超限      |
| SDK101     | image length error               | 图片边长不符合要求   |
| SDK102     | read image file error            | 读取图片文件错误    |
| SDK108     | connection or read data time out | 连接超时或读取数据超时 |
| SDK109     | unsupported image format         | 不支持的图片格式    |

服务端返回的错误码

| 错误码    | 错误信息                         | 描述           |
|--------|------------------------------|--------------|
| 216015 | module closed                | 模块关闭         |
| 216100 | invalid param                | 非法参数         |
| 216101 | not enough param             | 参数数量不够       |
| 216102 | service not support          | 业务不支持        |
| 216103 | param too long               | 参数太长         |
| 216110 | appid not exist              | APP ID不存在    |
| 216111 | invalid userid               | 非法用户ID       |
| 216200 | empty image                  | 空的图片         |
| 216201 | image format error           | 图片格式错误       |
| 216202 | image size error             | 图片大小错误       |
| 216300 | db error                     | DB错误         |
| 216400 | backend error                | 后端系统错误       |
| 216401 | internal error               | 内部错误         |
| 216500 | unknown error                | 未知错误         |
| 216600 | id number format error       | 身份证的ID格式错误   |
| 216601 | id number and name not match | 身份证的ID和名字不匹配 |
| 216630 | recognize error              | 识别错误         |
| 216631 | recognize bank card error    | 识别银行卡错误      |

| 错误码    | 错误信息                        | 描述                      |
|--------|-----------------------------|-------------------------|
| 216632 | ocr                         | unknown error           |
| 216633 | recognize idcard error      | 识别身份证错误                 |
| 216634 | detect error                | 检测错误                    |
| 216635 | get mask error              | 获取mask图片错误              |
| 282000 | logic internal error        | 业务逻辑层内部错误               |
| 282001 | logic backend error         | 业务逻辑层后端服务错误             |
| 282002 | input encoding error        | 请求参数编码错误                |
| 282100 | image transcode error       | 图片压缩转码错误                |
| 282807 | image too long              | 图片超过4M                  |
| 282808 | request id: xxxxx not exist | request id xxxxx 不存在    |
| 282809 | result type error           | 返回结果请求错误（不属于excel或json） |
| 282810 | image recognize error       | 图像识别错误                  |

## 6.4 通用文字识别

通用文字识别可以接受任意图片，并识别出图片中的文字。

图片参数仅支持图片文件内容。

举例，要对一张图片进行文字识别：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\AipOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 调用通用文字识别接口
$result = \AipOcr->basicGeneral(file_get_contents(' general.jpg' ));
```

传入图片时还想增加一些自定义参数配置：

```
// 引入文字识别OCR SDK
```

```
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 定义参数变量
\$option = array(' detect_direction' => ' false' , ' language_type' => " CHN_ENG" );

// 调用通用文字识别接口
\$result = \apiOcr->basicGeneral(file_get_contents(' general.jpg' ), \$option);
```

[通用文字识别 请求参数详情](#)

| 参数    | 是否必选  | 类型     | 可选值范围 | 说明  |
|-------|-------|--------|-------|---|
| image | true  | string | -     | 图 像 数 据，base64 编 码，要求base64编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |
| mask  | false | string | -     | 表示mask区域 的黑白灰度图 片，白色代表 选中, base64编 码  |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

#### 通用文字识别 返回数据参数详情

| 字段               | 必选 | 类型      | 说明  |
|------------------|----|---------|---|
| direction        | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result_num | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| words_result     | 是  | array() | 定位和识别结果数组   |
| +words           | 否  | string  | 识别结果字符串   |

## 6.5 通用文字识别（含位置信息版）

通用文字识别（含位置信息版）可以接受任意图片，并识别出图片中的文字以及全部文字串，以及字符在图片中的位置信息。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 调用通用文字识别接口
$result = \apiOcr->general(file_get_contents(' general.jpg' ));
```

传入图片时还想增加一些自定义参数配置：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 定义参数变量
\option = array(' detect_direction' => ' false' , ' language_type' => " CHN_ENG" );

// 调用通用文字识别接口
$result = \apiOcr->general(file_get_contents(' general.jpg' ), \option);
```

[通用文字识别（含位置信息版）请求参数详情](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明  |
|-------|------|--------|-------|---|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |



| 参数                | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|-------------------|-------|---------|---|---|
| mask              | false | string  | -                                       | 表示mask区域的黑白灰度图片，白色代表选中, base64编码  |
| language_type     | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction  | false | boolean | true、false                              | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language   | FALSE | string  | true、false                              | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |
| vertexes_location | false | boolean | true、false                              | 是否在返回结果中标识文字位置  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

#### 通用文字识别（含位置信息版）返回数据参数详情

| 字段                 | 必选 | 类型      | 说明  |
|--------------------|----|---------|---|
| direction          | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id             | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result       | 是  | array() | 定位和识别结果数组   |
| words_result_num   | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| +vertexes_location | 否  | array() | 当前为四个顶点：左上，右上，右下，左下。当vertexes_location=true时存在                                    |
| ++x                | 是  | uint32  | 水平坐标（坐标0点为左上角）  |
| ++y                | 是  | uint32  | 垂直坐标（坐标0点为左上角）  |
| +location          | 是  | array() | 位置数组（坐标0点为左上角）  |

| 字段         | 必选 | 类型      | 说明                                   |
|------------|----|---------|--------------------------------------|
| ++left     | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| ++top      | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| ++width    | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| ++height   | 是  | uint32  | 表示定位位置的长方形的高度                        |
| +words     | 否  | string  | 识别结果字符串                              |
| +chars     | 否  | array() | 单字符结果，recognize_granularity=small时存在 |
| ++location | 是  | array() | 位置数组（坐标0点为左上角）                       |
| +++left    | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| +++top     | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| +++width   | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| +++height  | 是  | uint32  | 表示位置的长方形的高度                          |
| ++char     | 是  | string  | 单字符识别结果                              |

## 6.6 通用文字识别（含生僻字版）

某些场景中，图片中的中文不光有常用字，还包含了生僻字，这时用户需要对该图进行文字识别，应使用通用文字识别（含生僻字版）。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
// 引入文字识别OCR SDK
```

```
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 调用生僻字识别接口
$result = \apiOcr->enhancedGeneral(file_get_contents(' enhanced_general.jpg' ));
```

传入图片时还想增加一些自定义参数配置：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 定义参数变量
\option = array(' detect_direction' => ' false' , ' language_type' => " CHN_ENG" );

// 调用生僻字识别接口
$result = \apiOcr->enhancedGeneral(file_get_contents(' enhanced_general.jpg' ), \
    $option);
```

[通用文字识别（含生僻字版）请求参数详情](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明   |
|-------|------|--------|-------|--|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。可选值包括：- true：检测朝向；- false：不检测朝向。                                |

#### 通用文字识别（含生僻字版）返回数据参数详情

| 字段        | 必选 | 类型     | 说明   |
|-----------|----|--------|--|
| direction | 否  | int32  | 图像方向，当detect_direction=true时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id    | 是  | uint64 | 唯一的log id，用于问题定位   |

| 字段               | 必选 | 类型      | 说明                          |
|------------------|----|---------|-----------------------------|
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数 |
| words_result     | 是  | array() | 定位和识别结果数组                   |
| +words           | 否  | string  | 识别结果字符串                     |

## 6.7 网络图片文字识别

网络图片文字识别用于识别一些网络上背景复杂，特殊字体的文字。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的 words\_result 字段中：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 调用网络图片文字识别接口
$result = \apiOcr->webImage(file_get_contents(' web_image.jpg' ));
```

传入图片时还想增加一些自定义参数配置：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);
```

```
// 定义参数变量
\ $option = array( ' detect_direction' => ' true' , ' language_type' => " CHN_ENG" );

// 调用网络图片文字识别接口
\ $result = \ $apiOcr->webImage(file_get_contents( ' web_image.jpg' ), \ $option);
```

网络图片文字识别 [请求参数详情](#)

| 参数            | 是否必选  | 类型     | 可选值范围   | 说明  |
|---------------|-------|--------|---|---|
| image         | true  | string | -   | 图 像 数 据，base64 编 码，要求base64编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px, 支 持 jpg/ png/ bmp 格式  |
| language_type | false | string | CHN_ENG、 ENG、 POR、 FRE、 GER、 ITA、 SPA、 RUS、 JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER： 德 语；- ITA： 意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |

| 参数               | 是否必选  | 类型      | 可选值范围      | 说明   |
|------------------|-------|---------|------------|--|
| detect_direction | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |
| detect_language  | FALSE | string  | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）   |

#### 网络图片文字识别 [返回数据参数详情](#)

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| words_result     | 是  | array() | 定位和识别结果数组  |
| +words           | 否  | string  | 识别结果字符串  |



## 6.8 银行卡识别

银行卡文字识别需要接受银行卡正面带数字的清晰图片，能识别出对应的银行卡号。

图片参数仅支持图片文件内容。

举例，要对一张银行卡进行文字识别：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 调用银行卡识别接口
$result = \apiOcr->bankcard(file_get_contents(' bankcard.jpg' ));
```

银行卡识别 [请求参数详情](#)

| 参数    | 类型     | 描述            | 是否必须 |
|-------|--------|---------------|------|
| image | String | 图像数据，仅支持图像文件流 | 是    |

银行卡识别 [返回数据参数详情](#)

| 参数                 | 类型     | 描述                           |
|--------------------|--------|------------------------------|
| log_id             | Unit64 | 唯一的log id，用于问题定位             |
| result             | Object | 定位和识别结果数组                    |
| \+bank_card_number | String | 银行卡识别结果                      |
| +bank_name         | string | 银行名，不能识别时为空                  |
| +bank_card_type    | uint32 | 银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡 |

## 6.9 身份证识别

身份证识别一次只能接受身份证正面或反面的清晰图片，能识别出证件上的文字。

图片参数仅支持图片文件内容。

举例，要对一张身份证进行文字识别，可以仅传入图片信息和注明正面/反面：返回的result记录了信息，正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发日期、失效日期

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = '你的_App_ID'
const API_KEY = '你的_API_Key' ;
const SECRET_KEY = '你的_Secret_Key' ;

// 初始化AipOcr
\AipOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 设置识别身份证正面参数
$isFront = false;

// 调用身份证识别接口
$result = \AipOcr->idcard(file_get_contents('idcard.jpg'), $isFront);
```

传入图片时还想增加一些自定义参数配置：

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = '你的_App_ID'
const API_KEY = '你的_API_Key' ;
const SECRET_KEY = '你的_Secret_Key' ;

// 初始化AipOcr
\AipOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 设置识别身份证正面参数
$isFront = false;

// 定义参数变量
$options = array('detectDirection' => false, 'accuracy' => 'high' );

// 调用身份证识别接口
$result = \AipOcr->idcard(file_get_contents('idcard.jpg'), $isFront, $options);
```

身份证识别 [请求参数详情](#)

| 参数               | 类型      | 描述  | 是否必须                          |
|------------------|---------|---|-------------------------------|
| id_card_side     | String  | front: 身份证正面, back: 身份证背面   | 是                             |
| image            | String  | 图像数据, 仅支持图像文件流  | 是                             |
| accuracy         | string  | auto、normal、high  | 精准度, 精度越高, 速度越慢。default: auto |
| detect_direction | Boolean | 检测图像朝向(指输入图像是正常方向、逆时针旋转90/180/270度), 有效值: true、false, 默认值: false。 | 否                             |

身份证识别 [返回数据参数详情](#)

| 参数               | 类型     | 描述  |
|------------------|--------|---|
| direction        | Int32  | 图像方向, 当 detect_direction=true 时存在。-1:未定义, 0:正向, 1: 逆时针90度, 2:逆时针180度, 3:逆时针270度       |
| log_id           | Uint64 | 唯一的log id, 用于问题定位   |
| words_result     | Array  | 定位和识别结果数组, 数组元素的key是身份证的主体字段(正面支持: 住址、公民身份号码、出生、姓名、性别、民族, 背面支持: 签发日期、失效日期)。只返回识别出的字段。 |
| words_result_num | Uint32 | 识别结果数, 表示 words_result 的元素个数  |
| \+location       | Array  | 位置数组(坐标0点为左上角)  |

| 参数        | 类型     | 描述                  |
|-----------|--------|---------------------|
| \+\left   | Uint32 | 表示定位位置的长方形左上顶点的水平坐标 |
| \+\top    | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标 |
| \+\width  | Uint32 | 表示定位位置的长方形的宽度       |
| \+\height | Uint32 | 表示定位位置的长方形的高度       |
| \+words   | String | 识别结果字符串             |

