

# 图片转文档（阿里云市场）

在线试用：

<https://try.dhconvert.com/>

小程序搜索：

度慧文档转换

## 概述

### 使用流程

由于转换需要时间，文件越大页数越多，转换越久，故采用**异步**的方式获得转换结果。即调用转换接口后会获得token，随后有2种方式查询转换结果：

1. 定时轮询结果，调用“查询结果接口”。详细见：[查询结果](#)
2. 回调，设置callbackurl，当转换结束后，系统会回调该URL直接推送转换结果。详细见：[回调URL](#)

### 调用转换接口

v1和v2接口功能相同，调用方式不一样，可自选合适的。

**v1接口包括3种转换方式：**

- 一张图片转为文档，图片是一个下载链接，用HTTP GET方式，见：[单图转换GET\\_v1](#)
- 一张图片转为文档，图片POST到服务器，用HTTP POST Form Data方式，见：[单图转换POST\\_v1](#)
- 多张图片转为文档，图片是多个下载链接，用HTTP POST JSON方式，见：[多图转换POST\\_v1](#)

**v2接口统一为HTTP POST JSON：**

- JSON支持输入：文件url；文件Base64字符串；多张图片的url。见：[文档转换\\_v2](#)
- v2版本的API除了异步，还支持同步调用，见：[同步调用](#)
- v2版本中的参数如果未出现在v1版本中，v1版本用同样参数也能工作

同时支持单图 [公式转换](#)

提供Office文档合并功能，用于合并单个转出的Word，PPT或Excel文件 [Office文档合并POST](#)

输出以下文件格式：

| 类型                                  | 扩展名                           |
|-------------------------------------|-------------------------------|
| 微软Office文档                          | doc, docx, pptx, xlsx         |
| 开放版式文档                              | ofd                           |
| 文本文件                                | txt, rtf                      |
| Markdown文件 ( <a href="#">公式转换</a> ) | md                            |
| JSON结构（单图）                          | json ( <a href="#">结构定义</a> ) |

调用转换API需要签名，详细见文档附录：[阿里签名](#) 调用[查询结果](#)API无需签名。

阿里云支持从OSS内网直接下载文件，节约流量，见：[阿里云独有部分](#)

# 图片转换\_v2

异步url:

```
https://img2doc.market.alicloudapi.com/v2/convert_async
```

同步url:

```
https://img2doc.market.alicloudapi.com/v2/convert_sync
```

HTTP方式: POST

Header中的Content-Type传入application/json

Body是JSON格式, 支持以下3种输入源文件的方法:

方法1: 图片url:

```
{"input": "http://xxx.jpg", "type": "docx"}
```

方法2: 图片Base64字符串, Base64字符串最大8M:

```
{"input": "base64字符串", "type": "docx"}
```

方法3: 多张图片url:

```
{"input": ["http://xxx.jpg", "http://xxx.png"], "type": "docx"}
```

必须签名才能调用成功, 签名见[阿里签名规则](#):

<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/use-cases/call-apis>

支持转换为以下文件格式:

| 类型          | 扩展名 (type取值)                  |
|-------------|-------------------------------|
| 微软Office文档  | doc, docx, pptx, xlsx         |
| 开放版式文档      | ofd                           |
| 文本文件        | txt, rtf                      |
| JSON结构 (单图) | json ( <a href="#">结构定义</a> ) |

自定义参数:

| 参数            | 类型     | 备注  | 默认值  |
|---------------|--------|---|------|
| type          | string | 小写, 需转换为的文件类型, 例如docx   | 必须发送 |
| 输出文件相关        |        |   |      |
| language_code | string | OCR识别语言, 默认简体中文: ch<br>支持近80种语言, 具体参考 <a href="#">OCR支持语言</a> | ch   |
| dewarp        | int    | 是否切边矫正, 默认0否, 1是。如果打开, 每次只允许传入一张图 (图片分辨率短边大于20, 长边小于10000)    | 0    |
| clean         | int    | 是否清除图像背景只显示文字:<br>默认0: 智能保留背景, 尽量让背景不影响文字编辑                   | 0    |

