

# 智能平台api说明

最后更新：2019年1月4日

## /policy/upload\_policy/

上传政策文件

### POST

form-data

key	value
file	政策文件
policy_id	政策id

政策文件格式需求：

目前可支持的格式如下

txt:要求以UTF-8编码txt

doc:要求以UTF-8编码doc（仅doc中的文字内容会被识别）

## /policy/upload\_guide/

### POST

上传指南文件，并将指南文件和对应的的事项以及政策相（系统会对指南文件进行理解）

form-data

输入：

key	value
file	指南文件
guide_id	指南文件id
policy_id	指南依据的政策id

指南文件格式需求：

目前可支持的格式如下

doc:要求以UTF-8编码doc

输返回：

```
{
  status: "状态"
  message: "和状态相关的扩展提示，例如指明失败原因是因为文件格式不支持等"
}
```

## /policy/recommend/

### GET

获取给该企业推荐的事项。

/policy/recommend/?company\_id=123

使用company\_id进行查询的时候，系统首先会将提交一个异步任务到后台，重新计算对该企业推荐哪些政策，异步任务的id放入返回内容中的 task\_id 字段中。查找是否存在这个企业的历史推荐记录，历史记录会放入到返回内容中的 result 字段中，若不存在历史推荐记录，则 result 为一个空的列表。

matching 表示推荐的结果和企业的匹配程度，例如，推荐的指南有10个条件，企业满足9个，则 matching 为 0.9。

reason 是字符串，表示推荐的原因。会列出指南中企业满足的条件。这些条件不一定是原文，但是在语义上一致。

matching 和 reason 是在推荐记录生成，也就是 time 计算出来的。当企业信息发生变化之后这些信息不会在本记录上改变，但是会反映在新的推荐记录上。

例如，在2018-11-01推荐了事项1，匹配度是0.7则会有一条 {"guide\_id": "01", "matching" : 0.70, "time": "2018-11-01 00:00:01"} 的记录。在一个月之后，企业数据发生变化，对于事项1的匹配度变为 0.9，则上述记录还会保留，但是会增加一条 {"guide\_id": "01", "matching" : 0.90, "time": "2018-12-01 00:00:01"} 的记录。

```
{
  "task_id": "异步任务id，可以用于更新推荐的政策",
  "result": [
    {
      "guide_id": "推荐事项的id",
      "reason": "推荐原因，可用于展示",
      "matching" : 0.90,
      "time": "推荐时间, fmt:YYYY-MM-DD HH:mm:ss"
    },
    {}
  ]
}
```

/policy/recommend/?task\_id=123

使用company\_id进行查询后，可以使用拿到的task\_id获取更新后的企业推荐记录。

```
{
  "status": "任务状态",
  "result": [
    {
      "policy_id": "推荐的政策id",
      "reason": "推荐原因",
      "matching" : 0.90,
      "time": "推荐时间, fmt:YYYY-MM-DD"
    },
    {}
  ]
}
```

## /policy/check\_recommend/

### POST

计算多个企业和单个政策的匹配情况。此接口针对政策推送功能。当一个新政策发布之后，平台在上传政策文档之后，调用此接口批量获取新政策与目标企业的匹配情况。匹配情况异步计算，需要提供回调函数url以接受异步任务执行情况。

输入：

json格式

```
{
  "companies": ["企业id1","企业id2",...,"企业idn"],
  "guide_id": "指南id",
  "callback": "回调函数url"
}
```

输出：

```
{
  "task_id": "异步任务id",
  "message":
  {
    "status": "SUCCESS",
    "traceback": "错误消息"
  }
}
```

如果指南id不存在或理解尚未完成则返回如下消息

```
{
  "task_id": "",
  "message": {
    "status": "NOT_FOUND",
    "traceback": "指南id"
  }
}
```

由于计算资源限制，对每个企业计算匹配度的任务会放在任务队列中进行调度，队列长度限制为100。也就是每次最多通过此接口传入100个企业id，等任务完成回调后再次传入下一批企业id。

如果传入企业id数量大于队列剩余长度，则会尽可能放入将企业id放入队列中，剩下未能放入的企业id将会返回到 `traceback` 字段中。例子如下

```
{
  "task_id": "任务id", # 如果所有企业都不能放进去则任务id为空
  "message": {
    "status": "FULL",
    "traceback": ["未能放入的企业id1", "未能放入的企业id2", "未能放入的企业id3"]
  }
}
```

## callback定义

`callback` 应为完整的url，如 `https://test.com/callback`

callback接口接受POST请求，请求数据为json格式

输入：

```
{
  "task_id": "任务id"
  "guide_id": "指南id",
  "result": {
    "企业id1": {"matching": 0.7, "status": "状态，如果计算失败则为FAIL否则为SUCCESS"},
    "企业id2": {"matching": 0.7, "status": "状态，如果计算失败则为FAIL否则为SUCCESS"},
    "...": {...}
  }
}
```

## callback测试

在系统接到 `/policy/check_recommend/` 的请求后，会对callback的url进行测试，测试时候会发送如下消息

```
{
  "guide_id": "指南id",
  "task_id": "test",
  "result": {
    "test": {"matching": "一个1到0之间的随机数", "status": "TEST"}
  }
}
```

callback应该返回该随机数，以确保callback接口能够正确读取结果。

测试不通过则不会将企业id加入队列中，并返回如下消息

```
{
  "task_id": "",
  "message": {
    "status": "CALLBACK_FAIL"
  }
}
```

## /policy/single\_recommend/

### GET

获取某个企业和某个政策的最近一次匹配情况。本接口只进行数据的查询，不会重新计算匹配情况。（在调用 `/policy/check_recommend/` 之后，得到callback响应之后，可以根据匹配度是否满足情况再通过本接口获取具体推荐原因）

输入：

key	value
guide_id	指南id
company_id	企业id

输出：

json格式

```
{
  "guide_id": "推荐事项的id",
  "reason": "推荐原因，可用于展示",
  "matching" : 0.90,
  "time": "推荐时间, fmt:YYYY-MM-DD HH:mm:ss"
}
```