6.10 行驶证识别

代码示例

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 调用行驶证识别接口
$result = \apiOcr->vehicleLicense(file_get_contents(' vehicleLicense.jpg' ));
```

请求参数

| 参数               | 类型     | 描述  | 是否必须 |
|------------------|--------|---|------|
| detect_direction | String | 是否检测图像朝向，默认不检测，即：false。可选值包括true - 检测朝向；false - 不检测朝向。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。 | 否    |

| 参数       | 类型     | 描述                          | 是否必须 |
|----------|--------|-----------------------------|------|
| image    | String | 图像数据                        | 是    |
| accuracy | string | normal 使用快速服务;缺省或其它值使用高精度服务 | 否    |

#### 返回参数

| 参数               | 类型     | 描述                        |
|------------------|--------|---------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位          |
| words_result_num | Uint32 | 识别结果数，表示words_result的元素个数 |
| words_result     | Array  | 定位和识别结果数组                 |
| \+location       | Array  | 位置数组（坐标0点为左上角）            |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标       |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标       |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度             |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度             |
| \+words          | String | 识别结果字符串                   |

#### 返回示例：

```
{
  "log_id": 2842358861,
  "words_result_num": 10,
  "words_result": {
    "品牌型号": {
      "location": {
        "width": 140,
        "top": 249,
        "height": 25,
        "left": 258
```

```
    },
    "words": "兰博基尼NB"
  },
  "发证日期": {
    "location": {
      "width": 77,
      "top": 337,
      "height": 24,
      "left": 342
    },
    "words": "20130701"
  },
  "使用性质": {
    "location": {
      "width": 48,
      "top": 249,
      "height": 23,
      "left": 130
    },
    "words": "炫富"
  },
  "发动机号码": {
    "location": {
      "width": 53,
      "top": 309,
      "height": 19,
      "left": 235
    },
    "words": "888888"
  },
  "号牌号码": {
    "location": {
      "width": 66,
      "top": 163,
      "height": 15,
      "left": 129
    },
    "words": "LBJNDN"
  },
  "所有人": {
    "location": {
      "width": 49,
      "top": 190,
      "height": 19,
      "left": 130
    }
```

```
    },
    "words": "小\DU熊"
  },
  "住址": {
    "location": {
      "width": 280,
      "top": 220,
      "height": 20,
      "left": 130
    },
    "words": "北京市063号"
  },
  "注册日期": {
    "location": {
      "width": 76,
      "top": 334,
      "height": 26,
      "left": 212
    },
    "words": "30130701"
  },
  "车辆识别代号": {
    "location": {
      "width": 196,
      "top": 279,
      "height": 21,
      "left": 248
    },
    "words": "444444444"
  },
  "车辆类型": {
    "location": {
      "width": 61,
      "top": 159,
      "height": 24,
      "left": 292
    },
    "words": "豪华"
  }
}
```

## 6.11 驾驶证识别

代码示例

```
// 引入文字识别OCR SDK
require_once 'AipOcr.php' ;

// 定义常量
const APP_ID = ' 你的_App_ID'
const API_KEY = ' 你的_API_Key' ;
const SECRET_KEY = ' 你的_Secret_Key' ;

// 初始化AipOcr
\apiOcr = new AipOcr(APP_ID, API_KEY, SECRET_KEY);

// 调用驾驶证识别接口
$result = \apiOcr->drivingLicense(file_get_contents(' drivingLicense.jpg' ));
```

请求参数

| 参数 | 类型 | 描述 | 是否必须 | | :----- | :-- | :----- | : |  
----- | | image | String | 图像数据 | 是 | 返回参数

| 参数               | 类型     | 描述                          |
|------------------|--------|-----------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位            |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数 |
| words_result     | Array  | 定位和识别结果数组                   |
| \+location       | Array  | 位置数组（坐标0点为左上角）              |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标         |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标         |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度               |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度               |
| \+words          | String | 识别结果字符串                     |

返回示例：



```
{
  "log_id": 2785054721,
  "words_result_num": 10,
  "words_result": {
    "证号": {
      "location": {
        "width": 181,
        "top": 168,
        "height": 22,
        "left": 182
      },
      "words": "66666666666"
    },
    "有效期限": {
      "location": {
        "width": 80,
        "top": 357,
        "height": 26,
        "left": 105
      },
      "words": "8888年13月1日"
    },
    "准驾车型": {
      "location": {
        "width": 40,
        "top": 344,
        "height": 13,
        "left": 223
      },
      "words": "A0"
    },
    "住址": {
      "location": {
        "width": 342,
        "top": 223,
        "height": 47,
        "left": 65
      },
      "words": "北京市073号"
    },
    "至": {
      "location": {
        "width": 83,
        "top": 368,
        "height": 21,
```

```
        "left": 223
      },
      "words": "1997年7月1日"
    },
    "姓名": {
      "location": {
        "width": 88,
        "top": 194,
        "height": 25,
        "left": 69
      },
      "words": "小DU熊"
    },
    "国籍": {
      "location": {
        "width": 35,
        "top": 196,
        "height": 26,
        "left": 334
      },
      "words": "中国"
    },
    "出生日期": {
      "location": {
        "width": 91,
        "top": 280,
        "height": 24,
        "left": 185
      },
      "words": "19851218"
    },
    "性别": {
      "location": {
        "width": 17,
        "top": 196,
        "height": 24,
        "left": 248
      },
      "words": "男"
    },
    "初次领证日期": {
      "location": {
        "width": 83,
        "top": 310,
        "height": 23,
```

```
        "left": 205
      },
      "words": "20081206"
    }
  }
}
```

## 6.12 表格识别请求

### 代码示例

```
\$aipOcr->tableRecognitionAsync(file_get_contents(' table.jpg' ));
```

### 请求参数

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|------|------|
| image | String | 图像数据 | 是    |

### 返回参数

| 参数名称        | 数据类型   | 说明   |
|-------------|--------|--|
| result      | list   | 返回的结果  |
| +request_id | string | 该请求生成的request_id，<br>后续使用该request_id获取<br>识别结果 |
| log_id      | string | 唯一的log id，排错用                                  |

### 返回示例：

```
{
  "result" : [
    {
      "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

## 6.13 表格识别结果获取

### 代码示例

```
\$aipOcr->getTableRecognitionResult(' 9041866_24' , array(
    ' result_type' => ' json' ,
));
```

### 请求参数

| 参数名称        | 数据类型   | 是否必须 | 备注   | 样例                |
|-------------|--------|------|--|-------------------|
| request_id  | string | 是    | 发送图片请求时返回的 request id  | '149691317905102' |
| result_type | string | 否    | 期望获取结果的类型, 取值为“excel”时返回xls文件的地址, 取值为“json”时返回json格式的字符串, 默认为“excel” | “excel”           |

### 返回参数

| 参数名称         | 数据类型   | 说明   |
|--------------|--------|--|
| result       |        | 返回的结果  |
| +result_data | string | 识别结果字符串, 如果request_type是excel, 则返回excel的文件下载地址, 如果request_type是json, 则返回json格式的字符串 |
| +percent     | int    | 图片识别进度(百分比)  |
| +request_id  | string | 该图片对应请求的 request_id  |
| +ret_code    | int    | 识别状态, 1: 任务未开始, 2: 进行中, 3: 已完成   |
| +ret_msg     | string | 图片的状态信息  |
| log_id       | long   | 唯一的log id, 排错用   |

当request\_type为json时，result\_data格式样例为：

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "表头信息1",
        }
      ],
      "footer": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "表尾信息1",
        }
      ],
      "body": [
        {
          "row": [
            1
          ],
          "column": [
            1,
            2
          ],
          "word": "单元格文字",
        }
      ]
    }
  ]
}
```

```
}
```

当request\_type为excel时，result\_data为url字符串：

```
"https://ai.baidu.com/file/xxxxffddd"
```

失败应答示例：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg": "unknown error"
}
```

## 6.14 表格识别轮询接口

### 代码示例

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口

```
\$aipOcr->tableRecognition(
file_get_contents('table.jpg'),
array(
'result_type' => 'json',
)
);
```

### 返回参数

与表格识别结果接口返回相同

## 第7章

# Python SDK文档

本文档主要介绍OCR图像识别Python SDK的使用，在使用前请先确保已创建应用并具有相应权限。

### 7.1 安装Python SDK

#### Python SDK目录结构

```
|— README.md
|— aip                //SDK目录
|   |— __init__.py    //导出类
|   |— imagecesor.py  //图像审核
|   |— base.py        //aip基类
|   |— face.py        //人脸识别
|   |— http.py        //http请求
|   |— nlp.py         //nlp自然语言处理
|   |— ocr.py         //OCR图像识别
|— setup.py          //setuptools安装
```

支持 Python版本: 2.7.+ ,3.+

安装使用SDK有如下方式:

- 如果已安装pip, 执行[pip install baidu-aip](#)即可。
- 如果已安装setuptools, 执行[python setup.py install](#)即可。

## 7.2 快速入门

### 7.2.1 初始化一个AipOcr对象

AipOcr类提供给开发一系列的图像识别方法，参考如下代码新建一个AipOcr对象：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)
```

在上面代码中，常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是由系统分配给用户的，均为字符串，用于标识用户，为访问服务做签名验证。获取方式请参考[获取AK/SK](#)。

### 7.2.2 配置AipOcr

如果用户需要配置AipOcr的一些细节参数，可以在构造AipOcr之后调用接口设置参数，目前只支持以下参数：

| 接口                           | 说明                      |
|------------------------------|-------------------------|
| setConnectionTimeoutInMillis | 建立连接的超时时间（单位：毫秒）        |
| setSocketTimeoutInMillis     | 通过打开的连接传输数据的超时时间（单位：毫秒） |

## 7.3 错误信息格式

若请求错误，服务器将返回的JSON文本包含以下参数：

- error\_code：错误码；关于错误码的详细信息请参考通用错误码和业务相关错误码。
- error\_msg：错误描述信息，帮助理解 and 解决发生的错误。

SDK本地检测参数返回的错误码：



| error_code | error_msg                        | 备注          |
|------------|----------------------------------|-------------|
| SDK100     | image size error                 | 图片大小超限      |
| SDK101     | image length error               | 图片边长不符合要求   |
| SDK102     | read image file error            | 读取图片文件错误    |
| SDK108     | connection or read data time out | 连接超时或读取数据超时 |
| SDK109     | unsupported image format         | 不支持的图片格式    |

#### 服务端返回的错误码

| 错误码    | 错误信息                         | 描述            |
|--------|------------------------------|---------------|
| 216015 | module closed                | 模块关闭          |
| 216100 | invalid param                | 非法参数          |
| 216101 | not enough param             | 参数数量不够        |
| 216102 | service not support          | 业务不支持         |
| 216103 | param too long               | 参数太长          |
| 216110 | appid not exist              | APP ID不存在     |
| 216111 | invalid userid               | 非法用户ID        |
| 216200 | empty image                  | 空的图片          |
| 216201 | image format error           | 图片格式错误        |
| 216202 | image size error             | 图片大小错误        |
| 216300 | db error                     | DB错误          |
| 216400 | backend error                | 后端系统错误        |
| 216401 | internal error               | 内部错误          |
| 216500 | unknown error                | 未知错误          |
| 216600 | id number format error       | 身份证的ID格式错误    |
| 216601 | id number and name not match | 身份证的ID和名字不匹配  |
| 216630 | recognize error              | 识别错误          |
| 216631 | recognize bank card error    | 识别银行卡错误       |
| 216632 | ocr                          | unknown error |
| 216633 | recognize idcard error       | 识别身份证错误       |
| 216634 | detect error                 | 检测错误          |
| 216635 | get mask error               | 获取mask图片错误    |

| 错误码    | 错误信息                        | 描述                      |
|--------|-----------------------------|-------------------------|
| 282000 | logic internal error        | 业务逻辑层内部错误               |
| 282001 | logic backend error         | 业务逻辑层后端服务错误             |
| 282002 | input encoding error        | 请求参数编码错误                |
| 282100 | image transcode error       | 图片压缩转码错误                |
| 282807 | image too long              | 图片超过4M                  |
| 282808 | request id: xxxxx not exist | request id xxxxx 不存在    |
| 282809 | result type error           | 返回结果请求错误（不属于excel或json） |
| 282810 | image recognize error       | 图像识别错误                  |

## 7.4 通用文字识别

通用文字识别可以识别出图片中的文字。

举例，对一张图片进行文字识别，示例代码如下：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 调用通用文字识别接口
result = aipOcr.basicGeneral(get_file_content('general.jpg'))
```

传入图片时还想增加一些自定义参数配置：

```
# 引入文字识别OCR SDK
```

```
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 定义参数变量
options = {
    'detect_direction': 'true',
    'language_type': 'CHN_ENG',
}

# 调用通用文字识别接口
result = aipOcr.basicGeneral(get_file_content('general.jpg'), options)
```

