

度慧解文 API (阿里) 概述:

在线试用:

<https://docudeep.com/>

小程序搜索:

解文识图

使用方式:

第一步: 用 文档添加 GET 或 文档添加 POST 添加文档, 成功后获得 token 和 owner, **添加文档是异步操作, 所以需要第二步查询添加结果**

第二步: 使用该 token 去轮询文档添加状态, 见 **查询文档添加结果** QUERY, 直到添加完成。也可以设置 callbackurl, 当添加结束后, 系统会回调该 URL 直接推送添加结果。详细见: **回调 URL**

第三步: 使用该 token 对文档提问, 见 **文档提问 GET**

API 调用需要签名, 详细见文档附录: **阿里签名**

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**

文档添加 / 替换 GET

提供文档下载地址 url 来添加, type 是源文档的 type, 比如要添加 docx, type 就是 docx, 默认 type 可以不填, 会自动使用 url 的文件扩展名, 但是如果 url 里的文件扩展名无法解析出来, 则会报错。

请求参数:

参数	类型及范围	备注	是否必须发送
url	string	要添加的文档 URL, 需要 URLEncoding, 必须 http(s), ftp 开头	是
type	string	要添加的文档扩展名, 默认不填取 url 的文件扩展名	否
password	string	文档的密码, 默认空无密码	否
language	int	如果是图片, OCR 识别语言选项, 默认 2 简体中文: 1: 英语 2: 简体中文 3: 繁体中文 4: 法语	否

参数	类型及范围	备注	是否必须发送
		5: 德语 6: 意大利语 7: 俄语 8: 日文 9: 韩文 10: 西班牙语 11: 葡萄牙语 12: 丹麦语 13: 荷兰语 14: 芬兰语 15: 挪威语 16: 瑞典语 17: 土耳其语	
callbackurl	string	回调 URL, 添加结束后, 会回调该 URL, 需要 URL Encoding, 详细见文档 回调 URL	否
owner	string	此参数有两种用途: 当添加文档时, 允许指定文档的 owner 当替换文档时, 必须传入文档的 owner	否
token	string	当替换文档时, 必须传入文档的 token	否
share	int	如果需要分享文档的 URL, 设为: 1。默认: 0	否

请求示例:

<https://gpt.market.alicloudapi.com/v1/add?url=https%3a%2f%2fxxx%2fxxx.docx&type=docx>

添加所在 url 地址的 docx 文件, type 就是源文件的 type, 这个例子里就是 docx。

文档添加替换的接口都可以添加也可以替换文档。当添加文档时, 如果指定了 **owner**, 那么返回的 owner 就是指定值, 如果不指定, 那么 owner 返回随机值。如要替换文档, 必须指定需替换文档的 token 和 owner。

请妥善保管 owner 作为替换, 删除文档的唯一凭证, 最好保留在服务器端避免泄漏。之后涉及到 owner 的地方都与此相同, 不再赘述。

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>

支持多种文件格式，具体如下（type 可传入如下格式）：

PDF 文件: pdf

微软 Office 文档: doc, docx, ppt, pptx, xls, xlsx, pot, pps, ppsx

WPS 文档: wps, wpt, dps, dpt, et, ett

苹果 iWork 文档: pages, key, numbers

开放版式文档: ofd

电子刊物: caj, nh, kdh

电子书: epub, chm, mobi, azw, azw3, fb2, cbr, cbz, djvu

Markdown 格式: md

SVG 格式: svg

网页文件: html, htm, mht, eml

图片文件: 几乎所有格式例如 png, jpg, jpeg, gif, tif, tiff, bmp, 可以统一传 img, 代表一切图片

文本文件: txt, rtf, java, js, c, cpp, jsp, css, xml, properties, log, 其他任意文本文件都可以传 txt

网址网页: url (网址, 例如: <https://www.duhuitech.com>)

返回数据结构:

名称	含义	类型及范围	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	含义	类型及范围	是否必须返回	备注
token		string	是	用于 query 接口
owner	替换, 删除文档的唯一凭证	string	是	用于文档删除 delete 接口

返回示例(成功状态):

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

返回示例(失败状态):

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

文档添加 / 替换 POST

直接将单个文档 POST 到服务器，大小限制 8M

请求参数:

参数	类型及范围	备注	是否必须发送
file	file	要添加的文档，Content-Type 使用 multipart/form-data，最大 8M	是
type	string	要添加的文档扩展名，默认不填取 file 的文件扩展名	否
password	string	文档的密码，默认空无密码	否
language	int	如果是图片，OCR 识别语言选项， 默认 2 简体中文： 1: 英语 2: 简体中文 3: 繁体中文 4: 法语 5: 德语 6: 意大利语 7: 俄语 8: 日文 9: 韩文	否