|                    |        |  |   |
|--------------------|--------|--|---|
|                    |        | 1是，完全清除背景<br>2否，完全保留背景   |   |
| norotate           | int    | 是否根据文字方向自动旋转图片，默认1: 不旋转，0: 自动判断。如果确定图片中的文字方向则传1，速度更快   | 1 |
| a4                 | int    | 是否输出A4大小文件，默认0否，1是。仅在pagesize为0时生效   | 0 |
| pagesize           | int    | 设定页面大小。默认0:维持原图比例。1:A3, 2:A4, 3:A5, 4:B4, 5:B5, 6:Letter, 7:Legal, 8:Tabloid, 9:Ledger<br>当不为0时，a4参数失效 | 0 |
| qr                 | int    | 是否识别条码和二维码。如果type是json，会输出条码和二维码。<br>其他type会避开码所在区域，识别更精准。   | 0 |
| split2p            | int    | 横向图片若有左右两部分（常见于试卷），分割为2页。默认0:否，1是  | 0 |
| outfilename        | string | 生成的文件的文件名，默认随机   | 空 |
| 输出Office文件相关       |        |  |   |
| wordnoimage        | int    | 如果转为Word文件，默认0：需要图片；1：不需要图片  | 0 |
| wordabsolutelayout | int    | 如果转为Word文件，默认0：流式布局；1：绝对布局（位置精准，浏览方便，但是编辑方式非流式不利于编辑）   | 0 |
| excelonesheet      | int    | 如果转为Excel文件，默认0：每张图一个工作表；1：合成一个工作表   | 0 |
| table              | int    | 如果转为Word，PPT或json，是否识别表格<br>默认0：否<br>1：只识别表格<br>2：识别表格和下划线   | 0 |
| 输出文本文件相关           |        |  |   |
| textnoformat       | int    | 如果转为Txt文本，默认0：保持原有布局，1：无需布局  | 0 |
| textperpage        | int    | 如果转为Txt文本，默认0：所有页转为单个txt文件，1：每一页转为一个txt文件，并且打包为一个zip包  | 0 |
| 其他                 |        |  |   |
| callbackurl        | string | 回调URL，转换结束后，会回调该URL，详细见 <b>回调URL</b>   | 空 |

#### 请求示例:

- 例1: 把单张图片转为word

```
{"input": "http://xxx.jpg", "type": "docx"}
```

- 例2: 把单张图片保留背景，识别表格，自动判断文字方向，转为ppt

```
{"input": ["http://xxx.jpg"], "type": "pptx", "clean": 2, "table": 1, "norotate": 0}
```

- 例3: 把2张图片每一张转为单独的txt文件

```
{"input": ["http://xxx.jpg", "http://xxx.png"], "textperpage": 1}
```

#### 返回数据结构【异步】：

| 名称     | 类型         | 是否必须返回 | 备注         |
|--------|------------|--------|------------|
| code   | number     | 是      | 10000:请求成功 |
| msg    | string     | 是      |            |
| result | Dictionary | 否      | 成功后返回      |

result:

| 名称    | 类型     | 是否必须返回 | 备注                         |
|-------|--------|--------|----------------------------|
| token | string | 是      | 用于 <a href="#">查询结果</a> 接口 |

返回示例(成功状态)【异步】：

```
{
  "code":10000,
  "msg": "",
  "result":{"token":"xxx"}
}
```

返回示例(失败状态)【异步】：

```
{
  "code":40001,
  "msg":"ParmNotRight"
}
```

同步调用

同步调用的最大返回时间是60秒，如果60秒内转换结束则直接返回结果。否则会返回token，之后和异步方式一样可以调用[查询结果](#)接口查询该token的转换结果。所以同步调用如果传入的文件过大，无法保证在60秒内结束，则转为异步流程。

返回数据结构【同步】：

| 名称     | 类型         | 是否必须返回 | 备注         |
|--------|------------|--------|------------|
| code   | number     | 是      | 10000:请求成功 |
| msg    | string     | 是      |            |
| token  | string     | 是      | 请求的token   |
| result | Dictionary | 否      | 成功后返回      |

result:

| 名称       | 含义     | 类型      | 是否必须返回            | 备注                       |
|----------|--------|---------|-------------------|--------------------------|
| fileurl  | 输出文件地址 | string  | 否（status为Done时返回） | 转换出来的文件地址，http和https都支持  |
| count    | 总图片数   | integer | 否（status为Done时返回） | 图片总数                     |
| filesize | 文件大小   | integer | 否（status为Done时返回） | 输出文件大小                   |
| status   | 状态     | string  | 是                 | Done：转换成功<br>Failed：转换失败 |