#### 通用文字识别 [请求参数详情](#)

| 参数    | 是否必选  | 类型     | 可选值范围 | 说明  |
|-------|-------|--------|-------|---|
| image | true  | string | -     | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 4M，最短边至少 15px，最长边最大 4096px，支持 jpg/ png/ bmp 格式 |
| mask  | false | string | -     | 表示mask区域的黑白灰度图片，白色代表选中，base64编码   |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

通用文字识别 返回数据参数详情

| 字段               | 必选 | 类型      | 说明  |
|------------------|----|---------|---|
| direction        | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result_num | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| words_result     | 是  | array() | 定位和识别结果数组   |
| +words           | 否  | string  | 识别结果字符串   |

返回样例：

```
{
  "log_id": 2216743163,
  "words_result": [
    {
      "words": "众里寻他千百度。蓦然回首,那人却在,灯火阑珊处。"
    }
  ],
}
```

```
"words_result_num": 1
}
```

## 7.5 通用文字识别（含位置信息版）

通用文字识别（含位置信息版）可以接受任意图片，并识别出图片中的文字以及全部文字串，以及字符在图片中的位置信息。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 调用通用文字识别接口
result = aipOcr.general(get_file_content('general.jpg'))
```

传入图片时还想增加一些自定义参数配置：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
```

```
with open(filePath, 'rb') as fp:
    return fp.read()

# 初始化ApiOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 定义参数变量
options = {
    'detect_direction': 'true',
    'language_type': 'CHN_ENG',
}

# 调用通用文字识别接口
result = aipOcr.general(get_file_content('general.jpg'), options)
```

[通用文字识别（含位置信息版）](#) [请求参数详情](#)

| 参数    | 是否必选  | 类型     | 可选值范围 | 说明   |
|-------|-------|--------|-------|--|
| image | true  | string | -     | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |
| mask  | false | string | -     | 表示mask区域的 黑白灰度图 片，白色代 表 选中, base64编 码  |

| 参数                | 是否必选  | 类型      | 可选值范围   | 说明  |
|-------------------|-------|---------|---|---|
| language_type     | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction  | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language   | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |
| vertexes_location | false | boolean | true、false  | 是否在返回结果中标识文字位置  |



| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

## 通用文字识别（含位置信息版） 返回数据参数详情

| 字段                 | 必选 | 类型      | 说明  |
|--------------------|----|---------|---|
| direction          | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id             | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result       | 是  | array() | 定位和识别结果数组   |
| words_result_num   | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| +vertexes_location | 否  | array() | 当前为四个顶点：左上，右上，右下，左下。当vertexes_location=true时存在                                    |
| ++x                | 是  | uint32  | 水平坐标（坐标0点为左上角）  |
| ++y                | 是  | uint32  | 垂直坐标（坐标0点为左上角）  |
| +location          | 是  | array() | 位置数组（坐标0点为左上角）  |

| 字段         | 必选 | 类型      | 说明                                     |
|------------|----|---------|--|
| ++left     | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                    |
| ++top      | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                    |
| ++width    | 是  | uint32  | 表示定位位置的长方形的宽度                          |
| ++height   | 是  | uint32  | 表示定位位置的长方形的高度                          |
| +words     | 否  | string  | 识别结果字符串                                |
| +chars     | 否  | array() | 单字符结果, recognize_granularity=small 时存在 |
| ++location | 是  | array() | 位置数组(坐标0点为左上角)                         |
| +++left    | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                    |
| +++top     | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                    |
| +++width   | 是  | uint32  | 表示定位位置的长方形的宽度                          |
| +++height  | 是  | uint32  | 表示位置的长方形的高度                            |
| ++char     | 是  | string  | 单字符识别结果                                |

## 7.6 通用文字识别(含生僻字版)

某些场景中, 图片中的中文不光有常用字, 还包含了生僻字, 这时用户需要对该图进行文字识别, 应使用通用文字识别(含生僻字版)。

图片接受参数类型: 支持本地图片路径字符串, 图片文件二进制数组。举例, 要对一张网络图片进行文字识别, 具体的文字的内容和信息在返回的words\_result字段中:

```
# 引入文字识别OCR SDK
```

```
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 调用生僻字识别接口
result = aipOcr.enhancedGeneral(get_file_content('enhanced_general.jpg'))
```

传入图片时还想增加一些自定义参数配置：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 定义参数变量
options = {
    'detect_direction': 'true',
    'language_type': 'CHN_ENG',
}

# 调用生僻字识别接口
result = aipOcr.enhancedGeneral(get_file_content('enhanced_general.jpg'), options)
```

[通用文字识别（含生僻字版）请求参数详情](#)

| 参数               | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|------------------|-------|---------|---|---|
| image            | true  | string  | -                                       | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 4M，最短边至少 15px，最长边最大 4096px, 支持 jpg/ png/ bmp 格式  |
| language_type    | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识 别 语 言 类 型，默 认 为 CHN_ENG。可 选 值 包 括：- CHN_ENG：中 英 文 混 合；- ENG：英 文；- POR：葡 萄 牙 语；- FRE：法 语；- GER：德 语；- ITA：意 大 利 语；- SPA：西 班 牙 语；- RUS：俄 语；- JAP：日 语 |
| detect_direction | false | boolean | true、false                              | 是 否 检 测 图 像 朝 向，默 认 不 检 测，即：false。朝 向 是 指 输 入 图 像 是 正 常 方 向、逆 时 针 旋 转 90/180/270 度。可 选 值 包 括：- true：检 测 朝 向；- false：不 检 测 朝 向。                      |

[通用文字识别（含生僻字版） 返回数据参数详情](#)

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图 像 方 向， 当 de-<br>tect_direction=true<br>时 存 在。 - -1: 未 定<br>义， - 0: 正 向， - 1:<br>逆 时 针 90 度， - 2: 逆<br>时 针 180 度， - 3: 逆<br>时 针 270 度 |
| log_id           | 是  | uint64  | 唯一的log id， 用于<br>问题定位  |
| words_result_num | 是  | uint32  | 识别结果数， 表示<br>words_result的元素<br>个数   |
| words_result     | 是  | array() | 定位和识别结果数<br>组  |
| +words           | 否  | string  | 识别结果字符串  |

## 7.7 网络图片文字识别

网络图片文字识别用于识别一些网络上背景复杂，特殊字体的文字。

图片接受参数类型：支持本地图片路径字符串，图片文件二进制数组。

举例，要对一张网络图片进行文字识别，具体的文字的内容和信息在返回的words\_result字段中：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)
```

```
# 网络图片文字识别接口
result = aipOcr.webImage(get_file_content('web_image.jpg'))
```

传入图片时还想增加一些自定义参数配置：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 定义参数变量
options = {
    'detect_direction': 'true',
    'language_type': 'CHN_ENG',
}

# 网络图片文字识别接口
result = aipOcr.webImage(get_file_content('web_image.jpg'), options)
```

[网络图片文字识别 请求参数详情](#)

| 参数    | 是否必选 | 类型     | 可选值范围 | 说明  |
|-------|------|--------|-------|---|
| image | true | string | -     | 图 像 数 据，base64 编 码，要求base64编码后大小不超过 4M，最短边至少15px，最长边最大4096px，支持jpg/ png/ bmp 格式 |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

[网络图片文字识别 返回数据参数详情](#)

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| words_result     | 是  | array() | 定位和识别结果数组  |
| +words           | 否  | string  | 识别结果字符串  |

## 7.8 银行卡识别

银行卡识别能识别出图片中对应的银行卡号。

举例，对一张银行卡进行识别，示例代码如下：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 调用银行卡识别接口
result = aipOcr.bankcard(get_file_content('bankcard.jpg'))
```



银行卡识别 [请求参数详情](#)

| 参数    | 类型     | 描述            | 是否必须 |
|-------|--------|---------------|------|
| image | String | 图像数据，仅支持图像文件流 | 是    |

银行卡识别 [返回数据参数详情](#)

| 参数                 | 类型     | 描述                           |
|--------------------|--------|------------------------------|
| log_id             | Unit64 | 唯一的log id，用于问题定位             |
| result             | Object | 定位和识别结果数组                    |
| \+bank_card_number | String | 银行卡识别结果                      |
| +bank_name         | string | 银行名，不能识别时为空                  |
| +bank_card_type    | uint32 | 银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡 |

返回样例：

```
// 银行卡识别会对银行卡号位数校验，即16位数字
// 识别或者校验错误则返回  recognize bank card error
// bank_card_number 即为银行卡号
{
  "log_id": 1447188951,
  "result": {
    "bank_card_number": "6225000000000000",
    "bank_name": "招商银行",
    "bank_card_type": 1
  }
}
```

## 7.9 身份证识别

身份证识别一次只能接受身份证正面或反面的清晰图片，能识别出证件上的文字。

举例，对一张身份证正面进行文字识别，示例代码如下：

```
# 引入文字识别OCR SDK
from aip import AipOcr
```

```
# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 设置识别身份证正面参数
isFront = False

# 调用身份证识别接口
result = aipOcr.idcard(get_file_content('idcard.jpg'), isFront)
```

传入图片时还想增加一些自定义参数配置：

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 设置识别身份证正面参数
isFront = False

# 定义参数变量
options = {
    'detect_direction': False,
    'accuracy': 'high'
}
```

```
}

# 调用身份证识别接口
result = aipOcr.idcard(get_file_content('idcard.jpg'), isFront, options)
```

身份证识别 [请求参数详情](#)

| 参数               | 类型      | 描述   | 是否必须 |
|------------------|---------|--|------|
| detect_direction | Boolean | 检测图像朝向(指输入图像是正常方向、逆时针旋转90/180/270度),有效值: true、false, 默认值: false。 | 否    |
| id_card_side     | String  | front: 身份证正面, back: 身份证背面  | 是    |
| image            | String  | 图像数据, 仅支持图像文件流   | 是    |

身份证识别 [返回数据参数详情](#)

| 参数               | 类型     | 描述  |
|------------------|--------|---|
| direction        | Int32  | 图像方向, 当 detect_direction=true 时存在。-1:未定义, 0:正向, 1: 逆时针90度, 2:逆时针180度, 3:逆时针270度       |
| log_id           | Unit64 | 唯一的log id, 用于问题定位   |
| words_result     | Array  | 定位和识别结果数组, 数组元素的key是身份证的主体字段(正面支持: 住址、公民身份号码、出生、姓名、性别、民族, 背面支持: 签发日期、失效日期)。只返回识别出的字段。 |
| words_result_num | Unit32 | 识别结果数, 表示 words_result 的元素个数  |

| 参数         | 类型     | 描述                  |
|------------|--------|---------------------|
| \+location | Array  | 位置数组（坐标0点为左上角）      |
| \+\+left   | Unit32 | 表示定位位置的长方形左上顶点的水平坐标 |
| \+\+top    | Unit32 | 表示定位位置的长方形左上顶点的垂直坐标 |
| \+\+width  | Unit32 | 表示定位位置的长方形的宽度       |
| \+\+height | Unit32 | 表示定位位置的长方形的高度       |
| \+words    | String | 识别结果字符串             |

#### 返回样例：

```
// 身份证识别对身份证号码进行校验，如果错误则返回空身份证号
{
  "log_id": 3043122372,
  "words_result": {
    "住址": {
      "location": {
        "left": 93,
        "top": 161,
        "width": 201,
        "height": 51
      },
      "words": "北京市东城区景山前街4紫禁城敬事房"
    },
    "公民身份号码": {
      "location": {
        "left": 97,
        "top": 200,
        "width": 202,
        "height": 27
      },
      "words": "110114198103214083"
    },
    "出生": {
      "location": {
        "left": 97,
        "top": 121,
```

```
        "width": 161,
        "height": 17
    },
    "words": "1654年12月20日"
},
"姓名": {
    "location": {
        "left": 99,
        "top": 40,
        "width": 64,
        "height": 22
    },
    "words": "韦小宝"
},
"性别": {
    "location": {
        "left": 100,
        "top": 83,
        "width": 15,
        "height": 18
    },
    "words": "男"
},
"民族": {
    "location": {
        "left": 197,
        "top": 84,
        "width": 14,
        "height": 17
    },
    "words": "汉"
}
},
"words_result_num": 6
}
```

## 7.10 行驶证识别

### 代码示例

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
```

```

APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
aipOcr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 调用行驶证识别接口
result = aipOcr.vehicleLicense(get_file_content('vehicleLicense.jpg'))

```

#### 请求参数

| 参数               | 类型     | 描述  | 是否必须 |
|------------------|--------|---|------|
| detect_direction | String | 是否检测图像朝向，默认不检测，即：false。可选值包括true - 检测朝向；false - 不检测朝向。朝向是指输入图像是正常方向、逆时针旋转90/180/270度。 | 否    |
| image            | String | 图像数据  | 是    |
| accuracy         | string | normal 使用快速服务;缺省或其它值使用高精度服务   | 否    |

