

八爪鱼增值 API 开发者文档

目录

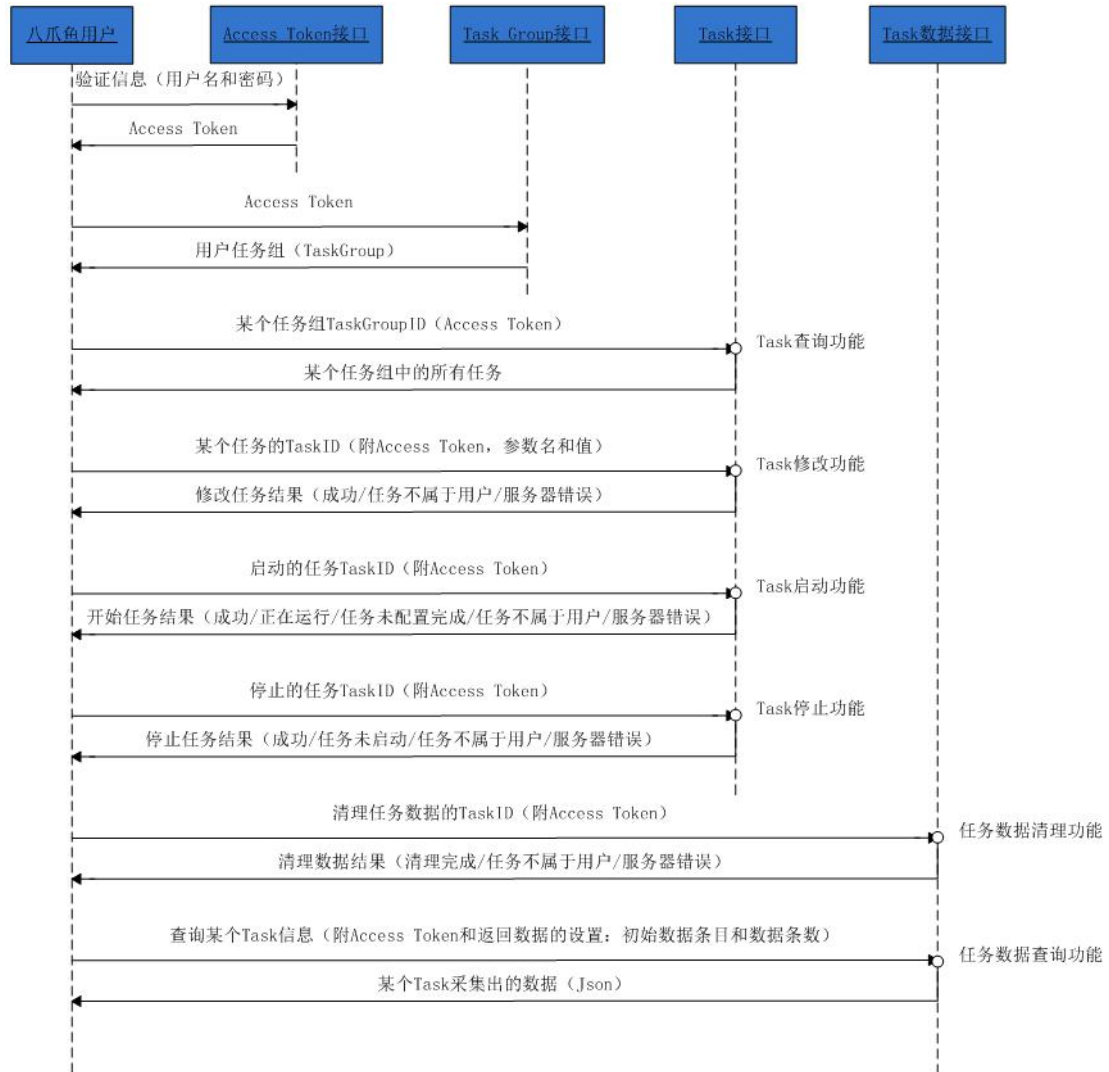
八爪鱼增值 API 开发者文档.....	1
1. 概述.....	1
2. 获取 Access Token.....	2
3. 任务数据相关接口.....	3
3.1. 分页获取任务所有数据.....	3
3.2. 分时分页获取任务所有数据.....	5
3.3. 分页获取任务最近一次启动时采集到的数据.....	6
3.4. 获取任务未导出的数据.....	7
3.5. 清理任务数据.....	9
4. Task 操作和修改任务相关接口.....	10
4.1. 启动任务.....	10
4.2. 停止任务.....	12
4.3. 查看任务运行状态.....	13
4.4. 修改任务采集流程中的配置参数.....	14
4.5. 获取任务采集流程中的配置参数原始值.....	17
5. 获取 TaskId 的两种方式.....	18
5.1. 通过 API 获取 TaskId.....	18
5.2. 通过八爪鱼客户端获取 TaskId.....	20
6. 示例代码.....	21

