## 美能华能手产品介绍及说明

## 平台介绍

## 概述

美能华能手平台采用计算机视觉，图像处理，自然语言处理和深度学习等技术，实现对影印文本的智能文字识别和智能表格识别，在识别的同时通过语义分析理解，实现对文档的结构化处理，进行关键内容的抽取和检索功能，公司可以从大量的文档、数据源中提取信息，提高数据获取和信息分析效率，降低人工识别的工作量，针对拥有大量文档以及对数据清洗整理有需求的企业，本系统可以极大地提高其工作效率，从而提升企业效益和收益。

## 应用场景

美能华能手平台识别准确率高超过99%，并根据业务需求支持各类场景的应用：

### 通用文档识别

* 支持单个文件导入和批量导入
* 支持多格式的图片文件JPG、JPEG、PNG、BMP
* 支持PDF文件（文字版和影印版）
* 支持根据业务场景输出多种形式的文件，例如pdf, json等
* 支持图片方向矫正

### 表格识别

* 支持结构化表格识别

### 自定义表单抽取

* 支持自定义模板制作
* 支持输出抽取结果

# 通用接口说明

### 接口说明

**[POST]** https://api-expert.meinenghua.com/v1/api/{workspaceId}

### 请求参数

Path Parameters

|  |  |  |
| --- | --- | --- |
| workspaceId | string | 工作区id |

Query Parameters

|  |  |  |
| --- | --- | --- |
| appKey | string | api应用标识 |
| appSecret | string | api 应用密钥 |

Form Data Parameters

|  |  |  |
| --- | --- | --- |
| default | object | 待识别的图片或pdf |

### 返回结果

●200: OK

具体响应内容详见具体工作区或接口定义， 这里仅为示意

{

"姓名": "韦小宝",

"性别": "男",

"民族": "汉",

"出生": "1654年12020日",

"住址": "北京市东城区景山前街4号紫禁城敬事房",

"公民身份号码": "11204416541220243X",

"行政区划代码": "110101",

"行政区划名称": "北京市东城区",

"详细地址": "景山前街4号紫禁城敬事房"

}

●400: Bad Request

当发生错误时，状态码会大于等于400

|  |  |  |
| --- | --- | --- |
| errorCode | string | 请求内部错误码 |
| errorMessage | string | 错误信息 |

### 请求示例

POST https://api-expert.meinenghua.com/v1/api/{workspaceId}?appKey={appKey}&appSecret={appSecret}

Body: multipart content

# 代码示例

## python

import requests

def request():

workspace\_id='[Workspace Id]'

app\_key='[App Key]'

app\_secret='[App Secret]'

url = 'https://api-expert.meinenghua.com/v1/api/workflows/{0}?appKey={1}&appSecret={2}'.format(workspace\_id, app\_key, app\_secret)

payload = {}

files = [ ('default', open('id\_card.jpg', 'rb')) ]

headers = {

'Accept': 'application/json'

}

response = requests.request('POST', url, headers=headers, data=payload, files=files)

return

## Java

### Java with string response

public void send() {

String workspaceId = "[Workspace Id]";

String appKey = "[App Key]";

String appSecret = "[App Secret]";

String url = "https://api-expert.meinenghua.com/v1/api/workflows/" + workspaceId + "?appKey=" + appKey + "&appSecret=" + appSecret;

String filePath = "[Your File]";

HttpHeaders headers = new HttpHeaders();

MediaType type = MediaType.parseMediaType("multipart/form-data");

headers.setContentType(type);

FileSystemResource fileSystemResource = new FileSystemResource(path);

MultiValueMap<String, Object> form = new LinkedMultiValueMap<>();

form.add("default", fileSystemResource);

HttpEntity<MultiValueMap<String, Object>> files = new HttpEntity<>(form, headers);

String response = restTemplate.postForEntity(url, files, String.class).getBody();