返回示例(成功状态)【同步】：

```
{
  "code": 10000,
  "msg": "",

```

```
"result": {  
  "count": 1,  
  "filesize": 17747,  
  "fileurl": "https://file.duhuitech.com/o/xxx/xxx.docx",  
  "status": "Done"  
},  
"token": "xxx"  
}
```

**返回示例(超时状态)【同步】：**

```
{  
  "code": 40500,  
  "msg": "Timeout, query token later",  
  "token": "xxx"  
}
```

# 查询结果

请求参数:

| 参数    | 类型     | 备注             | 是否必须发送 |
|-------|--------|----------------|--------|
| token | string | 调用转换接口拿到的token | 是      |

请求示例:

```
https://api.duhuitech.com/q?token=xxx
```

无需签名，无调用次数限制

由于转换需要时间，文件越大页数越多，转换越久，故需要**轮询**查询接口来获得结果。查询频率可以是1s一次，也可以更长一些。**查询后先看status，如果是Done或Failed，则转换结束，停止轮询。如果是Doing或Pending，则继续轮询。**

返回数据结构:

| 名称     | 类型         | 是否必须返回 | 备注         |
|--------|------------|--------|------------|
| code   | number     | 是      | 10000:请求成功 |
| msg    | string     | 是      |            |
| token  | string     | 是      | 请求的token   |
| result | Dictionary | 否      | 成功后返回      |

result:

| 名称       | 含义     | 类型      | 是否必须返回                | 备注   |
|----------|--------|---------|-----------------------|--|
| status   | 状态     | string  | 是                     | Pending：还未开始<br>Doing：正在转换<br>Done：转换成功<br>Failed：转换失败 |
| progress | 进度     | number  | 否（status为Doing时返回）    | 范围：0.00 - 1.00， 比如0.88表示88%                            |
| fileurl  | 输出文件地址 | string  | 否（status为Done时返回）     | 转换出来的文件地址， http和https都支持                               |
| count    | 总图片数   | integer | 否（status为Done时返回）     | 图片总数   |
| filesize | 文件大小   | integer | 否（status为Done时返回）     | 输出文件大小   |
| reason   | 失败原因   | string  | 否（status为Failed时可能返回） | 转换失败的原因  |

返回示例(成功状态):

```
{
  "code":10000,
  "msg": "",
  "token": "xxx",
  "result":
  {
    "progress":0.02,
    "status": "Doing"
  }
}
```

```
{
  "code":10000,
  "msg": "",
  "token":"xxx",
  "result":
  {
    "status":"Done",
    "fileurl":"https://file.duhuitech.com/o/xxx/xxx.docx",
    "filesize":17747,
    "count":1
  }
}
```

**返回示例(失败状态):**

```
{
  "code":40000,
  "msg":"No such token"
}
```

# 公式转换POST

将一张图片转换为一个文档，可以是url传入一张图片的地址，也可以是file参数传入图片的Base64。高精度识别公式，可以转换为公式可编辑的word文件，或者markdown文件。**注意版式不会保留。**

传入JSON字符串，POST方式

Header中的Content-Type传入application/json

POST Body传入JSON，格式如下：

请求参数BODY:

| 参数          | 类型     | 备注  | 是否必须发送         |
|-------------|--------|---|----------------|
| url         | string | 要转换的图片URL，支持http(s)，ftp开头                     | 是（url和file2选1） |
| file        | string | 要转换的图片的Base64数据字符串，最大8M                       | 是（url和file2选1） |
| type        | string | 小写，需转换为的文件类型，docx或md                          | 是              |
| outfilename | string | 生成的文件的文件名，默认随机                                | 否              |
| callbackurl | string | 回调URL，转换结束后，会回调该URL，详细见 <a href="#">回调URL</a> | 否              |