1. 概述

开发者需要按以下流程来使用八爪鱼数据 API 获取采集的数据。在此之前，您应该拥有一个八爪鱼数据平台的账号，并且建立了能够正常采集数据的任务。

API 使用的基本流程如下：

- 通过用户名和密码获取 Access Token
- 在采集服务器中使用 Access Token 和任务 ID（taskId）来获取任务采集的数据



2. 获取 Access Token

使用如下接口并提供用户名和密码来获取 Access Token。

```

http 请求方式: POST
http://advancedapi.bazhuayu.com/token
POST 表单数据格式: application/x-www-form-urlencoded
POST 表单参数示例:
username={username}&password={password}&grant_type=password
POST 表单数据中 username、password 的值需要进行 url 编码。
    
```

以上请求如果用户名和密码验证通过, Http Response 会返回类似如下的 Json 格式文本。

```

{
  "access_token": "ABCD1234",
  "token_type": "bearer",
  "expires_in": 86399,
  "refresh_token": "refresh_token"
}
    
```

返回结果中各个字段的说明如下:

字段	说明
access_token	访问标识 access token
token_type	access token 类型, 目前为 bearer
expires_in	access token 过期时间 (秒), 当前为 24 小时, 即 86400 秒
refresh_token	在 access token 过期后刷新 access token 的一个标记

Access Token 是以下所有 API 访问的访问许可标志, 在获取数据的 API 中务必加入到 Http 请求的头文件中。

```
Name: Authorization
Value: bearer {access token}
```

如果获取 Access Token 失败会返回若干种 Json 格式文本, 每种错误原因和 Json 格式解释如下。

1. POST 数据格式不正确, 要保证格式为:

```
username={username}&password={password}&grant_type=password
```

```
{
  "error": "unsupported_grant_type"
}
```

2. POST 数据中用户名和密码不匹配

```
{
  "error": "invalid_grant",
  "error_description": "The user name or password is incorrect."
}
```

3. 任务数据相关接口

3.1. 分页获取任务所有数据

使用此接口来分页获取任务的数据, 接口需要传入任务 ID (TaskId) 及分页信息 (pageIndex、pagesize), 同时还需要在 Header 中添加 AccessToken 信息。

```
http 请求方式: GET
http://advancedapi.bazhuayu.com/api/alldata?taskid={taskid}&pageIndex={
pageIndex}&pagesize={pageSize}
Http 头文件参数 1:
  Name: Authorization
  Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json
```

HTTP URL 参数示例:

```
http://advancedapi.bazhuayu.com/api/alldata?taskid=taskid&pageindex=1&pagesize=2
```

根据给定的 `pagesize`(`pagesize` 最大为 1000)对当前任务的数据进行分页, 然后在所有分页中返回 `pageindex` 这一页数据, 共返回 `pagesize` 条数据。例如某任务有 1000 条数据, 若传入 `pagesize=2`, `pageindex=1`, 则将数据分为 $1000/2=500$ 页, 每页 2 条数据, 然后返回第 1 页的 2 条数据。若以上请求中 Access Token 和 Taskid 均合法, 将会得到类似如下 Json 格式的任务数据:

```
{
  "data": {
    "total": 1000,
    "currentTotal": 2,
    "dataList": [
      {
        "省份": "安徽",
        "地域": "安庆",
        "日期": "2013-1-1",
        "最高气温": "0℃",
        "天气状况": "多云",
        "风力风向": "东北风微风",
        "最低气温": ""
      },
      {
        "省份": "安徽",
        "地域": "安庆",
        "日期": "2013-1-2",
        "最高气温": "5℃",
        "天气状况": "多云转阴",
        "风力风向": "东北风 3~4 级",
        "最低气温": "-2℃"
      }
    ]
  },
  "error": "success"
}
```

返回的数据有如下字段:

字段	解释
total	当前任务的数据记录总数
currentTotal	此次请求的数据记录数
dataList	数据记录集
error	提示信息, 若为 success 则表示成功