#### 返回参数

| 参数               | 类型     | 描述                        |
|------------------|--------|---------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位          |
| words_result_num | Uint32 | 识别结果数，表示words_result的元素个数 |
| words_result     | Array  | 定位和识别结果数组                 |

| 参数         | 类型     | 描述                  |
|------------|--------|---------------------|
| \+location | Array  | 位置数组（坐标0点为左上角）      |
| \+\+left   | Uint32 | 表示定位位置的长方形左上顶点的水平坐标 |
| \+\+top    | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标 |
| \+\+width  | Uint32 | 表示定位位置的长方形的宽度       |
| \+\+height | Uint32 | 表示定位位置的长方形的高度       |
| \+words    | String | 识别结果字符串             |

返回示例：

```
{
  "log_id": 2842358861,
  "words_result_num": 10,
  "words_result": {
    "品牌型号": {
      "location": {
        "width": 140,
        "top": 249,
        "height": 25,
        "left": 258
      },
      "words": "兰博基尼NB"
    },
    "发证日期": {
      "location": {
        "width": 77,
        "top": 337,
        "height": 24,
        "left": 342
      },
      "words": "20130701"
    },
    "使用性质": {
      "location": {
        "width": 48,
        "top": 249,
```

```
        "height": 23,
        "left": 130
    },
    "words": "炫富"
},
"发动机号码": {
    "location": {
        "width": 53,
        "top": 309,
        "height": 19,
        "left": 235
    },
    "words": "888888"
},
"号牌号码": {
    "location": {
        "width": 66,
        "top": 163,
        "height": 15,
        "left": 129
    },
    "words": "LBJNDN"
},
"所有人": {
    "location": {
        "width": 49,
        "top": 190,
        "height": 19,
        "left": 130
    },
    "words": "小DU熊"
},
"住址": {
    "location": {
        "width": 280,
        "top": 220,
        "height": 20,
        "left": 130
    },
    "words": "北京市063号"
},
"注册日期": {
    "location": {
        "width": 76,
        "top": 334,
```



```
        "height": 26,
        "left": 212
    },
    "words": "30130701"
},
"车辆识别代号": {
    "location": {
        "width": 196,
        "top": 279,
        "height": 21,
        "left": 248
    },
    "words": "44444444"
},
"车辆类型": {
    "location": {
        "width": 61,
        "top": 159,
        "height": 24,
        "left": 292
    },
    "words": "豪华"
}
}
```

## 7.11 驾驶证识别

### 代码示例

```
# 引入文字识别OCR SDK
from aip import AipOcr

# 定义常量
APP_ID = '你的 App ID'
API_KEY = '你的 API Key'
SECRET_KEY = '你的 Secret Key'

# 读取图片
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

# 初始化AipOcr对象
```

```
aip0cr = AipOcr(APP_ID, API_KEY, SECRET_KEY)

# 调用驾驶证识别接口
result = aip0cr.drivingLicense(get_file_content('drivingLicense.jpg'))
```

请求参数

| 参数 | 类型 | 描述 | 是否必须 | | :----- | :-- | :----- | :  
----- | | image | String | 图像数据 | 是 | 返回参数

| 参数               | 类型     | 描述                          |
|------------------|--------|-----------------------------|
| log_id           | Uint64 | 唯一的log id，用于问题定位            |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数 |
| words_result     | Array  | 定位和识别结果数组                   |
| \+location       | Array  | 位置数组（坐标0点为左上角）              |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标         |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标         |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度               |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度               |
| \+words          | String | 识别结果字符串                     |

返回示例：

```
{
  "log_id": 2785054721,
  "words_result_num": 10,
  "words_result": {
    "证号": {
      "location": {
        "width": 181,
        "top": 168,
        "height": 22,
        "left": 182
```

```
    },
    "words": "666666666666"
  },
  "有效期限": {
    "location": {
      "width": 80,
      "top": 357,
      "height": 26,
      "left": 105
    },
    "words": "8888年13月1日"
  },
  "准驾车型": {
    "location": {
      "width": 40,
      "top": 344,
      "height": 13,
      "left": 223
    },
    "words": "A0"
  },
  "住址": {
    "location": {
      "width": 342,
      "top": 223,
      "height": 47,
      "left": 65
    },
    "words": "北京市073号"
  },
  "至": {
    "location": {
      "width": 83,
      "top": 368,
      "height": 21,
      "left": 223
    },
    "words": "1997年7月1日"
  },
  "姓名": {
    "location": {
      "width": 88,
      "top": 194,
      "height": 25,
      "left": 69
```

```

        },
        "words": "小\DU熊"
    },
    "国籍": {
        "location": {
            "width": 35,
            "top": 196,
            "height": 26,
            "left": 334
        },
        "words": "中国"
    },
    "出生日期": {
        "location": {
            "width": 91,
            "top": 280,
            "height": 24,
            "left": 185
        },
        "words": "19851218"
    },
    "性别": {
        "location": {
            "width": 17,
            "top": 196,
            "height": 24,
            "left": 248
        },
        "words": "男"
    },
    "初次领证日期": {
        "location": {
            "width": 83,
            "top": 310,
            "height": 23,
            "left": 205
        },
        "words": "20081206"
    }
}
}

```

## 7.12 表格识别请求

代码示例

```
aipOcr.tableRecognitionAsync(get_file_content('table.jpg'))
```

请求参数

| 参数    | 类型     | 描述   | 是否必须 |
|-------|--------|------|------|
| image | String | 图像数据 | 是    |

返回参数

| 参数名称        | 数据类型   | 说明   |
|-------------|--------|--|
| result      | list   | 返回的结果  |
| +request_id | string | 该请求生成的request_id，<br>后续使用该request_id获取<br>识别结果 |
| log_id      | string | 唯一的log id，排错用                                  |

返回示例：

```
{
  "result" : [
    {
      "request_id" : "1234_6789"
    }
  ],
  "log_id":149689853984104
}
```

### 7.13 表格识别结果获取

代码示例

```
aipOcr.getTableRecognitionResult('9041866_24', {
  'result_type': 'json',
})
```

请求参数

| 参数名称        | 数据类型   | 是否必须 | 备注   | 样例                |
|-------------|--------|------|--|-------------------|
| request_id  | string | 是    | 发送图片请求时返回的 request id  | '149691317905102' |
| result_type | string | 否    | 期望获取结果的类型, 取值为“excel”时返回xls文件的地址, 取值为“json”时返回json格式的字符串, 默认为“excel” | “excel”           |

### 返回参数

| 参数名称         | 数据类型   | 说明   |
|--------------|--------|--|
| result       |        | 返回的结果  |
| +result_data | string | 识别结果字符串, 如果request_type是excel, 则返回excel的文件下载地址, 如果request_type是json, 则返回json格式的字符串 |
| +percent     | int    | 图片识别进度(百分比)  |
| +request_id  | string | 该图片对应请求的 request_id  |
| +ret_code    | int    | 识别状态, 1: 任务未开始, 2: 进行中, 3: 已完成   |
| +ret_msg     | string | 图片的状态信息  |
| log_id       | long   | 唯一的log id, 排错用   |

当request\_type为json时, result\_data格式样例为:

```
{
  "form_num": 1,
  "forms": [
    {
      "header": [
        {
```

```
        "row": [
            1
        ],
        "column": [
            1,
            2
        ],
        "word": "表头信息1",
    }
],
"footer": [
    {
        "row": [
            1
        ],
        "column": [
            1,
            2
        ],
        "word": "表尾信息1",
    }
],
"body": [
    {
        "row": [
            1
        ],
        "column": [
            1,
            2
        ],
        "word": "单元格文字",
    }
]
}
```

当request\_type为excel时，result\_data为url字符串：

"https://ai.baidu.com/file/xxxxffddd"

失败应答示例：

```
{
  "log_id": 149319909347709,
  "error_code": 282000
  "error_msg": "unknown error"
}
```

## 7.14 表格识别轮询接口

### 代码示例

调用表格识别请求，获取请求id之后轮询调用表格识别获取结果的接口

```
aipOcr.tableRecognition(
  get_file_content('table.jpg'),
  {
    'result_type': 'json',
  }
)
```

### 返回参数

与表格识别结果接口返回相同



## 第8章

# OCR iOS SDK 开发者文档

### 8.1 简介

本文档主要介绍OCR iOS SDK的安装和使用。在使用本文档前，您需要先了解Optical Character Recognition(OCR)的基础知识，并已经开通了OCR服务。

支持的系统和硬件版本

- iOS: 8.0 以上
- 架构: i386 x86\_64 armv7 armv7s arm64

\*新增 IdcardQuality.framework 支持的系统和硬件版本

- iOS: 8.0 以上
- 架构: armv7 armv7s arm64
- 如果开发者想要在模拟器中集成IdcardQuality，可以使用 宏定义屏蔽相关代码

```
## if !TARGET_IPHONE_SIMULATOR
    IdcardQualityAdaptor *idcard = [[IdcardQualityAdaptor alloc] init];
    [idcard initWithToken:token];

## endif
```

### 8.2 快速入门

#### 8.2.1 SDK工程结构

AipOcrSdk // OCR Framework 工程，包含相机相关操作

```
|-AipBase.framework // 身份验证、OCR基础操作框架包
|-IdcardQuality.framework // 身份证质量控制框架包
|-AipOcrSdk
|-IOSeScaleLayout...
|-AipOcrService.h/m // OCR服务包装类
|-AipOcrResultDelegate.h/m
|-View // 界面相关类
|-ViewController
|-AipOcrBundle
AipOcrDemo // Demo 工程
|-AipOcrDemo
```

AipOcrDemo工程依赖了AipOcrSdk工程以及AipBase.framework、IdcardQuality.framework框架。（Dynamic框架，请使用Embed Binary方式嵌入）

其中

- AipBase.framework中包含了OCR服务相关基础类，必须引入。
- IdcardQuality.framework包含了身份证质量控制的功能，按需引入。
- AipOcrSdk工程包含了相机相关操作，如果需要相机接口，需要引入
- 如果只需直接调用相机接口，直接引入AipOcrSdk工程生成的AipOcrSdk.framework即可。
- 如果需要修改相机界面和逻辑，可像demo工程一样引入AipOcrSdk工程，添加为依赖。

### 8.2.2 Demo工程

iOS SDK提供了一个可快速运行的Demo工程，建议首先运行一下Demo工程，其中包含了使用SDK的主要步骤。

运行步骤如下：

1. 在[官网](#)下载iOS SDK 压缩包
2. 解压缩，双击打开 AipOcr.xcworkspace
3. 在[管理控制台](#)中新建文字识别应用，配置BundleId为AipOcrDemo的BundleId(默认为com.baidu.AipOcrDemo)
4. 在AipOcrDemo工程中AipOcrDemo/ViewController.m viewDidLoad方法中配置相应[管理控制台](#)中新建的应用的Api Key, Secret Key
5. Build AipOcrSdk, 再运行AipOcrDemo > 若提示 AipOcrSdk/AipOcrSdk.h 不存在，请先 Build AipOcrSdk 工程 > 若运行提示“身份验证错误”，请确认填写了步骤4中的Api Key/Secret Key

### 8.2.3 身份验证与安全

百度AIP开放平台使用OAuth2.0授权调用开放API，调用API时必须在URL中带上access\_token参数。AccessToken可用AK/SK或者授权文件的方式获得。

OCR iOS SDK提供了以下3种AccessToken管理方法。

**API Key / Secret Key** 此种身份验证方案使用AK/SK获得AccessToken，缓存在本地。

虽然SDK对网络传输的敏感数据进行了二次加密，但由于AK/SK是明文填写在代码中，在移动设备中可能会存在AK/SK被盗取的风险。有安全考虑的开发者可使用第二种授权方案。

使用步骤：

1. 在[管理控制台](#)中配置OCR应用
2. 复制应用的Api Key（简称AK）和 Secret Key（简称SK），初始化AipOcrService单例：

```
// 接口
- (void) authWithAK: (NSString *)ak andSK: (NSString *)sk;
// 示例
[[AipOcrService shardService] authWithAK:@"Api Key" andSK:@"Secret Key"];
```

**授权文件（安全模式）** 此种身份验证方案使用授权文件获得AccessToken，缓存在本地。建议有安全考虑的开发者使用此种身份验证方式。

在您的移动APP分发出去之后，APP存在被反编译的可能，所以直接将AK / SK 置于APP源码之中，存在被盗取的风险。采用授权文件的身份验证方法，可有效保护AK/SK在移动设备中的安全。攻击者即使拦截了流量，盗取了授权文件，也难以盗用您的配额。