}

### Java with stream response

public void send() throws IOException {

String workspaceId = "[Workspace Id]";

String appKey = "[App Key]";

String appSecret = "[App Secret]";

String url = "https://api-expert.meinenghua.com/v1/api/workflows/" + workspaceId + "?appKey=" + appKey + "&appSecret=" + appSecret

String filePath = "[Your File]";

String destPath = "[Response File Path]";

FileSystemResource fileSystemResource = new FileSystemResource(filePath);

HttpHeaders headers = new HttpHeaders();

MediaType type = MediaType.parseMediaType("multipart/form-data");

headers.setContentType(type);

MultiValueMap<String, Object> form = new LinkedMultiValueMap<>();

form.add("default", fileSystemResource);

HttpEntity<MultiValueMap<String, Object>> files = new HttpEntity<>(form, headers);

RestTemplate restTemplate = new RestTemplate();

ResponseEntity<Resource> exchange = restTemplate.exchange(url, HttpMethod.POST, files, Resource.class);

InputStream inputStream = exchange.getBody().getInputStream();

FileOutputStream outputStream = new FileOutputStream(path);

inputStream.transferTo(outputStream);

outputStream.flush();

}

## javascript

### javascript with XMLHttpRequest (browser)

function (){

const workspace\_id = '[Workspace Id]'

const app\_key = '[App Key]'

const app\_secret = '[App Secret]'

const url =`https://api-expert.meinenghua.com/v1/api/workflows/${workspace\_id}?appKey=${app\_key}&appSecret=${app\_secret}`

const data = new FormData();

const xhr = new XMLHttpRequest();

data.append('default', fileInput.files[0], 'id\_card.jpg');//fileInput存储上传文件

xhr.open('POST', url, true);

xhr.send(data);

xhr.onreadystatechange = function () {

if (xhr.readyState === 4 && xhr.status === 200) {

const response = xhr.responseText;

console.log(response)

return response

}

}

return

}

### javascript with axios (browser)

postFormData () {

const workspace\_id = '[Workspace Id]'

const app\_key = '[App Key]'

const app\_secret = '[App Secret]'

const url = `https://api-expert.meinenghua.com/v1/api/workflows/${workspace\_id}?appKey=${app\_key}&appSecret=${app\_secret}`

const data = new FormData();

data.append('default', fileInput.files[0], 'id\_card.jpg');//fileInput存储上传文件

return new Promise((resolve, reject) => {

axios({

headers: { 'Content-Type': 'multipart/form-data'},

url: url,

method: 'post',

data: data

}).then(res => resolve(res.data))

.catch(err => reject(err))

}

### javascript with es6 (browser)

async postFormData () {

const workspace\_id = '[Workspace Id]'

const app\_key = '[App Key]'

const app\_secret = '[App Secret]'

const url = `https://api-expert.meinenghua.com/v1/api/workflows/${workspace\_id}?appKey=${app\_key}&appSecret=${app\_secret}`

const data = new FormData();

const headers = { 'Content-Type': 'multipart/form-data'}

data.append('default', fileInput.files[0], 'id\_card.jpg');//fileInput存储上传文件

const response = await axios({ headers, url, method: 'post', data })

return response.data

}

### javascript (nodejs)

const request = require('request')

const fs = require('fs')

const post = () => {

const workspace\_id = "[Workspace Id]"

const app\_key = "[App Key]"

const app\_secret = "[App Secret]"

const url =`https://api-expert.meinenghua.com/v1/api/workflows/${workspace\_id}?appKey=${app\_key}&appSecret=${app\_secret}`

const formData = { filename: fs.createReadStream('/id\_card.jpg') }

const headers = { 'Content-Type': 'multipart/form-data' }

const options = { method: 'POST', headers, url, formData},

request.post(options, function(error, response, body) {

if (!error && response.statusCode === 200) {

console.log(error, response, body)