参数	类型及范围	备注	是否必须发送
		10: 西班牙语 11: 葡萄牙语 12: 丹麦语 13: 荷兰语 14: 芬兰语 15: 挪威语 16: 瑞典语 17: 土耳其语	
callbackurl	string	回调 URL, 添加结束后, 会回调该 URL, 需要 URL Encoding, 详细见文档 回调 URL	否
owner	string	此参数有两种用途: 当添加文档时, 允许指定文档的 owner 当替换文档时, 必须传入文档的 owner	否
token	string	当替换文档时, 必须传入文档的 token	否
share	int	如果需要分享文档的 URL, 设为: 1。默认: 0	否

请求示例:

<https://gpt.market.alicloudapi.com/v1/add>

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

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>

支持多种文件格式, 具体如下 (type 可传入如下格式) :

PDF 文件: pdf

微软 Office 文档: doc, docx, ppt, pptx, xls, xlsx, pot, pps, ppsx

WPS 文档: wps, wpt, dps, dpt, et, ett

苹果 iWork 文档: pages, key, numbers

开放版式文档: ofd

电子刊物: caj, nh, kdh

电子书: epub, chm, mobi, azw, azw3, fb2, cbr, cbz, djvu

Markdown 格式: md

SVG 格式: svg

网页文件: html, htm, mht, eml

图片文件: 几乎所有格式例如 png, jpg, jpeg, gif, tif, tiff, bmp, 可以统一传 img, 代表一切图片

文本文件: txt, rtf, java, js, c, cpp, jsp, css, xml, properties, log, 其他任意文本文件都可以传 txt

返回数据结构:

名称	含义	类型及范围	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	含义	类型及范围	是否必须返回	备注
token		string	是	用于 query 接口
owner	替换, 删除文档的唯一凭证	string	是	用于文档删除 delete 接口

返回示例(成功状态):

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

返回示例(失败状态):

```
{
```

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

多张图片添加 / 替换 POST

添加多张图片，传入每张图片的地址，**传入非图片格式无效，最多 50 张图片，或 1 张长图**

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	类型及范围	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
language	int	如果是图片，OCR 识别语言选项，默认 2 简体中文： 1: 英语 2: 简体中文 3: 繁体中文 4: 法语 5: 德语 6: 意大利语 7: 俄语 8: 日文 9: 韩文 10: 西班牙语 11: 葡萄牙语 12: 丹麦语 13: 荷兰语 14: 芬兰语 15: 挪威语 16: 瑞典语 17: 土耳其语	否

参数	类型及范围	备注	是否必须发送
callbackurl	string	回调 URL, 添加结束后, 会回调该 URL, 需要 URL Encoding, 详细见文档 回调 URL	否
owner	string	此参数有两种用途: 当添加文档时, 允许指定文档的 owner 当替换文档时, 必须传入文档的 owner	否
token	string	当替换文档时, 必须传入文档的 token	否
share	int	如果需要分享文档的 URL, 设为: 1。默认: 0	否

请求示例:

https://gpt.market.alicloudapi.com/v1/add_images

传入的 Body 为 JSON 格式, 如下:

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

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	含义	类型及范围	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	含义	类型及范围	是否必须返回	备注
token		string	是	用于 query 接口

名称	含义	类型及范围	是否必须返回	备注
owner	替换, 删除文档的唯一凭证	string	是	用于文档删除 delete 接口

返回示例(成功状态):

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

返回示例(失败状态):

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

查询文档添加结果 QUERY

请求参数:

参数	类型及范围	备注	是否必须发送
token	string		是

请求示例:

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

无需签名, 无调用次数限制

由于添加需要时间, 文件越大页数越多, 添加越久, 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, 则添加结束, 停止轮询。如果是 Doing 或 Pending, 则继续轮询。

返回数据结构:

名称	含义	类型及范围	是否必须返回	备注
----	----	-------	--------	----

名称	含义	类型及范围	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	类型及范围	是否必须返回	备注
status	状态	string	是	Pending: 还未开始 Doing: 正在转换 Done: 转换成功 Failed: 转换失败
progress	进度	number(0.00 - 1.00)	否 (status 为 Doing 时返回)	比如 0.88 表示 88%
count	总页数	integer	否 (status 为 Doing 或 Done 时返回)	页面总数
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":{}
```