请求示例:

```
https://img2doc.market.alicloudapi.com/v1/formula
```

传入的Body为JSON格式，例如：

```
{"url": "http://xxx/xxx.png", "type": "docx"}
```

或者Base64，大小限制**8M**

```
{"file": "iVBORw0KGgoAAAAN.....", "type": "md"}
```

输出文件类型type可取值：**docx, md**。

必须签名才能调用成功，签名见[阿里签名规则](#):

<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/use-cases/call-apis>

支持几乎所有图片格式

返回数据结构:

| 名称     | 类型         | 是否必须返回 | 备注         |
|--------|------------|--------|------------|
| code   | number     | 是      | 10000:请求成功 |
| msg    | string     | 是      |            |
| result | Dictionary | 否      | 成功后返回      |

result:

| 名称    | 类型     | 是否必须返回 | 备注                         |
|-------|--------|--------|----------------------------|
| token | string | 是      | 用于 <a href="#">查询结果</a> 接口 |

返回示例(成功状态):



```
{  
  "code":10000,  
  "msg":",",  
  "result":{"token":"xxx"}  
}
```

返回示例(失败状态):

```
{  
  "code":40001,  
  "msg":"ParmNotRight"  
}
```

# Office文档合并POST

将多个Office文档合并为一个文档，url传入多个文档的地址。支持docx, pptx, xlsx格式。

Header中的Content-Type传入application/json

POST Body传入JSON，格式如下：

请求参数BODY：

| 参数   | 类型       | 备注                                 | 是否必须发送 |
|------|----------|------------------------------------|--------|
| url  | string数组 | 要转换的文档URL数组，支持http(s)，ftp开头        | 是      |
| type | string   | 小写，合并文档的类型，例如docx。如果为空，取第一个url的扩展名 | 否      |

请求示例：

```
https://img2doc.market.alicloudapi.com/v1/mergedoc
```

传入的Body为JSON格式，如下：

```
{"url": ["http://xxx/xxx1.docx", "http://xxx/xxx2.docx"]}
```

必须签名才能调用成功，签名见[阿里签名规则](#)：

<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/use-cases/call-apis>

返回数据结构：

| 名称     | 类型         | 是否必须返回 | 备注         |
|--------|------------|--------|------------|
| code   | number     | 是      | 10000:请求成功 |
| msg    | string     | 是      |            |
| result | Dictionary | 否      | 成功后返回      |

result:

| 名称    | 类型     | 是否必须返回 | 备注                         |
|-------|--------|--------|----------------------------|
| token | string | 是      | 用于 <a href="#">查询结果</a> 接口 |

返回示例(成功状态)：

```
{
  "code":10000,
  "msg": "",
  "result":{"token":"xxx"}
}
```

返回示例(失败状态)：

```
{
  "code":40001,
  "msg":"ParmNotRight"
}
```

# 单图转换GET\_v1

将一张图片下载链接url转换为其他格式，type是目标文档的类型，比如要转为docx，type就是docx

请求参数:

| 参数           | 类型     | 备注  | 是否必须发送 |
|--------------|--------|---|--------|
| url          | string | 文件url, 必须http(s), ftp开头, 需要URL Encoding   | 是      |
| type         | string | 小写, 需转换为的文件类型, 例如docx   | 是      |
| language     | int    | OCR识别语言选项, 默认2简体中文:<br>1: 英语<br>2: 简体中文<br>3: 繁体中文<br>4: 法语<br>5: 德语<br>6: 意大利语<br>7: 俄语<br>8: 日文<br>9: 韩文<br>10: 西班牙语<br>11: 葡萄牙语<br>12: 丹麦语<br>13: 荷兰语<br>14: 芬兰语<br>15: 挪威语<br>16: 瑞典语<br>17: 土耳其语 | 否      |
| clean        | int    | 是否清除图像背景只显示文字:<br>默认0: 智能保留背景, 尽量让背景不影响文字编辑<br>1是, 完全清除背景<br>2否, 完全保留背景   | 否      |
| table        | int    | 是否识别表格 (转为Word或PPT时有效)<br>默认0: 否<br>1: 只识别表格<br>2: 识别表格和下划线   | 否      |
| a4           | int    | 是否输出A4大小文件, 默认0否, 1是。仅在pagesize为0时生效  | 否      |
| pagesize     | int    | 设定页面大小。默认0:维持原图比例。1:A3, 2:A4, 3:A5, 4:B4, 5:B5, 6:Letter, 7:Legal, 8:Tabloid, 9:Ledger<br>当不为0时, a4参数失效   | 否      |
| split2p      | int    | 横向图片若有左右两部分 (常见于试卷), 分割为2页。默认0:否, 1是  | 否      |
| norotate     | int    | 是否自动根据文字方向旋转图片, 默认1: 不旋转, 0: 自动判断, 如果确定图片中的文字方向则传1, 速度更快  | 否      |
| textnoformat | int    | 如果转为Txt文本, 默认0: 保持原有布局, 1: 无需布局   | 否      |
| outfilename  | string | 生成的文件的文件名, 默认随机   | 否      |
| callbackurl  | string | 回调URL, 转换结束后, 会回调该URL, 需要URL Encoding, 详细见 回调URL  | 否      |

