

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、 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

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、 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

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 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

7.7 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、 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）

[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， 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
}
```

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