3.2. 分时分页获取任务所有数据

使用此接口来分页获取任务的数据,接口需要传入任务 ID(TaskId),采集起止时间(from、to)及分页信息(pageindex、pagesize),同时还需要在 Header 中添加 AccessToken 信息。

```
http 请求方式: GET
http://advancedapi.bazhuayu.com/api/alldata/getdataoftaskbytimeandpagin
g?taskid={taskid}&from={from}&to={to}&pageindex={pageindex}&pagesize={p
ageSize}
Http 头文件参数 1:
  Name: Authorization
  Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json
HTTP URL 参数示例
http://advancedapi.bazhuayu.com/api/alldata/getdataoftaskbytimeandpagin
g?taskid=taskid&from=2017-02-21 11:00:00&to=2017-02-21
12:00:00&pageindex=1&pagesize=2
```

根据采集起止时间 from (采集开始时间), to (采集结束时间), 开始时间和结束时间两者间隔不能超过 1 小时, 且必须为“yyyy-MM-dd HH:mm:ss”格式, 以及给定的 pagesize(pagesize 最大为 1000)对当前任务的数据进行分页, 然后返回在 from 到 to 这个时间段内采集到的所有数据中, 分页为 pageindex 这一页的数据, 共返回 pagesize 条数据。例如某任务有 1000 条数据, 若传入 pagesize=2, pageindex=1, 则将数据分为 1000/2=500 页, 每页 2 条数据, 然后返回第 1 页的 2 条数据。若以上请求中 Access Token 和 TaskId 均合法, 将会得到类似如下 Json 格式的任务数据:

```
{
  "data": {
    "total": 1000,
    "currentTotal": 2,
    "dataList": [
      {
        "省份": "安徽",
        "地域": "安庆",
        "日期": "2013-1-1",
        "最高气温": "0℃",
        "天气状况": "多云",
        "风力风向": "东北风微风",
        "最低气温": ""
      },
      {
        "省份": "安徽",
        "地域": "安庆",
        "日期": "2013-1-2",
```

```

        "最高气温": "5℃",
        "天气状况": "多云转阴",
        "风力风向": "东北风 3~4 级",
        "最低气温": "-2℃"
    }
]
},
"error": "success"
}
    
```

返回的数据有如下字段:

字段	解释
total	当前任务的数据记录总数
currentTotal	此次请求的数据记录数
dataList	数据记录集
error	提示信息, 若为 success 则表示成功

3.3. 分页获取任务最近一次启动时采集到的数据

使用此接口来分页获取任务的数据, 接口需要传入任务 ID (TaskId) 及分页信息 (pageIndex、pagesize), 同时还需要在 Header 中添加 AccessToken 信息。

```

http 请求方式: GET
http://advancedapi.bazhuayu.com/api/alldata/getlastdataoftaskbypaging?t
askid={taskId}&pageIndex={pageIndex}&pagesize={pageSize}
Http 头文件参数 1:
    Name: Authorization
    Value: bearer {access token}
Http 头文件参数 2:
    Name: Accept
    Value: application/json
HTTP URL 参数示例:
http://advancedapi.bazhuayu.com/api/alldata/getlastdataoftaskbypaging?t
askid =taskId&pageIndex=1&pagesize=2
    
```

根据给定的 pagesize(pagesize 最大为 1000)对当前任务的数据进行分页, 然后在所有分页中返回 pageIndex 这一页数据, 共返回 pagesize 条数据。例如某任务启动云采集 3 次, 分别采集了 500, 800, 1000 条数据。若传入 pagesize=2, pageIndex=1, 则将第 3 次采集到的数据分为 1000/2=500 页, 每页 2 条数据, 然后返回第 3 次采集中第 1 页的 2 条数据。若以上请求中 Access Token 和 TaskId 均合法, 将会得到类似如下 Json 格式的任务数据:

```

{
  "data": {
    "total": 1000,
    "currentTotal": 2,
    "dataList": [
    
```

```

    {
      "省份": "安徽",
      "地域": "安庆",
      "日期": "2013-1-1",
      "最高气温": "0℃",
      "天气状况": "多云",
      "风力风向": "东北风微风",
      "最低气温": ""
    },
    {
      "省份": "安徽",
      "地域": "安庆",
      "日期": "2013-1-2",
      "最高气温": "5℃",
      "天气状况": "多云转阴",
      "风力风向": "东北风 3~4 级",
      "最低气温": "-2℃"
    }
  ]
},
"error": "success"
}