请求示例:

```
https://img2doc.market.alicloudapi.com/v1/convert?url=https%3a%2f%2fxxx%2fxxx.jpg&type=docx&ocr=0
```

将所在url地址的jpg文件转为docx，type就是需转换为的文件类型，这个例子里就是docx

输出文件类型type可取值：**doc, docx, pptx, xlsx, rtf, txt, ofd, json。**

**必须签名才能调用成功，签名见[阿里签名规则](#)：**

<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/use-cases/call-apis>

**返回数据结构：**

| 名称     | 类型         | 是否必须返回 | 备注         |
|--------|------------|--------|------------|
| code   | number     | 是      | 10000:请求成功 |
| msg    | string     | 是      |            |
| result | Dictionary | 否      | 成功后返回      |

result:

| 名称    | 类型     | 是否必须返回 | 备注                         |
|-------|--------|--------|----------------------------|
| token | string | 是      | 用于 <a href="#">查询结果</a> 接口 |

**返回示例(成功状态):**

```
{
  "code":10000,
  "msg":",
  "result":{"token":"xxx"}
}
```

**返回示例(失败状态):**

```
{
  "code":40001,
  "msg":"ParmNotRight"
}
```

# 单图转换POST\_v1

直接将单个图片POST到服务器，大小限制**8M**

请求参数:

| 参数           | 类型     | 备注   | 是否必须发送 |
|--------------|--------|--|--------|
| file         | file   | 要转换的文档，Content-Type使用multipart/form-data，最大8M  | 是      |
| type         | string | 小写，需转换为的文件类型，例如docx  | 是      |
| language     | int    | OCR识别语言选项，默认2简体中文，值同 <a href="#">单图转换GET_v1</a>  | 否      |
| clean        | int    | 是否清除图像背景只显示文字：<br>默认0：智能保留背景，尽量让背景不影响文字编辑<br>1是，完全清除背景<br>2否，完全保留背景                                    | 否      |
| table        | int    | 是否识别表格（转为Word或PPT时有效）<br>默认0：否<br>1: 只识别表格<br>2: 识别表格和下划线  | 否      |
| a4           | int    | 是否输出A4大小文件，默认0否，1是。仅在pagesize为0时生效   | 否      |
| pagesize     | int    | 设定页面大小。默认0:维持原图比例。1:A3, 2:A4, 3:A5, 4:B4, 5:B5, 6:Letter, 7:Legal, 8:Tabloid, 9:Ledger<br>当不为0时，a4参数失效 | 否      |
| split2p      | int    | 横向图片若有左右两部分（常见于试卷），分割为2页。默认0:否，1是  | 否      |
| norotate     | int    | 是否自动根据文字方向旋转图片，默认1: 不旋转，0: 自动判断，如果确定图片中的文字方向则传1，速度更快   | 否      |
| textnoformat | int    | 如果转为Txt文本，默认0：保持原有布局，1：无需布局  | 否      |
| outfilename  | string | 生成的文件的文件名，默认随机   | 否      |
| callbackurl  | string | 回调URL，转换结束后，会回调该URL，详见 <a href="#">回调URL</a>   | 否      |