使用步骤：

1. 在[官网](#)中配置应用
2. 下载对应应用的授权文件（默认名字为aip.license）
3. 将授权文件添加至XCode工程（配置为资源并拷贝，Target -> Build Phases -> Copy Bundle Resource 中添加该文件）
4. 读取授权文件原始字节，NSData格式，初始化AipOcrService单例：

```
// 接口
- (void) authWithLicenseFileData: (NSData *)licenseFileContent;

// 示例
```

```
// 若未添加至主工程, 则[NSBundle mainBundle]修改为对应bundle
NSString *licenseFile = [[NSBundle mainBundle] pathForResource:@"aip" ofType:@"license"];
NSData *licenseFileData = [NSData dataWithContentsOfFile:licenseFile];
[[AipOcrService shardService] authWithLicenseFileData:licenseFileData];
```

**自助AccessToken管理** 此种身份验证方案直接使用开发者提供的AccessToken, 不做缓存。

若开发者的应用有自行搭建的服务端, 也可在自己的服务端进行token的获取与管理, 然后分配给移动客户端使用。Token的获得可以参考[API鉴权认证机制](#)。此种授权方案在移动客户端上没有任何AK/SK信息, 风险系数低, 但需要开发者自行管理Token的获取与分配, 适合有条件的开发者使用。

使用步骤:

1. 将您需要使用的token传入对应的构造函数, 初始化[AipOcrService](#)单例:

```
// 接口
- (void) authWithToken: (NSString *)token;
// 示例
[[AipOcrService shardService] authWithToken:@"Token here"];
```

注意: 此种方式下, SDK不再缓存token, 开发者需自行处理token过期问题。

## 8.3 SDK集成与接口调用

使用SDK时, 需要将AipBase.framework和AipOcrSdk.framework嵌入工程 (Embed Binary 方式, 可参考AipOcrDemo工程), 并导入相关头文件。

在调用接口前, 请确认已经进行过身份验证。见[身份验证与安全](#)。

### 8.3.1 相机接口

该调用方法将会自动打开相机, 用户拍照、确认, 完成识别之后, 使用delegate直接返回识别结果。

内部使用了[\[AipOcrService shardService\]](#)进行数据调用。

具体返回格式见 [数据接口](#)

操作步骤:

1. 身份验证：调用 `[[AipOcrService shardService] authWithAK:SK]` 或其他验证方法；
2. 在需要回调的类中实现AipOcrDelegate的接口（按需实现即可）。可在这些方法中pop出OCR的View,具体参考Demo工程。

- 通用文字识别成功：- `(void) ocrOnGeneralSuccessful:(id)result`
- 银行卡识别成功：- `(void) ocrOnBankCardSuccessful:(id)result`
- 身份证识别成功：- `(void) ocrOnIdCardSuccessful:(id)result`
- 识别失败：- `(void) ocrOnFail:(NSError *)error`

3. 在工程Info.plist中添加NSCameraUsageDescription，相机权限说明

4. 初始化ViewController，其中

- 通用文字识别： `UIViewController * vc = [AipGeneralVC ViewControllerWithDelegate:self]`
- 银行卡识别 `UIViewController * vc = [AipCaptureCardVC ViewControllerWithCardType:CardTypeBankCard andDelegate:self]`
- 身份证识别
- 正面: `UIViewController * vc = [AipCaptureCardVC ViewControllerWithCardType:CardTypeIdCardFront andDelegate:self]`
- 背面: `UIViewController * vc = [AipCaptureCardVC ViewControllerWithCardType:CardTypeIdCardBack andDelegate:self]`

5. 在合适的地方启动ViewController: 如 `[self presentViewController:vc animated:YES completion:nil]`

AipOcrSdk.framework，即AipOcr工程包含了相机相关操作代码，如开发者想直接使用，无需引用AipOcrSdk工程，直接引入AipOcrSdk.framework即可。若需要修改相机操作相关UI，可引入AipOcrSdk工程，编辑相关UI代码即可。

### 8.3.2 数据接口

该调用方法传入需要识别的UIImage，异步识别，识别完成之后，回调返回识别结果。

主要类为AipOcrService类，使用单例`[[AipOcrService sharedService]]`来调用相关接口即可。

操作步骤：

1. 身份验证：调用 `[[AipOcrService shardService] authWithAK:SK]` 或其他验证方法；
2. 调用以下相应接口

- 通用文字识别 `detectTextFromImage`

- 通用文字识别（不带位置信息）[detectTextBasicFromImage](#)
- 通用文字识别（含生僻字）[detectTextEnhancedFromImage](#)
- 网图识别 [detectWebImageFromImage](#)
- 身份证正面[detectIdCardFrontFromImage](#)
- 身份证背面[detectIdCardBackFromImage](#)
- 银行卡[detectBankCardFromImage](#)

所有回调函数均在后台线程中被调用，如需在主线程中操作，请使用[[NSOperationQueue mainQueue] addOperationWithBlock]patch到主线程中，示例参考demo工程。

通用文字识别

- 调用示例

```
NSDictionary *options = @{@"language_type": @"CHN_ENG", @"detect_direction": @"true"};
[[AipOcrService shardService] detectTextFromImage:finalImage withOptions:options successHandler:^(i
// 成功识别的后续逻辑
} failHandler:^(NSError *err) {
// 失败的后续逻辑
}];
```

[options参数详情](#)

| 参数                    | 是否必选  | 类型     | 可选值范围     | 说明  |
|-----------------------|-------|--------|-----------|---|
| image（已由参数替代）         | true  | string | -         | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 1M，最短边至少 15px，最长边最大 2048px，支持 jpg/ png/ bmp 格式 |
| recognize_granularity | false | string | big、small | 是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置  |

| 参数               | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|------------------|-------|---------|---|---|
| mask             | false | string  | -                                       | 表示mask区域的黑白灰度图片，白色代表选中, base64编码  |
| language_type    | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false                              | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | false | string  | true、false                              | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选  | 类型     | 可选值范围      | 说明  |
|--------------------|-------|--------|------------|---|
| classify_dimension | false | string | lottery    | 分类维度（根据 OCR 结果进行分类），逗号分隔，当前只支持 lottery。lottery：彩票分类，设置 detect_direction 有助于提升精度 |
| vertexes_location  | false | string | true、false | 是否返回文字外接多边形顶点位置，不支持单字位置。默认为false  |

• 结果返回

| 字段                 | 必选 | 类型      | 说明   |
|--------------------|----|---------|--|
| direction          | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id             | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result       | 是  | array() | 定位和识别结果数组  |
| words_result_num   | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| +vertexes_location | 否  | array() | 当前为四个顶点: 左上，右上，右下，左下。当 vertexes_location=true 时存在  |
| ++x                | 是  | uint32  | 水平坐标（坐标0点为左上角）   |



| 字段         | 必选 | 类型      | 说明                                   |
|------------|----|---------|--------------------------------------|
| ++y        | 是  | uint32  | 垂直坐标（坐标0点为左上角）                       |
| +location  | 是  | array() | 位置数组（坐标0点为左上角）                       |
| ++left     | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| ++top      | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| ++width    | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| ++height   | 是  | uint32  | 表示定位位置的长方形的高度                        |
| +words     | 否  | string  | 识别结果字符串                              |
| +chars     | 否  | array() | 单字符结果，recognize_granularity=small时存在 |
| ++location | 是  | array() | 位置数组（坐标0点为左上角）                       |
| +++left    | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                  |
| +++top     | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                  |
| +++width   | 是  | uint32  | 表示定位位置的长方形的宽度                        |
| +++height  | 是  | uint32  | 表示位置的长方形的高度                          |
| ++char     | 是  | string  | 单字符识别结果                              |

// 示例

```
{
  direction : 2,
  log_id : 676709620,
  words_result : [ {
```

```
        location : {
            height : 20;
            left : 86;
            top : 387;
            width : 22;
        };
        words : "N";
    },
],
words_result_num : 1;
}
```

通用文字识别（不带位置信息版）

- 调用示例

```
NSDictionary *options = @{@"language_type": @"CHN_ENG", @"detect_direction": @"true"};
[[AipOcrService shardService] detectTextBasicFromImage:finalImage withOptions:options successHandle
// 成功识别的后续逻辑
} failHandler:^(NSError *err) {
// 失败的后续逻辑
}];
```

options参数详情

| 参数                    | 是否必选  | 类型     | 可选值范围     | 说明  |
|-----------------------|-------|--------|-----------|---|
| image（已由参数替代）         | true  | string | -         | 图 像 数 据，base64 编 码，要求base64编码后大小不超过 1M，最短边至少 15px，最长边最大 2048px, 支持 jpg/ png/ bmp 格式 |
| recognize_granularity | false | string | big、small | 是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置  |

| 参数               | 是否必选  | 类型      | 可选值范围                                   | 说明  |
|------------------|-------|---------|---|---|
| mask             | false | string  | -                                       | 表示mask区域的黑白灰度图片，白色代表选中, base64编码  |
| language_type    | false | string  | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false                              | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | false | string  | true、false                              | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选  | 类型     | 可选值范围   | 说明   |
|--------------------|-------|--------|---------|--|
| classify_dimension | false | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

• 结果返回

| 字段               | 必选 | 类型      | 说明  |
|------------------|----|---------|---|
| direction        | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result     | 是  | array() | 定位和识别结果数组   |
| words_result_num | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| +words           | 否  | string  | 识别结果字符串   |

```
// 示例
{
  direction : 2,
  log_id : 676709620,
  words_result : [ {
    words : "N",
  },
],
```

```

        words_result_num : 1,
    }

```

### 通用文字识别（带生僻字版）

- 调用示例

```

NSMutableDictionary *options = @{@"language_type": @"CHN_ENG", @"detect_direction": @"true"};
[[AipOcrService shardService] detectTextEnhancedFromImage:finalImage withOptions:options successHan
// 成功识别的后续逻辑
} failHandler:^(NSError *err) {
// 失败的后续逻辑
}];

```

#### options参数详情

与通用文字识别（不带位置信息版）一致

- 结果返回

与通用文字识别（不带位置信息版）一致

```

// 示例
{
    direction : 2,
    log_id : 676709620,
    words_result : [ {
        words : "N",
    },
    ],
    words_result_num : 1,
}

```

### 网图识别

- 调用示例

```

NSMutableDictionary *options = @{@"language_type": @"CHN_ENG", @"detect_direction": @"true"};
[[AipOcrService shardService] detectWebImageFromImage:finalImage withOptions:options successHandler
// 成功识别的后续逻辑

```

```
} failHandler:^(NSError *err) {  
    // 失败的后续逻辑  
}];
```

options参数详情

与通用文字识别（不带位置信息版）一致

- 结果返回

与通用文字识别（不带位置信息版）一致

```
// 示例  
{  
    direction : 2,  
    log_id : 676709620,  
    words_result : [ {  
        words : "N",  
    },  
    ],  
    words_result_num : 1,  
}
```

银行卡识别

- 调用示例

```
[[AipOcrService shardService] detectBankCardFromImage:finalImage successHandler:^(id result) {  
    //成功  
} failHandler:^(NSError *err) {  
    //失败  
}];
```

- 结果返回

| 参数                | 类型     | 描述               |
|-------------------|--------|------------------|
| log_id            | UInt64 | 唯一的log id，用于问题定位 |
| result            | Object | 定位和识别结果数组        |
| +bank_card_number | String | 银行卡识别结果          |

| 参数              | 类型     | 描述                           |
|-----------------|--------|------------------------------|
| +bank_name      | string | 银行名，不能识别时为空                  |
| +bank_card_type | uint32 | 银行卡类型，0:不能识别; 1: 借记卡; 2: 信用卡 |

```
// 示例
{
  "log_id": 2742598501;
  "result": {
    "bank_card_number": "622202 0200046230377";
    "bank_card_type": 1;
    "bank_name": "\\U5de5\\U5546\\U94f6\\U884c";
  };
}
```

身份证识别

- 调用示例

```
// 正面
[[AipOcrService shardService] detectIdCardFrontFromImage:finalImage withOptions:nil successHandler:
  // 成功
} failHandler:^(NSError *err) {
  // 失败
}];
```

options参数

| 参数                    | 必选    | 范围               | 类型      | 说明  |
|-----------------------|-------|------------------|---------|---|
| image (已由 image 参数代替) | true  |                  | String  | 图像数据, 支持本地图像文件路径, 图像文件二进制数据, 要求 base64 编码后大小不超过 1M, 最短边至少 15px, 最长边最大 2048px, 支持 jpg/ png/ bmp 格式 |
| isFront (已由函数名区分)     | true  | true、false       | Boolean | true: 身份证正面, false: 身份证背面   |
| detect_direction      | false | true、false       | string  | 是否检测图像朝向, 默认不检测, 即: false。可选值为: true - 检测图像朝向; false - 不检测图像朝向。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度   |
| accuracy              | false | auto、normal、high | string  | 精准度, 精度越高, 速度越慢。default: auto   |