```

"token":"xxx",
"result":
{
    "status":"Done",
    "count":10
}
}

```

返回示例(失败状态):

```

{
    "code":40400,
    "msg":"No such token",
    "token":"xxx"
}

```

文档提问 GET / POST

对文档进行提问，总结，信息提取，翻译等。其中提问通过 parentid 参数支持上下文。

请求参数:

参数	类型及范围	备注	是否必须发送
token	string	文档添加成功后获得的 token	是
action	string	提问类型, question: 自由提问 summary: 总结 keyword: 关键词 oneword: 一句话 title: 起标题 extract: 提取信息 translation: 翻译 classification: 分类 tone: 语气 mood: 心情 create_table: 创建表格 create_outline: 创建大纲 create_category: 创建目录 create_todo: 创建待办	是

参数	类型及范围	备注	是否必须发送
		create_question: 创建问题 create_qa: 创建问答 create_note: 创建笔记 custom: 提示词自定义, 见 subparm 参数说明	
parm	string	提问参数, 比如 action 是 question, 这个参数就是实际的问题。 对于提取信息 extract, 就是提取哪种信息。 对于分类 classification, 就是按照什么来分类。 以上三个 action 都需要传这个参数。其余 action 可传可不传, 传的话就是关于什么, 比如 keyword 类型传智能就是偏向智能的关键词。 GET 方式需要 urlencoding	否
pageindex	int	除了 question 以外, 其他 action 都需要传页数, action 的范围是该页内容。如果传 0 会尽可能的寻找 parm 指定的相关内容。	否
language	string	返回的语言, 默认 cn: 中文, 其它直接填写语言名即可, 比如: 英语	否
markdown	int	是否输出 markdown 格式, 默认 0 否, 1 是	否
json	int	是否输出 json 格式, 默认 0 否, 1 是	否
jsonschema	string	自定义输出 json 格式, 默认空。详细见附录: 自定义 json 格式 如果自定义 schema, streaming 将失效 GET 方式需要 urlencoding	否
nolimit	int	是否输出不局限于原文, 默认	否

参数	类型及范围	备注	是否必须发送
		1 是, 0 否	
temperature	float	较高的数值会使输出更加随机, 取值范围 0.0–1.0。默认 0.1, 中间值 0.5	否
reasoning	int	是否开启深度思考, 默认 0 否, 1 是	否
reasoning_effort	int	思考深度, 默认 0 低, 1 中, 2 高	否
websearch	int	是否开启联网搜索, 默认 0 否, 1 是	否
subparm	string	<p>高级参数, 可自定义提示词</p> <p>提问参数 <code>parm</code> 的补充。仅在 <code>action</code> 是 <code>custom</code> 时生效。</p> <p><code>parm</code> 可以是提出的问题。</p> <p><code>subparm</code> 可以是如何输出的指令。</p> <p>通过 <code>parm</code> 寻找原文片段, 原文会被包含在提示词的 <code>system</code> 里,</p> <p>输入的提示词则是:</p> <p><code>subparm+parm</code></p> <p>可以用 \n 换行, 用“原文”指代文档。</p> <p>GET 方式需要 urlencoding</p>	否
parentid	string	对于 <code>question</code> 类型, 指定上下文的 <code>id</code> 。第一次调用该值为空, 服务端会返回该值, 后续调用附上该值即可	否
async	int	是否开启异步返回模式, 默认 0 否, 1 是。详情见后面“异步返回模式”	否
callbackurl	string	异步返回模式回调 URL, 回复结束后, 会回调该 URL, 需要 URL Encoding, 回调数据同 <code>查询提问结果</code> 接口, 回调规则见 <code>回调 URL</code>	callbackurl

参数	类型及范围	备注	是否必须发送
stream	int	是否流式 EventSource 输出，即逐字输出， 默认 0 否， 1 是	否

请求示例：

GET:

<https://gpt.market.alicloudapi.com/v1/ask?token=xxx&action=question&parm=%E6%94%AF%E6%8C%81%E5%8E%8B%E7%BC%A9%E5%90%97%EF%BC%9F>

POST:

<https://gpt.market.alicloudapi.com/v1/ask>

由于参数在 body 里，所以可以不需要做 **urlencoding**

如果参数较长，建议 POST 方式，GET 方式参数长度有限制

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: (stream 为 0 时)

名称	含义	类型及范围	是否必须返回	备注
code		number	是	10000: 请求成功
msg		string	是	
token		string	是	输入的 token
result		Dictionary	否	成功后返回

result:

名称	含义	类型及范围	是否必须返回	备注
answer		string	是	提问的回复
parentid		string	否	仅当 action 是 提问 question 才返回

返回示例(成功状态)：

{

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

返回示例(失败状态):

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

返回数据结构: (stream 为 1 时)

返回 content-type: text/event-stream
Type: message
Data: 'xxx'
注意 data 里的数据前后被 ' 包围, 使用时要去掉 '

如果 action 是提问 question, 最后一行会返回 parentid:
Type: parentid
Data: xxx
注意 data 里没有 ' 包围

异步返回模式:

当开启了 async=1, 那么 api 会立即返回 queryid, 同时不会进行 stream 流式输出:

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

然后通过轮询 **查询提问结果 QUERY** 接口, 来获得结果。

异步查询提问结果 QUERY

请求参数:

参数	类型及范围	备注	是否必须发送
token	string		是
queryid	string	从文档提问接口的异步模式获得	是

请求示例:

```
https://gpt.market.alicloudapi.com/v1/result?token=xxx&queryid=xxx
```

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>
查询频率建议 3-5 秒。查询后先看 status，如果是 Done 或 Failed，则结束，停止轮询。如果是 Doing，则继续轮询。

返回数据结构:

名称	含义	类型及范围	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	类型及范围	是否必须返回	备注
status	状态	string	是	Doing: 正在回答 Done: 完成回答 Failed: 回答失败
queryid	查询 id	string	是	查询的 queryid
content	回答内容	string	否 (status 为 Done 时返回)	提问结果
parentid	上下文 id	string	否 (status 为 Done 时返回)	用于 question 类型的上下文 id

返回示例(成功状态):

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

```
{  
    "code":10000,  
    "msg": "",  
    "token": "xxx",  
    "result":  
    {  
        "status": "Done",  
        "queryid": "xxx",  
        "content": "xxx",  
        "parentid": "xxx"  
    }  
}
```

返回示例(失败状态):

```
{  
    "code":40500,  
    "msg": "NoSuchQueryId",  
    "token": "xxx"  
}
```

文档分享 GET

分享文档的 URL，获得该 URL 的用户都可以提问。

请求参数:

参数	类型及范围	备注	是否必须发送
----	-------	----	--------

参数	类型及范围	备注	是否必须发送
token	string	文档添加成功后获得的 token	是
owner	string	文档添加成功后获得的 owner	是
appcode	string	阿里云购买的 appcode	是
filename	string	分享页显示的文件名, 需要 urlencoding	是
showdoc	int	分享页是否显示原文件, 默认 1: 是。0: 否	否

请求示例:

```
https://gpt.market.alicloudapi.com/v1/share?token=xxx&owner=xxx&appcode=xxx&filename=xxx
```

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	含义	类型及范围	是否必须返回	备注
code		number	是	10000: 请求成功
msg		string	是	
token		string	是	输入的 token
result		Dictionary	否	成功后返回

result:

名称	含义	类型及范围	是否必须返回	备注
shareurl	分享的 URL	string	是	

返回示例(成功状态):

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

```
    "msg":"",
    "token":"xxx",
    "result":
    {
        "shareurl":"xxx"
    }
}
```

返回示例(失败状态):

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

文档删除 GET

删除文档，对不需要再次提问的文档，可以删除。

请求参数:

参数	类型及范围	备注	是否必须发送
token	string	文档添加成功后获得的 token	是
owner	string	文档添加成功后获得的 owner	是

请求示例:

```
https://gpt.market.alicloudapi.com/v1/delete?token=xxx&owner=xxx
```

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	含义	类型及范围	是否必须返回	备注
code		number	是	10000:请求成功
msg		string	是	
token		string	是	输入的 token

返回示例(成功状态)：

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

返回示例(失败状态)：

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

注意：

- 上传文件单个**最大 1500M**
- 多张图片一次**最多 50 张**
- 支持**单张长图**
- 文件**有条件**永久保留，如果文件 1 个月内没有提问，则会被自动删除

回调 URL：

用途：客户可以自行部署服务器，文件添加结束后会调用客户提供的回调 URL，直接发送添加结果，从而无需再轮询 Query。

当设置了回调 URL，添加结束后（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 Query 的结果相同，例如：