请求示例:

```
https://img2doc.market.alicloudapi.com/v1/convert_post
```

Header中的Content-Type必须是multipart/form-data

输出文件类型type可取值：**doc, docx, pptx, xlsx, rtf, txt, ofd, json。**

必须签名才能调用成功，签名见[阿里签名规则](#):

<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/use-cases/call-apis>

| 名称     | 类型         | 是否必须返回 | 备注         |
|--------|------------|--------|------------|
| code   | number     | 是      | 10000:请求成功 |
| msg    | string     | 是      |            |
| result | Dictionary | 否      | 成功后返回      |

result:

| 名称    | 类型     | 是否必须返回 | 备注                         |
|-------|--------|--------|----------------------------|
| token | string | 是      | 用于 <a href="#">查询结果</a> 接口 |

返回示例(成功状态):

```
{
  "code":10000,
  "msg": "",
  "result":{"token":"xxx"}
}
```

返回示例(失败状态):

```
{
  "code":40001,
  "msg":"ParmNotRight"
}
```

# 多图转换POST\_v1

将多张图片转换为一个文档，url传入多张图片的地址，传入非图片格式无效。单次最多50张图片，不能有长图。

Header中的Content-Type传入application/json

POST Body传入JSON，格式如下：

请求参数BODY：

| 参数            | 类型       | 备注   | 是否必须发送 |
|---------------|----------|--|--------|
| url           | string数组 | 要转换的图片URL数组，支持http(s)，ftp开头  | 是      |
| type          | string   | 小写，需转换为的文件类型，例如docx  | 是      |
| language      | int      | OCR识别语言选项，默认2简体中文，值同 <a href="#">单图转换GET_v1</a>  | 否      |
| clean         | int      | 是否清除图像背景只显示文字：<br>默认0：智能保留背景，尽量让背景不影响文字编辑<br>1是，完全清除背景<br>2否，完全保留背景                            | 否      |
| table         | int      | 是否识别表格（转为Word或PPT时有效）<br>默认0：否<br>1：只识别表格<br>2：识别表格和下划线  | 否      |
| a4            | int      | 是否输出A4大小文件，默认0否，1是。仅在pagesize为0时生效   | 否      |
| pagesize      | int      | 设定页面大小。默认0:维持原图比例。1:A3，2:A4，3:A5，4:B4，5:B5，6:Letter，7:Legal，8:Tabloid，9:Ledger<br>当不为0时，a4参数失效 | 否      |
| split2p       | int      | 横向图片若有左右两部分（常见于试卷），分割为2页。默认0:否，1是  | 否      |
| norotate      | int      | 是否自动根据文字方向旋转图片，默认1: 不旋转，0: 自动判断，如果确定图片中的文字方向则传1，速度更快   | 否      |
| excelonesheet | int      | 如果转为Excel文件，默认0：每张图一个工作表；1：合成一个工作表   | 否      |
| textnoformat  | int      | 如果转为Txt文本，默认0：保持原有布局，1：无需布局  | 否      |
| textperpage   | int      | 如果转为Txt文本，默认0：所有页转为单个txt文件，1：每一页转为一个txt文件，并且打包为一个zip包  | 否      |
| outfilename   | string   | 生成的文件的文件名，默认随机   | 否      |
| callbackurl   | string   | 回调URL，转换结束后，会回调该URL，详细见 <a href="#">回调URL</a>  | 否      |

请求示例：

```
https://img2doc.market.alicloudapi.com/v1/convert
```

传入的Body为JSON格式，如下：

```
{ "url": [ "http://xxx/xxx1.png", "http://xxx/xxx2.png" ], "type": "docx", "clean": 0, "a4": 0 }
```

输出文件类型type可取值：**doc, docx, pptx, xlsx, rtf, txt, ofd。**

必须签名才能调用成功，签名见[阿里签名规则](#)：

<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/use-cases/call-apis>

支持几乎所有图片格式

返回数据结构:

| 名称     | 类型         | 是否必须返回 | 备注         |
|--------|------------|--------|------------|
| code   | number     | 是      | 10000:请求成功 |
| msg    | string     | 是      |            |
| result | Dictionary | 否      | 成功后返回      |

result:

| 名称    | 类型     | 是否必须返回 | 备注                         |
|-------|--------|--------|----------------------------|
| token | string | 是      | 用于 <a href="#">查询结果</a> 接口 |