}

})

return

}

## C#

using RestSharp;

static void post(string[] args) {

string workspaceId = "[Workspace Id]";

string appKey = "[App Key]";

string appSecret = "[App Secret]";

string url = "https://api-expert.meinenghua.com/v1/api/workflows/" + workspaceId + "?appKey=" + appKey + "&appSecret=" + appSecret");

try {

var client = new RestClient(url);

client.Timeout = -1;

var request = new RestRequest(Method.POST);

request.AddFile("default", "/id\_card.jpg");

IRestResponse response = client.Execute(request);

return response.Content

} catch (Exception ex) {

throw ex;

}

}

## ruby

require "uri"

require "net/http"

def post()

workspace\_id = "[Workspace Id]"

app\_key = "[App Key]"

app\_secret = "[App Secret]"

url = URI("https://api-expert.meinenghua.com/v1/api/workflows/#{workspace\_id}?appKey=#{app\_key}&appSecret=#{app\_secret}")

https = Net::HTTP.new(url.host, url.port);

https.use\_ssl = true

request = Net::HTTP::Post.new(url)

form\_data = [['default', File.open('/id\_card.jpg')]]

request.set\_form form\_data, 'multipart/form-data'

response = https.request(request)

puts response.read\_body

end

# 数据结构说明

### Region

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| page\_id | String | 分页页码 |
| type | String | 类型，为text时是纯文本，为table时该区域为表格 |
| text | String | 识别文字 |
| range | Range | 识别区域坐标对象 |
| cells | List<Cell> | 表格中单元格列表, 仅当type为table时生效 |

### Range

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| left | Int | 坐标left值 |
| top | Int | 坐标top值 |
| width | Int | 坐标width值 |
| height | Int | 坐标height值 |
| center\_x | Float | 坐标center\_x值 |
| center\_y | Float | 坐标center\_y值 |

### Cell

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 说明 |
| left | Int | 坐标left值 |
| top | Int | 坐标top值 |
| width | Int | 坐标width值 |
| height | Int | 坐标height值 |
| center\_x | Float | 坐标center\_x值 |
| center\_y | Float | 坐标center\_y值 |
| text | String | 单元格中文字内容 |

更多信息也可访问：https://docs.meinenghua.com/#/expert/api/index

## 美能华能手平台内容清单（上海档案局）

## 能手平台信息

美能华能手平台已私有化部署至上海档案局内网环境，访问ip为：172.16.2.135。

能手平台浏览器访问地址：172.16.2.135:8085

能手API访问地址：172.16.2.135:7001

## 能手平台账号信息

账号：[danganju\_admin@admin.com](http://mailto:danganju_admin@admin.com)

密码：123456

## 能手工作区信息

|  |  |  |  |
| --- | --- | --- | --- |
| 工作区名称 | 输入 | 输出 | API |
| 通用文档识别 | 图片或pdf | 1. UI展示OCR识别结果 2. api返回OCR识别结果 3. UI可下载双层pdf | 返回OCR识别json结果  **workspace\_id:** 6f70f0cd-54ad-4d00-a769-b49fff85cf67  **app\_key:** 46e1d6c0-c66e-11eb-921f-4f45561eb7a9  **app\_secret:**46e1d6c1-c66e-11eb-921f-4f45561eb7a9  **限制：**无限制 |
| 表格读取模型 | 图片或pdf | 1. UI展示表结构json 2. UI展示OCR识别结果 3. api返回表结构json | 返回表结构json结果  **workspace\_id:** 81bfe1b6-02ed-4d3f-bc6d-d2d7dee30ea3  **app\_key:** dcf3f5d0-c66e-11eb-93c1-eda14bdcfc67  **app\_secret:** dcf3f5d1-c66e-11eb-93c1-eda14bdcfc67  **限制：**无限制 |
| 表单抽取 | 图片或pdf | 1. UI展示json形式的抽取结果 2. UI可下载excel形式的抽取结果 | 无 |