```
{"code":10000,"msg":"","token":"xxx","result":{"status":"Done","count":10}}
```

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 里的 url 地址包含上述域名则自动支持

错误码表：

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
40400	找不到输入的 token
40401	没有权限
40500	找不到输入的 queryid
40501	找不到分享页
40502	添加文件时需要先打开分享
40503	阿里云 appcode 错误

附录：自定义输出 JSON 格式

支持的数据类型：

String
Number
Boolean
Integer
Object
Array
Enum
anyOf

以下是每个类型及用法

String

说明：表示字符串类型。

写法：填写 “str” 或 “String”。

示例：
 {
 "companyName": "str"
 }

Number

说明：表示数字类型（浮点数）。

写法：填写 “Number” 或 “float”。

示例：
 {
 "revenue": "Number"
 }

Boolean

说明：表示布尔值（True/False）。

写法：填写 “Boolean” 或 “bool”。

示例：
 {
 "active": "Boolean"
 }

Integer

说明：表示整数类型。

写法：填写 “Integer” 或 “int”。

示例：
 {
 "Founded": "Integer"
 }

Object

说明：用于描述嵌套对象，即由若干字段构成的子模型。

写法：直接使用一个字典，键为子字段名称，值为其类型定义。

示例：

```
  "headquarters": {  
    "address": "str",  
    "openHours": {  
      "weekdays": "list[str]",  
      "weekend": "list[str]"  
    }  
  }  
}
```

Array

说明：用于描述数组类型，所有元素必须为同一种类型。

写法：有两种方式：

使用列表形式，列表内只包含一个类型定义，如 `["str"]`，表示每项是字符串。

示例：

```
  "tags": ["str"]  
}
```

或者使用简写字符串形式，直接写成 `"list[子类型]"`，例如 `"list[str]"`。

示例：

```
  "skills": "list[str]"  
}
```

Enum

说明：用于限定取值范围，只允许取预定义的几个值。

写法：使用 `"Enum[值 1, 值 2, 值 3]"` 格式。

示例：

```
  "status": "Enum[planned, active, completed]"  
}
```

anyOf

说明：表示联合类型，字段的值可以是多种类型中的任意一种。

写法：使用 `"anyOf[类型 1, 类型 2, ...]"` 格式。

示例：

```
  "score": "anyOf[int, Number]"  
}
```

完整示例

下面是一个完整例子，该定义描述了一个公司信息模型，含多层嵌套的对象和数组：

```
{  
    "companyName": "str",  
    "founded": "Integer",  
    "departments": [ # 部门数组（第一层数组中的每个元素为对象）  
        {  
            "deptName": "str",  
            "budget": "Number",  
            "employees": [ # 员工数组  
                {  
                    "empName": "str",  
                    "age": "Integer",  
                    "skills": "list[str]", # 技能数组  
                    "certifications": { # 认证信息对象，包含数组  
                        "certList": "list[str]",  
                        "valid": "Boolean"  
                    }  
                }  
            ],  
            "projects": [ # 部门项目数组  
                {  
                    "projectName": "str",  
                    "startDate": "str",  
                    "status": "Enum[planned, active, completed]",  
                    "milestones": [ # 里程碑数组（第三层嵌套）  
                        {  
                            "milestoneName": "str",  
                            "dueDate": "str"  
                        }  
                    ]  
                }  
            ],  
            "headquarters": { # 总部信息对象  
                "address": "str",  
                "coordinates": "list[Number]", # 坐标数组  
                "openHours": {  
                    "weekdays": "list[str]",  
                    "weekend": "list[str]"  
                }  
            }  
        }  
    ]  
}
```

公司基本信息（公司名称、成立年份）。

部门数组（第一层数组），每个部门包含部门名称、预算、员工信息和部门项目。

部门中的 员工数组：每个员工具有姓名、年龄、技能数组和认证信息（一个对象，其中包含认证列表数组和是否有效标识）。

部门中的 项目数组：每个项目包含项目名称、开始日期、状态（枚举）以及 里程碑数组（第三层嵌套，每个里程碑具有名称和截止日期）。

总部信息对象，包含地址、坐标数组和营业时间（分别针对工作日和周末）。

使用说明

- 用户需要输入一个 JSON 定义，形式必须为字典（键值对），各字段的值使用上面提到的类型标识。
- 数组类型可以写成列表形式（如 `["str"]`）或用 `"list[...]"` 的字符串形式；注意整个 JSON 定义仍必须是字典。
- 对于枚举类型，请使用 `"Enum[...]"`，不同选项之间用逗号分隔。
- 如果需要联合多种类型，请使用 `"anyOf[...]"` 格式。
- 嵌套对象只需直接写子字段及其类型，无需额外的 `"type"`、`"properties"` 等标识。

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.aliyun.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>