• 结果

| 参数        | 类型    | 描述  |
|-----------|-------|---|
| direction | Int32 | 图像方向, 当 detect_direction=true 时存在。-1:未定义, 0:正向, 1: 逆时针90度, 2:逆时针180度, 3:逆时针270度 |



| 参数               | 类型     | 描述   |
|------------------|--------|--|
| log_id           | Uint64 | 唯一的log id，用于问题定位   |
| words_result     | Array  | 定位和识别结果数组，数组元素的key是身份证的主体字段（正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发日期、失效日期、签发机关）。只返回识别出的字段。若身份证号码校验不通过，则不返回 |
| words_result_num | Uint32 | 识别结果数，表示words_result的元素个数  |
| \+location       | Array  | 位置数组（坐标0点为左上角）   |
| \+\+left         | Uint32 | 表示定位位置的长方形左上顶点的水平坐标  |
| \+\+top          | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标  |
| \+\+width        | Uint32 | 表示定位位置的长方形的宽度  |
| \+\+height       | Uint32 | 表示定位位置的长方形的高度  |
| \+words          | String | 识别结果字符串  |

//示例

```
{
  "log_id": 7037721,
  "direction": 0,
  "words_result_num": 2,
  "words_result": {
    "住址": {
      "location": {
        "left": 227,
        "top": 235,
        "width": 229,
        "height": 51
      },

```

```

        "words": "湖北省天门市渔薪镇杨咀村一组2号",
    }
    ...
}
}
}

```

## 身份证质量控制

- 调用示例
- 初始化

```

[[AipOcrService shardService]getTokenSuccessHandler:^(NSString *token) {
    /**获取到身份证质量控制token*/
} failHandler:^(NSError *error) {

}];
// 利用获取到的token 完成 IdcardQualityAdaptor的初始化
IdcardQualityAdaptor *idcard = [[IdcardQualityAdaptor alloc]init];

[idcard initWithToken:token];

```

- 检测图片

```

- (IdcardQualityModel *)process:(UIImage *)image width:(int)width height:(int)height channel:
(int)channel cardType:(idcard_quality::IdCardType)type;

```

## options参数

参数 | 类型 | 说明 |

----- | ---- | ----- |

image (已由image参数代替) | String | 图像数据，支持本地图像文件路径，图像文件二进制数据，要求base64编码后大小不超过1M，最短边至少15px，最长边最大2048px,支持jpg/png/bmp格式 长宽比应与身份证规格(1.6:1)接近。当身份证 (1)正方向完整位于输入图像内，占比80%-90% (2)正面姓名，号码，反面失效期 清晰，无反光 (3)无明显倾斜 时接口返回IDCARD\_NORMAL, 其他情况返回细分错误码用于引导用户扫描到符合要求的图像。 |

Width, height, channel | Int | UIImage的宽、高、channel (RGB彩图设置为3) |

type | idcard\_quality::IdCardType | IDCARD\_FRONT\_SIDE为检测身份证头像面，IDCARD\_BACK\_SIDE为检测国徽面 |

IdcardQualityModel 中ImageStatusType属性 | Int | IDCARD\_NORMAL = 0 图像包含占比合适，清晰，无反光的身份证 IDCARD\_WRONG\_LOCATION = 1 图像不包含占比合适的身份证。可能的情况包括非身份证，过于倾斜等 IDCARD\_BLURRED = 2 图像包含占比合适的身

份证，但关键字段模糊 IDCARD\_OVER\_EXPOSURE = 3 图像包含占比合适的身份证，但关键字段反光 IDCARD\_REVERSED\_SIDE = 4 输入图像与输入参数设置的身份证国徽/人脸面不匹配 IDCARD\_MOVING = 5 连续输入算法的两帧之间差异过大，可能是镜头或身份证在晃动 IDCARD\_TOO\_SMALL = 6 图像包含身份证，但占比过小 |

- 释放操作

```
- (void)releaseIdcardQuality;
```

### 8.3.3 特殊配置

截图分辨率系数 该系数影响到拍摄照片之后截取的图片大小

在AipOcrSdk/AipOcrSdk/View/AipCutImageView.m 中

```
//截图的分辨率系数 开发者可自行配置
static CGFloat const scale = 1.0;
```

图片放大/缩小系数 在AipOcrSdk/AipOcrSdk/View/AipCutImageView.m 中

```
//捏合操作最大/最小系数
static CGFloat const pinchMaxscale = 10.0;
static CGFloat const pinchMinscale = 0.5;
```

自动重试 识别失败后，自动重试一次。

在AipOcrSdk/AipOcrSdk/AipOcrService 中，配置属性 retry即可

```
/**
 * 是否重试。默认为NO。YES会在失败时自动重试一次。
 */
@property (atomic, assign) bool retry;
```

## 8.4 错误码表

[IdcardQuality错误](#)

| 错误信息                  | 描述                                    |
|-----------------------|---------------------------------------|
| IDCQ_NO_AUTHORITY     | 授权失败，Token无效                          |
| IDCQ_PATH_ERROR       | 模型地址错误，请保证idcardquality.framework完整性  |
| IDCQ_MODEL_INIT_ERROR | 模型初始化错误，请保证idcardquality.framework完整性 |
| IDCQ_IMAGE_ERROR      | image错误，请保证image参数合规                  |
| IDCQ_MEMORY_ERROR     | MEMORY错误，请保证正确调用                      |

### 验证错误

| 错误码    | 错误信息                                    | 说明                         | 备注  |
|--------|---|----------------------------|---|
| 110    | Access token invalid or no longer valid | Access Token 过期失效          | 请重新获得有效的Token   |
| 283501 | License file check error                | 授权文件不匹配                    | 请在 <a href="#">控制台</a> 中配置正确的包名，并确认使用了正确的授权文件           |
| 283502 | App identifier unmatch                  | BundleId不匹配                | 请在 <a href="#">控制台</a> 中配置正确的包名，并确认使用了正确的授权文件           |
| 283504 | Network error                           | 网络请求失败                     | 请授权App网络权限并保证网络通畅                                       |
| 283505 | Server illegal response                 | 服务器返回数据异常                  |   |
| 283601 | Server authentication error             | 身份验证错误。                    | 请在 <a href="#">控制台</a> 中配置应用，并确认填写了正确的AK/SK，或使用了正确的授权文件 |
| 283602 | Authentication time error               | 时间戳不正确，可能是设备时间异常。          |   |
| 283604 | App identifier unmatch                  | 错误的PackageName 或者 BundleId | 请在 <a href="#">控制台</a> 中配置正确的包名，并确认使用了正确的授权文件           |

| 错误码    | 错误信息                  | 说明      | 备注                                 |
|--------|-----------------------|---------|------------------------------------|
| 283700 | Server internal error | 服务器内部错误 | 您可以在工单系统中提交错误信息中的logId，我们将尝试帮您排查原因 |

#### 服务错误

| 错误码    | 错误信息                         | 描述                  |
|--------|------------------------------|---------------------|
| 216015 | module closed                | 模块关闭                |
| 216100 | invalid param                | 非法参数                |
| 216101 | not enough param             | 参数数量不够              |
| 216102 | service not support          | 业务不支持               |
| 216103 | param too long               | 参数太长                |
| 216110 | appid not exist              | APP ID不存在           |
| 216111 | invalid userid               | 非法用户ID              |
| 216200 | empty image                  | 空的图片                |
| 216201 | image format error           | 图片格式错误              |
| 216202 | image size error             | 图片大小错误              |
| 216300 | db error                     | DB错误                |
| 216400 | backend error                | 后端系统错误              |
| 216401 | internal error               | 内部错误                |
| 216500 | unknown error                | 未知错误                |
| 216600 | id number format error       | 身份证的ID格式错误          |
| 216601 | id number and name not match | 身份证的ID和名字不匹配        |
| 216630 | recognize error              | 识别错误                |
| 216631 | recognize bank card error    | 识别银行卡错误（通常为检测不到银行卡） |
| 216632 | ocr                          | unknown error       |
| 216633 | recognize idcard error       | 识别身份证错误（通常为检测不到身份证） |
| 216634 | detect error                 | 检测错误                |
| 216635 | get mask error               | 获取mask图片错误          |
| 282000 | logic internal error         | 业务逻辑层内部错误           |

| 错误码    | 错误信息                  | 描述          |
|--------|-----------------------|-------------|
| 282001 | logic backend error   | 业务逻辑层后端服务错误 |
| 282002 | input encoding error  | 请求参数编码错误    |
| 282100 | image transcode error | 图片压缩转码错误    |

## 8.5 版本更新记录

| 上线日期      | 版本号   | 更新内容                               |
|-----------|-------|------------------------------------|
| 2017.6.20 | 1.2.0 | 新增身份证质量控制 Id-cardQuality.framework |
| 2017.5.18 | 1.1.0 | 新增网图、生僻字接口                         |
| 2017.4.27 | 1.0.1 | 新增模拟器支持                            |
| 2017.3.16 | 1.0.0 | 在线OCR第一版！                          |

## 第9章

# OCR Android SDK 开发者文档

### 9.1 简介

本文档主要介绍OCR Android SDK的安装和使用。在使用本文档前，您需要先了解Optical Character Recognition(OCR)的基础知识，并已经开通了OCR服务。

### 9.2 快速入门

支持的系统和硬件版本

- 系统：支持 Android 4.0 ( API Level 15 ) 到Android7.0 ( API Level 25 ) 系统。需要开发者通过minSdkVersion来保证支持系统的检测。
- CPU架构：armeabi, arm64-v8a, armeabi-v7a, x86
- 机型：手机和平板皆可
- 硬件要求：要求设备上有相机模块。
- 网络：支持WIFI及移动网络，移动网络支持使用NET网关及WAP网关（CMWAP、CTWAP、UNIWAP、3GWAP）。

#### 9.2.1 开发包说明

|                         |                                |
|-------------------------|--------------------------------|
| aip-ocr-android-sdk.zip | // OCR SDK包，包括文档，demo工程，SDK核心库 |
| - libs                  | // lib 库,包括各平台的so库及 jar包。      |
| - OCRDemo               | // demo工程                      |
| - ocr-ui                | // ocr UI模块                    |
| - OCR-android-SDK.md    | // 使用说明文档                      |

sdk的包含的UI部分和demo工程以Android Studio方式提供，sdk部分则可以较方便的集成到eclipse工程中。

- 1. 在官网下载Android SDK压缩包。
- 2. (必须)将下载包libs目录中的ocr-sdk.jar文件拷贝到工程libs目录中，并加入工程依赖。
- 3. (必须)将libs目录下armeabi, arm64-v8a, armeabi-v7a, x86文件夹按需添加到android studio工程src/main/jniLibs目录中， eclipse用户默认为libs目录。
- 4. (可选)如果需要使用UI模块，请在Android studio中以模块方式导入下载包中的ocr-ui文件夹。

9.2.2 DEMO使用说明

Andoird SDK提供了一个可快速运行的demo工程，该工程已经集成了sdk，UI库，您只需直接在Android Studio中导入开发包OCRDemo目录即可运行。

若运行提示“身份验证错误”，请确认在demo工程的MainActivity中填写了正确的Api Key/Secret Key

为工程添加必要的权限 在工程AndroidManifest.xml文件中添加如下权限：

```
<uses-permission android:name=" android.permission.INTERNET" />
<uses-permission android:name=" android.permission.CAMERA" />
<uses-permission android:name=" android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name=" android.permission.WRITE_EXTERNAL_STORAGE" />
```

各个权限的用途说明见下表：

| 名称                     | 用途                      |
|------------------------|-------------------------|
| INTERNET               | 应用联网，发送请求数据至服务器，获得识别结果。 |
| CAMERA                 | 调用相机进行拍照（仅UI部分需要）       |
| WRITE_EXTERNAL_STORAGE | 图片裁剪临时存储                |
| READ_EXTERNAL_STORAGE  | 图片裁剪临时存储                |

Proguard配置 在Proguard配置文件中增加： {\color{emcolor}\textbf{keep class com.baidu.ocr.sdk.\*\*{\*};}}

9.2.3 身份验证与安全

百度AI开放平台使用OAuth2.0授权调用开放API，调用API时必须在URL中带上accesss\_token参数。AccessToken可用AK/SK或者授权文件的方式获得。

OCR Android SDK提供了以下3种AccessToken管理方法.