返回示例(成功状态):

```
{
  "code":10000,
  "msg":",
  "result":{"token":"xxx"}
}
```

返回示例(失败状态):

```
{
  "code":40001,
  "msg":"ParmNotRight"
}
```



OCR支持语言

| 语言     | language_code | 语言            | language_code | 语言               | language_code |
|--------|---------------|---------------|---------------|------------------|---------------|
| 中文     | ch            | 英文            | en            | 法文               | fr            |
| 德文     | de            | 日文            | japan         | 韩文               | korean        |
| 中文繁体   | chinese_cht   | 意大利文          | it            | 西班牙文             | es            |
| 葡萄牙文   | pt            | 俄罗斯文          | ru            | 阿拉伯文             | ar            |
| 印地文    | hi            | 维吾尔           | ug            | 波斯文              | fa            |
| 乌尔都文   | ur            | 塞尔维亚文 (latin) | rs_latin      | 欧西坦文             | oc            |
| 马拉地文   | mr            | 尼泊尔文          | ne            | 塞尔维亚文 (cyrillic) | rs_cyrillic   |
| 毛利文    | mi            | 马来文           | ms            | 马耳他文             | mt            |
| 荷兰文    | nl            | 挪威文           | no            | 波兰文              | pl            |
| 罗马尼亚文  | ro            | 斯洛伐克文         | sk            | 斯洛文尼亚文           | sl            |
| 阿尔巴尼亚文 | sq            | 瑞典文           | sv            | 西瓦希里文            | sw            |
| 塔加洛文   | tl            | 土耳其文          | tr            | 乌兹别克文            | uz            |
| 越南文    | vi            | 蒙古文           | mn            | 车臣文              | che           |
| 哈里亚纳语  | bgc           | 保加利亚文         | bg            | 乌克兰文             | uk            |
| 白俄罗斯文  | be            | 泰卢固文          | te            | 阿巴扎文             | abq           |
| 泰米尔文   | ta            | 南非荷兰文         | af            | 阿塞拜疆文            | az            |
| 波斯尼亚文  | bs            | 捷克文           | cs            | 威尔士文             | cy            |
| 丹麦文    | da            | 爱沙尼亚文         | et            | 爱尔兰文             | ga            |
| 克罗地亚文  | hr            | 匈牙利文          | hu            | 印尼文              | id            |
| 冰岛文    | is            | 库尔德文          | ku            | 立陶宛文             | lt            |
| 拉脱维亚文  | lv            | 达尔瓦文          | dar           | 因古什文             | inh           |
| 拉克文    | lbe           | 莱兹甘文          | lez           | 塔巴萨兰文            | tab           |
| 比尔哈文   | bh            | 迈蒂利文          | mai           | 昂加文              | ang           |
| 孟加拉文   | bho           | 摩揭陀文          | mah           | 那格浦尔文            | sck           |
| 尼瓦尔文   | new           | 保加利亚文         | gom           | 梵文               | sa            |
| 阿瓦尔文   | ava           | 阿迪赫文          | ady           | 巴利文              | pi            |
| 拉丁文    | la            |               |               |                  |               |

## 输出JSON结构

样例：

```
{
  "image_dewarped": "base64字符串",
  "texts": [
    {
      "content": "发票号码：12345678",
      "bbox": [100, 150, 300, 180]
    },
    {
      "content": "金额：¥888.00",
      "bbox": [200, 250, 350, 280]
    }
  ],
  "bars": [
    {
      "content": "123456789012",
      "bbox": [400, 100, 560, 180],
      "type": "EAN13",
      "image": "base64字符串"
    }
  ],
  "pictures": [
    {
      "bbox": [50, 400, 200, 600],
      "image": "base64字符串"
    }
  ],
  "tables": [
    {
      "bbox": [100, 700, 700, 1200]
    }
  ],
  "lines": [
    {
      "bbox": [80, 200, 400, 210]
    },
    {
      "bbox": [80, 850, 700, 860]
    }
  ]
}
```