API Key / Secret Key 此种身份验证方案使用AK/SK获得AccessToken。

虽然SDK对网络传输的敏感数据进行了二次加密，但由于AK/SK是明文填写在代码中，在移动设备中可能会存在AK/SK被盗取的风险。有安全考虑的开发者可使用第二种授权方案。

使用步骤：

1. 在[管理控制台](#)中配置OCR应用
2. 复制应用的Api Key（简称AK）和 Secret Key（简称SK），初始化OCR单例：

```
OCR.getInstance().initAccessTokenWithAkSk(new OnResultListener<AccessToken>() {  
    @Override  
    public void onResult(AccessToken result) {  
        // 调用成功，返回AccessToken对象  
        String token = result.getAccessToken();  
    }  
    @Override  
    public void onError(OCRError error) {  
        // 调用失败，返回OCRError子类SDKError对象  
    }  
}, getApplicationContext(), "您的应用AK", "您的应用SK");
```

由于AK/SK是明文填写在代码中，在移动设备中可能会存在AK/SK被盗取的风险。有安全考虑的开发者可使用第二种授权方案。

授权文件（安全模式） 此种身份验证方案使用授权文件获得AccessToken，缓存在本地。建议有安全考虑的开发者使用此种身份验证方式。

在您的移动APP分发出去之后，APP存在被反编译的可能，所以直接将AK / SK 置于APP源码之中，存在被盗取的风险。采用授权文件的身份验证方法，可有效保护AK/SK在移动设备中的安全。攻击者即使拦截了流量，盗取了授权文件，也难以盗用您的配额。

使用步骤：

1. 在[官网](#)中配置应用
2. 下载对应应用的授权文件
3. 将授权文件添加至工程asserts文件夹，文件名必须为[aip.license](#)
4. 调用initAccessToken方法，初始化OCR单例：

```
OCR.getInstance().initAccessToken(new OnResultListener<AccessToken>() {  
    @Override  
    public void onResult(AccessToken result) {  
        // 调用成功，返回AccessToken对象
```

```

        String token = result.getAccessToken();
    }
    @Override
    public void onError(OCRError error) {
        // 调用失败，返回OCRError子类SDKError对象
    }
}, getApplicationContext());

```

**自助AccessToken管理** 此种身份验证方案直接使用开发者提供的AccessToken。

若开发者的应用有自行搭建的服务端，也可在自己的服务端进行token的获取与管理，然后分配给移动客户端使用。token的获得可以参考[API鉴权认证机制](#)。此种授权方案在移动客户端上没有任何AK/SK信息，风险系数低，但需要开发者自行管理token的获取与分配，适合有条件的开发者使用。

使用步骤：

1. 将您需要使用的token传入对应的构造函数，初始化OCR单例：

```
OCR.getInstance().initWithToken(getApplicationContext(), "自行获取的access_token");
```

## 9.3 接口调用说明

### 9.3.1 OCR-UI模块

OCR-UI模块提供了一套默认的UI。如需使用，请将ocr\_ui模块包含到您的工程，具体使用可参考示例工程（OCRDemo）的IdCardActivity、BankCardActivity和GeneralActivity文件。

**OCR-UI模块调用示例** 调用拍摄activity

```

// 生成intent对象
Intent intent = new Intent(IdCardActivity.this, CameraActivity.class);

// 设置临时存储
intent.putExtra(CameraActivity.KEY_OUTPUT_FILE_PATH, FileUtil.getSaveFile(getApplicationContext()));

// 调用拍摄银行卡正面的activity
intent.putExtra(CameraActivity.KEY_CONTENT_TYPE, CameraActivity.CONTENT_TYPE_ID_CARD_FRONT);
startActivityForResult(intent, REQUEST_CODE_CAMERA);

```

## 通过onActivityResult获取拍摄结果

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    // 获取调用参数
    String contentType = data.getStringExtra(CameraActivity.KEY_CONTENT_TYPE);
    // 通过临时文件获取拍摄的图片
    String filePath = FileUtil.getSaveFile(getApplicationContext()).getAbsolutePath();
    // 判断拍摄类型（通用，身份证，银行卡）
    if (requestCode == REQUEST_CODE_CAMERA && resultCode == Activity.RESULT_OK) {
        // 判断是否是身份证正面
        if (CameraActivity.CONTENT_TYPE_ID_CARD_FRONT.equals(contentType)){
            // 获取图片文件调用sdk数据接口，见数据接口说明
        }
    }
}

```

## 9.3.2 数据接口

## 通用文字识别（含位置信息版）

- 调用示例

```

// 通用文字识别参数设置
GeneralParams param = new GeneralParams();
param.setDetectDirection(true);
param.setImageFile(new File(filePath));

// 调用通用文字识别服务
OCR.getInstance().recognizeGeneral(param, new OnResultListener<GeneralResult>() {
    @Override
    public void onResult(GeneralResult result) {
        // 调用成功，返回GeneralResult对象
        for (WordSimple wordSimple : result.getWordList()) {
            // Word类包含位置信息
            Word word = (Word) wordSimple;
            sb.append(word.getWords());
            sb.append("\n");
        }
    }
    @Override
    public void onError(OCRError error) {

```

```

        // 调用失败，返回OCRError对象
    }
});

```

### options参数详情

| 参数                    | 是否必选  | 类型     | 可选值范围                                   | 说明   |
|-----------------------|-------|--------|---|--|
| image                 | true  | string | -                                       | 图 像 数 据，base64 编 码，要求base64 编码后大小不超过 1M，最短边至少 15px，最长边最大 2048px, 支持 jpg/ png/ bmp 格式                                   |
| recognize_granularity | false | string | big、small                               | 是否定位单字符位置，big：不定位单字符位置，默认值；small：定位单字符位置   |
| mask                  | false | string | -                                       | 表示mask区域的黑白灰度图片，白色代表选中, base64编码   |
| language_type         | false | string | CHN_ENG、ENG、POR、FRE、GER、ITA、SPA、RUS、JAP | 识别语言类型，默认为 CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |

| 参数                 | 是否必选  | 类型      | 可选值范围      | 说明   |
|--------------------|-------|---------|------------|--|
| detect_direction   | false | boolean | true、false | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。 |
| detect_language    | false | string  | true、false | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）   |
| classify_dimension | false | string  | lottery    | 分类维度（根据 OCR 结果进行分类），逗号分隔，当前只支持 lottery。lottery：彩票分类，设置 detect_direction 有助于提升精度          |
| vertexes_location  | false | string  | true、false | 是否返回文字外接多边形顶点位置，不支持单字位置。默认为false   |

- 结果返回

| 字段                 | 必选 | 类型      | 说明   |
|--------------------|----|---------|--|
| direction          | 否  | int32   | 图 像 方 向， 当 detect_direction=true 时 存 在。 - -1: 未 定 义， - 0: 正 向， - 1: 逆 时 针 90 度， - 2: 逆 时 针 180 度， - 3: 逆 时 针 270 度 |
| log_id             | 是  | uint64  | 唯 一 的 log id， 用 于 问 题 定 位  |
| words_result       | 是  | array() | 定 位 和 识 别 结 果 数 组  |
| words_result_num   | 是  | uint32  | 识 别 结 果 数， 表 示 words_result 的 元 素 个 数  |
| +vertexes_location | 否  | array() | 当 前 为 四 个 顶 点: 左 上， 右 上， 右 下， 左 下。 当 vertexes_location=true 时 存 在  |
| ++x                | 是  | uint32  | 水 平 坐 标（ 坐 标 0 点 为 左 上 角 ）   |
| ++y                | 是  | uint32  | 垂 直 坐 标（ 坐 标 0 点 为 左 上 角 ）   |
| +location          | 是  | array() | 位 置 数 组（ 坐 标 0 点 为 左 上 角 ）   |
| ++left             | 是  | uint32  | 表 示 定 位 位 置 的 长 方 形 左 上 顶 点 的 水 平 坐 标  |
| ++top              | 是  | uint32  | 表 示 定 位 位 置 的 长 方 形 左 上 顶 点 的 垂 直 坐 标  |
| ++width            | 是  | uint32  | 表 示 定 位 位 置 的 长 方 形 的 宽 度  |
| ++height           | 是  | uint32  | 表 示 定 位 位 置 的 长 方 形 的 高 度  |
| +words             | 否  | string  | 识 别 结 果 字 符 串  |

| 字段         | 必选 | 类型      | 说明   |
|------------|----|---------|--|
| +chars     | 否  | array() | 单 字 符 结 果， recognize_granularity=small 时存在 |
| ++location | 是  | array() | 位置数组（坐标0点为左上角）                             |
| +++left    | 是  | uint32  | 表示定位位置的长方形左上顶点的水平坐标                        |
| +++top     | 是  | uint32  | 表示定位位置的长方形左上顶点的垂直坐标                        |
| +++width   | 是  | uint32  | 表示定位定位位置的长方形的宽度                            |
| +++height  | 是  | uint32  | 表示位置的长方形的高度                                |
| ++char     | 是  | string  | 单字符识别结果                                    |

```
// 示例
{
  direction : 2,
  log_id : 676709620,
  words_result : [ {
    location : {
      height : 20;
      left : 86;
      top : 387;
      width : 22;
    };
    words : "N";
  },
  ],
  words_result_num : 1;
}
```

通用文字识别

- 调用示例

```
// 通用文字识别参数设置
GeneralBasicParams param = new GeneralBasicParams();
param.setDetectDirection(true);
param.setImageFile(new File(filePath));

// 调用通用文字识别服务
OCR.getInstance().recognizeGeneralBasic(param, new OnResultListener<GeneralResult>() {
    @Override
    public void onResult(GeneralResult result) {
        // 调用成功, 返回GeneralResult对象
        for (WordSimple wordSimple : result.getWordList()) {
            // wordSimple不包含位置信息
            wordSimple word = wordSimple;
            sb.append(word.getWords());
            sb.append("\n");
        }
    }
    @Override
    public void onError(OCRError error) {
        // 调用失败, 返回OCRError对象
    }
});
```

options参数详情

| 参数    | 是否必选  | 类型     | 可选值范围 | 说明   |
|-------|-------|--------|-------|--|
| image | true  | string | -     | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |
| mask  | false | string | -     | 表示mask区域 的黑白灰度图 片，白色代 表选中，base64编 码  |



| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明  |
|--------------------|------|--------|---------|---|
| classify_dimension | 否    | string | lottery | 分类维度（根据 OCR 结果进行分类），逗号分隔，当前只支持 lottery。lottery：彩票分类，设置 detect_direction 有助于提升精度 |

• 结果返回

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| words_result     | 是  | array() | 定位和识别结果数组  |
| +words           | 否  | string  | 识别结果字符串  |

通用文字识别(含生僻字版)

• 调用示例

```
// 通用文字识别(含生僻字版)参数设置
GeneralBasicParams param = new GeneralBasicParams();
param.setDetectDirection(true);
param.setImageFile(new File(filePath));
```

```
// 调用通用文字识别服务
OCR.getInstance().recognizeGeneralEnhanced(param, new OnResultListener<GeneralResult>() {
    @Override
    public void onResult(GeneralResult result) {
        // 调用成功, 返回GeneralResult对象
        for (WordSimple wordSimple : result.getWordList()) {
            // wordSimple不包含位置信息
            wordSimple word = wordSimple;
            sb.append(word.getWords());
            sb.append("\n");
        }
    }
    @Override
    public void onError(OCRError error) {
        // 调用失败, 返回OCRError对象
    }
});
```

options参数详情

| 参数    | 是否必选  | 类型     | 可选值范围 | 说明  |
|-------|-------|--------|-------|---|
| image | true  | string | -     | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |
| mask  | false | string | -     | 表示mask区域的黑白灰度图 片，白色代表选中, base64编 码  |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明   |
|--------------------|------|--------|---------|--|
| classify_dimension | 否    | string | lottery | 分类维度（根据OCR结果进行分类），逗号分隔，当前只支持lottery。lottery：彩票分类，设置detect_direction有助于提升精度 |

• 结果返回

| 字段               | 必选 | 类型      | 说明  |
|------------------|----|---------|---|
| direction        | 否  | int32   | 图像方向，当detect_direction=true时存在。- -1:未定义，- 0:正向，- 1:逆时针90度，- 2:逆时针180度，- 3:逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位  |
| words_result_num | 是  | uint32  | 识别结果数，表示words_result的元素个数   |
| words_result     | 是  | array() | 定位和识别结果数组   |
| +words           | 否  | string  | 识别结果字符串   |

网络图片文字识别

• 调用示例

```
// 网络图片文字识别参数设置
GeneralBasicParams param = new GeneralBasicParams();
param.setDetectDirection(true);
param.setImageFile(new File(filePath));
```