## 说明

- JSON里的坐标左上方为原点(0,0)，bbox代表矩形左上和右下坐标。
- 如果打开dewarp参数，则会产生新的dewarp图片，**坐标基于该图**。dewarp图片会存在image\_dewarped字段，base64形式。
- 如果关闭dewarp参数，则**坐标基于原图**，image\_dewarped字段不再有价值。
- content字段是OCR后的文字内容。
- texts是OCR后的文字。
- bars支持条码和二维码，type返回值QRCODE, EAN13, EAN8, UPCA, UPCE, CODE39, CODE93, CODE128, ITF, CODABAR, PDF417, DATA MATRIX, AZTEC
- pictures是图片区域，image字段是base64字符串。
- tables是表格区域。
- lines是横线，bbox是两点坐标。

备注：

- 文件url方式支持文件大小 1500M。
- 多张图片转换为文档单次最高 50张 图片。
- 非长图，长或宽不得超过 8192像素。支持单张长图。
- 公式识别，图片宽高范围 30-4000 像素。
- 输出JSON仅支持单图，图片长或宽不得超过 8192像素
- 转换完成后，在 1小时 内下载文件。

回调URL：

**用途：** 客户可以自行部署服务器，系统转换结束后会调用客户提供的回调URL，直接发送转换结果，从而无需再轮询[查询结果](#)。

当设置了回调URL，转换结束后（无论成功失败），系统都会尝试调用该URL，具体如下：

以POST方式调用该URL，Header头中Content-Type: application/json

Body为JSON格式，内容和[查询结果](#)的结果相同，例如：

```
{
  "code":10000,
  "msg": "",
  "token": "xxx",
  "result":
  {
    "status": "Done",
    "fileurl": "https://file.duhuitech.com/o/xxx/xxx.docx",
    "filesize": 17747,
    "count": 1
  }
}
```

服务端收到该POST后需在10秒内返回HTTP STATUS CODE 200，视为调用成功，否则系统认为回调失败，会再次尝试。规则如下：

系统共计最多会调用3次回调URL，如果第一次失败，则等待3秒后尝试第二次，如果第二次失败，则等待5秒后尝试第三次，如果第三次失败，则不再尝试。

回调URL超时时间10秒。

阿里云独有部分：

支持从阿里云OSS内网直接下载文件，目前支持的是上海地区的阿里云OSS内网：

oss-cn-shanghai-internal.aliyuncs.com

文档转换GET或多张图片转换POST里的url地址包含上述域名则自动支持

错误码表：

返回的code如果是10000，代表成功，其余是失败

| JSON里返回的code | 错误信息    |
|--------------|---------|
| 40000        | 通用错误    |
| 40001        | 参数错误    |
| 40002        | 参数不符合规范 |
| 40500        | 同步调用超时  |

## 附录：阿里签名方式

参考链接：

<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/use-cases/call-apis>

在调用API商品时，首先您需要了解采用哪种API认证方式，云市场API商品的认证方式主要有以下两种方式。两种方式可同时使用，您可以根据不同情况来选择。

- 简单身份认证（AppCode）
- 签名认证

### 简单身份认证（AppCode）

简单认证（AppCode）调用API，有两种方式，一种是将AppCode放在Header中进行调用，一种是将AppCode放在Query参数中进行调用。

方式一：将AppCode放在Header中

在请求Header中添加一个Authorization参数。

Authorization字段的值的格式为APPCODE + 半角空格 + APPCODE值。格式如下：

Authorization:APPCODE AppCode值

示例：

Authorization:APPCODE 3F2504E04F8911D39A0C0305E82C3301

方式二：将AppCode放在Query中

在请求Query中添加AppCode参数（同时支持appcode , appCode , APPCODE , APPCode四种写法）。

AppCode参数的值为AppCode的值。

示例：

<http://www.aliyum.com?AppCode=3F2504E04F8911D39A0C0305E82C3301>

参考链接：<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/user-guide/call-an-api-operation-by-using-an-appcode>

### 签名认证

比较复杂，推荐用阿里自己的SDK来调用，参考链接：<https://help.aliyun.com/zh/api-gateway/traditional-api-gateway/user-guide/use-digest-authentication-to-call-an-api>