```
// 调用通用文字识别服务
OCR.getInstance().recognizeWebimage(param, new OnResultListener<GeneralResult>() {
    @Override
    public void onResult(GeneralResult result) {
        // 调用成功, 返回GeneralResult对象
        for (WordSimple wordSimple : result.getWordList()) {
            // wordSimple不包含位置信息
            wordSimple word = wordSimple;
            sb.append(word.getWords());
            sb.append("\n");
        }
    }
    @Override
    public void onError(OCRError error) {
        // 调用失败, 返回OCRError对象
    }
});
```

options参数详情

| 参数    | 是否必选  | 类型     | 可选值范围 | 说明   |
|-------|-------|--------|-------|--|
| image | true  | string | -     | 图 像 数 据，base64 编 码，要求base64 编 码后大小不超 过 4M，最 短 边 至少 15px，最 长 边 最大 4096px，支 持 jpg/ png/ bmp 格式 |
| mask  | false | string | -     | 表示mask区域 的黑白灰度图 片，白色代表 选中, base64编 码   |

| 参数               | 是否必选  | 类型      | 可选值范围   | 说明  |
|------------------|-------|---------|---|---|
| language_type    | false | string  | CHN_ENG、<br>ENG、POR、<br>FRE、GER、<br>ITA、SPA、<br>RUS、JAP | 识别语言类型，默认为CHN_ENG。可选值包括：- CHN_ENG：中英文混合；- ENG：英文；- POR：葡萄牙语；- FRE：法语；- GER：德语；- ITA：意大利语；- SPA：西班牙语；- RUS：俄语；- JAP：日语 |
| detect_direction | false | boolean | true、false  | 是否检测图像朝向，默认不检测，即：false。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度。可选值包括：- true：检测朝向；- false：不检测朝向。                              |
| detect_language  | FALSE | string  | true、false  | 是否检测语言，默认不检测。当前支持（中文、英语、日语、韩语）  |

| 参数                 | 是否必选 | 类型     | 可选值范围   | 说明  |
|--------------------|------|--------|---------|---|
| classify_dimension | 否    | string | lottery | 分类维度（根据 OCR 结果进行分类），逗号分隔，当前只支持 lottery。lottery：彩票分类，设置 detect_direction 有助于提升精度 |

• 结果返回

| 字段               | 必选 | 类型      | 说明   |
|------------------|----|---------|--|
| direction        | 否  | int32   | 图像方向，当 detect_direction=true 时存在。- -1: 未定义，- 0: 正向，- 1: 逆时针90度，- 2: 逆时针180度，- 3: 逆时针270度 |
| log_id           | 是  | uint64  | 唯一的log id，用于问题定位   |
| words_result_num | 是  | uint32  | 识别结果数，表示 words_result 的元素个数  |
| words_result     | 是  | array() | 定位和识别结果数组  |
| +words           | 否  | string  | 识别结果字符串  |

银行卡识别

• 调用示例

```
// 银行卡识别参数设置
BankCardParams param = new BankCardParams();
param.setImageFile(new File(filePath));
```



```
// 调用银行卡识别服务
OCR.getInstance().recognizeBankCard(param, new OnResultListener<BankCardResult>() {
    @Override
    public void onResult(BankCardResult result) {
        // 调用成功, 返回BankCardResult对象
    }
    @Override
    public void onError(OCRError error) {
        // 调用失败, 返回OCRError对象
    }
});
```

• 结果返回

| 参数                 | 类型     | 描述                            |
|--------------------|--------|-------------------------------|
| log_id             | Uint64 | 唯一的log id, 用于问题定位             |
| result             | Object | 定位和识别结果数组                     |
| \+bank_card_number | String | 银行卡识别结果                       |
| \+bank_name        | String | 银行名, 不能识别时为空                  |
| \+bank_card_type   | uint32 | 银行卡类型, 0:不能识别; 1: 借记卡; 2: 信用卡 |

```
// 示例
{
    "log_id": 3207866271;
    result: {
        "bank_card_number": "6226 2288 8888 8888",
        "bank_card_type": 1,
        "bank_name": "\\U5de5\\U5546\\U94f6\\U884c"
    };
}
```

身份证识别

• 调用示例

```
// 身份证识别参数设置
IDCardParams param = new IDCardParams();
```

```
param.setImageFile(new File(filePath));

// 调用身份证识别服务
OCR.getInstance().recognizeIDCard(param, new OnResultListener<IDCardResult>() {
    @Override
    public void onResult(IDCardResult result) {
        // 调用成功, 返回IDCardResult对象
    }
    @Override
    public void onError(OCRError error) {
        // 调用失败, 返回OCRError对象
    }
});
```

options参数

| 参数      | 必选   | 范围         | 类型      | 说明   |
|---------|------|------------|---------|--|
| image   | true |            | String  | 图像数据, 支持本地图像文件路径, 图像文件二进制数据, 要求base64编码后大小不超过1M, 最短边至少15px, 最长边最大2048px, 支持jpg/ png/ bmp格式 |
| isFront | true | true、false | Boolean | true: 身份证正面, false: 身份证背面  |

| 参数               | 必选    | 范围               | 类型     | 说明   |
|------------------|-------|------------------|--------|--|
| detect_direction | false | true、false       | string | 是否检测图像朝向，默认不检测，即：false。可选值为：true - 检测图像朝向；false - 不检测图像朝向。朝向是指输入图像是正常方向、逆时针旋转 90/180/270 度 |
| accuracy         | false | auto、normal、high | string | 精准度，精度越高，速度越慢。default：auto   |

• 结果返回

| 参数               | 类型     | 描述   |
|------------------|--------|--|
| direction        | Int32  | 图像方向，当 detect_direction=true 时存在。-1:未定义，0:正向，1: 逆时针90度， 2:逆时针180度， 3:逆时针270度                           |
| log_id           | Uint64 | 唯一的log id，用于问题定位   |
| words_result     | Array  | 定位和识别结果数组，数组元素的key是身份证的主体字段（正面支持：住址、公民身份号码、出生、姓名、性别、民族，背面支持：签发机关、签发日期、失效日期）。只返回识别出的字段。若身份证号码校验不通过，则不返回 |
| words_result_num | Uint32 | 识别结果数，表示 words_result 的元素个数  |

| 参数         | 类型     | 描述                  |
|------------|--------|---------------------|
| \+location | Array  | 位置数组（坐标0点为左上角）      |
| \+\+left   | Uint32 | 表示定位位置的长方形左上顶点的水平坐标 |
| \+\+top    | Uint32 | 表示定位位置的长方形左上顶点的垂直坐标 |
| \+\+width  | Uint32 | 表示定位位置的长方形的宽度       |
| \+\+height | Uint32 | 表示定位位置的长方形的高度       |
| \+words    | String | 识别结果字符串             |

```
//示例
{
  "log_id": 7037721,
  "direction": 0,
  "words_result_num": 2,
  "words_result": {
    "住址": {
      "location": {
        "left": 227,
        "top": 235,
        "width": 229,
        "height": 51
      },
      "words": "湖北省天门市渔薪镇杨咀村一组2号",
    },
    ...
  }
}
```

9.4 错误码表

[验证错误](#)

| 错误码    | 错误信息                                    | 说明                            | 备注  |
|--------|---|-------------------------------|---|
| 110    | Access token invalid or no longer valid | Access Token 过期失效             | 请重新获得有效的Token   |
| 283501 | License file check error                | 授权文件不匹配                       | 请在 <a href="#">控制台</a> 中配置正确的包名，并确认使用了正确的授权文件           |
| 283502 | App identifier unmatch                  | BundleId不匹配                   | 请在 <a href="#">控制台</a> 中配置正确的包名，并确认使用了正确的授权文件           |
| 283503 | License file not exists                 | 请确认aip.licence文件存在于assets文件夹中 |   |
| 283504 | Network error                           | 网络请求失败                        | 请授权App网络权限并保证网络通畅                                       |
| 283505 | Server illegal response                 | 服务器返回数据异常                     |   |
| 283506 | Load jni so library error               | JNI加载异常                       | 请确认开发包中的so库被正确加载  |
| 283601 | Server authentication error             | 身份验证错误。                       | 请在 <a href="#">控制台</a> 中配置应用，并确认填写了正确的AK/SK，或使用了正确的授权文件 |
| 283602 | Authentication time error               | 时间戳不正确，可能是设备时间异常。             | 请确保不要改变调用设备的本地时间  |
| 283604 | App identifier unmatch                  | 错误的PackageName 或者 BundleId    | 请在 <a href="#">控制台</a> 中配置正确的包名，并确认使用了正确的授权文件           |
| 283700 | Server internal error                   | 服务器内部错误                       | 您可以在工单系统中提交错误信息中的logId，我们将尝试帮您排查错误原因                    |

| 错误码    | 错误信息                         | 描述                  |
|--------|------------------------------|---------------------|
| 216015 | module closed                | 模块关闭                |
| 216100 | invalid param                | 非法参数                |
| 216101 | not enough param             | 参数数量不够              |
| 216102 | service not support          | 业务不支持               |
| 216103 | param too long               | 参数太长                |
| 216110 | appid not exist              | APP ID不存在           |
| 216111 | invalid userid               | 非法用户ID              |
| 216200 | empty image                  | 空的图片                |
| 216201 | image format error           | 图片格式错误              |
| 216202 | image size error             | 图片大小错误              |
| 216300 | db error                     | DB错误                |
| 216400 | backend error                | 后端系统错误              |
| 216401 | internal error               | 内部错误                |
| 216500 | unknown error                | 未知错误                |
| 216600 | id number format error       | 身份证的ID格式错误          |
| 216601 | id number and name not match | 身份证的ID和名字不匹配        |
| 216630 | recognize error              | 识别错误                |
| 216631 | recognize bank card error    | 识别银行卡错误（通常为检测不到银行卡） |
| 216632 | ocr                          | unknown error       |
| 216633 | recognize idcard error       | 识别身份证错误（通常为检测不到身份证） |
| 216634 | detect error                 | 检测错误                |
| 216635 | get mask error               | 获取mask图片错误          |
| 282000 | logic internal error         | 业务逻辑层内部错误           |
| 282001 | logic backend error          | 业务逻辑层后端服务错误         |
| 282100 | image transcode error        | 图片压缩转码错误            |

## 9.5 版本更新记录

| 上线日期      | 版本号   | 更新内容   |
|-----------|-------|--|
| 2017.6.20 | 1.2.0 | ocr_ui库身份证识别升级，交互修改为基于本地模型实现实时扫描判断后自动上传识别身份证                         |
| 2017.5.18 | 1.1.0 | 增加通用文字识别基础版，生僻字，网图接口的SDK接口和demo演示；移除okhttp依赖；支持x86架构CPU；略微优化了demo的交互 |
| 2017.4.13 | 1.0.2 | 修复部分用户使用ak，sk方式无法获取token的问题  |
| 2017.3.23 | 1.0.1 | 更新demo获取token失败的错误提示的交互  |
| 2017.3.16 | 1.0.0 | 在线OCR第一版！  |





## 第10章

# 常见问题

Q：图片格式和分辨率有限制吗？

A：目前图片格式支持jpg、png、bmp，图片的长和宽要求最短边大于10px，最长边小于2048 px。

Q：图片大小有限制吗？

A：图像编码后大小必须小于4M，建议不要超过1M；最小边长不小于15，最大边长不超过4096，建议不要超过2048（最大边长超过2048或编码后大于1M的图像会被缩放，影响效果，建议控制输入图像大小）

Q：识别的图片支持怎样输入？

A：目前文字识别接口仅支持base64编码输入。

Q：什么是base64编码，如何提供？

A：图片的base64编码指将一副图片数据编码成一串字符串，使用该字符串代替图像地址。您可以首先得到图片的二进制，然后用Base64格式编码即可。注：图片的base64编码是不包含图片头的，如（data:image/jpg;base64,）

Q：身份证识别支持图片的竖拍吗？

A：横拍竖拍都支持。

Q：银行卡识别支持翻转识别吗？

A：目前翻转不支持。

Q：怎么提高识别的准确率和识别速度？

A：文字识别的准确率跟拍摄光照、背景、清晰度等因素有关，可在图片采集端尽量缩小要识别文字区域，尽量保证图片清晰度人眼可以辨认；

「身份证识别」：在保证文字清晰度的前提下，降低图片大小可以提高识别速度。请尽量将图片高和宽都控制在1000 px以下，图片尽量没有反光、倾斜小于10度。

「银行卡识别」：在保证文字清晰度的前提下，降低图片大小可以提高识别速度。请尽量保证银行卡号清晰、倾斜角度不超过30度，银行卡尽量占满图片的大部分。

「通用文字识别」：在保证文字清晰度的前提下，降低图片大小、减少文字个数都可以提高识别速度，请尽量将图片高和宽都控制在1000 px以下，单图字数控制在70字以内。

Q: 通用文字识别对字体有要求吗？

A: 字体没要求，但很特殊字体可能支持不好。

Q: 通用文字识别支持哪些语言识别？

A: 支持十国语言（中/英/日/韩/西班牙/法/德/俄/葡萄牙/意大利，语言类型有参数设置）

Q: 支持文档扫描图片的文字识别吗？

A: 目前开放的自然场景下，通用文字识别接口也同样支持文档扫描图片。