```

返回的数据有如下字段:

字段	解释
total	当前任务的数据记录总数
currentTotal	此次请求的数据记录数
dataList	数据记录集
error	提示信息, 若为 success 则表示成功

3.4. 获取任务未导出的数据

使用此接口来分批获取任务未导出的数据, 接口需要传入任务 ID (TaskId) 及每批返回的数据条数 (size), 同时还需要在 Header 中添加 Access Token 信息, 接口将会返回先采集到的数据。

```

http 请求方式: GET
http://advancedapi.bazhuayu.com/api/notexportdata?taskid={testtaskid}&size={size}
Http 头文件参数 1:
  Name: Authorization
  Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json 或 application/xml

```

HTTP URL 参数示例:

```
http://advancedapi.bazhuayu.com/api/notexportdata?taskid=testtaskid&size=100
```

接口返回 size 条未导出的数据, 同时把返回的数据标识为已导出, 下次调用这个接口时将会跳过这部分已导出的数据。例如某任务有 1000 条数据, 若第一次调用传 size =2, 返回 2 条未导出的数据, 第二次调用的时候将会返回剩下 998 中的 2 条数据。若以上请求中 Access Token 和 TaskId 均合法, 将会得到类似如下 Json 格式的任务数据:

```
{
  "data": {
    "total": 1000,
    "currentTotal": 2,
    "dataList": [
      {
        "省份": "安徽",
        "地域": "安庆",
        "日期": "2013-1-1",
        "最高气温": "0℃",
        "天气状况": "多云",
        "风力风向": "东北风微风",
        "最低气温": ""
      },
      {
        "省份": "安徽",
        "地域": "安庆",
        "日期": "2013-1-2",
        "最高气温": "5℃",
        "天气状况": "多云转阴",
        "风力风向": "东北风 3~4 级",
        "最低气温": "-2℃"
      }
    ]
  },
  "error": "success"
}
```

返回的数据有如下字段:

字段	解释
total	当前任务的未导出的数据记录总个数
currentTotal	此次请求的数据记录个数
dataList	数据记录集
error	提示信息, 若为 success 则表示成功

若没有返回正确的数据, 那返回的数据有以下几种错误提示:

1. Access Token 无效，有可能过期，请使用正确的用户名和密码重新获取 Access Token

```
{
  "error": "unauthorized",
  "error_Description": "access_token 无效"
}
```

2. 采用了 POST 方法，请使用 GET 方法

```
{
  "message": "请求的资源不支持 http 方法“POST”。”
}
```

3. taskId 错误或该任务不属于 Access Token 对应的用户，请使用正确的 taskId

```
{
  "error": "taskid_error",
  "error_Description": "taskId 错误或该 task 不属于您"
}
```

4. size 太大，size 不允许超过 MaxSize 的值，此值目前系统设置为 1000

```
{
  "error": "export_pagesize_error",
  "error_Description": "size 限制在 1 到 1000"
}
```

5. 服务器暂时不可用

```
{
  "error": "server_error",
  "error_Description": "服务器错误，请稍后再试"
}
```

3.5. 清理任务数据

使用 RemoveData 接口并在 Header 中提供 Access Token 验证信息、任务 ID 来清理任务数据。

```
http 请求方式: POST
http://advancedapi.bazhuayu.com/api/task/RemoveDataByTaskId?taskId={taskId}
Http 头文件参数 1:
  Name: Authorization
  Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json
```

HTTP URL 参数示例:

```
http://advancedapi.bazhuayu.com/api/task/RemoveDataByTaskId?taskid=taskid
```

在以上请求中若 **access token** 合法，参数传递正确将会得到类似如下的任务组列表结构体文本:

```
{
  "error": "Success",
  "error_Description": "该任务的数据已经清理完毕"
}
```

若没有返回正确的数据，那返回的数据有以下几种错误提示:

1. **Access Token** 无效，有可能过期，请使用正确的用户名和密码重新获取 **Access Token**

```
{
  "error": "unauthorized",
  "error_Description": "access_token 无效"
}
```

2. 采用了 **GET** 方法，请使用 **POST** 方法

```
{
  "message": "请求的资源不支持 http 方法“GET”。"
}
```

3. **taskId** 错误或该任务不属于 **Access Token** 对应的用户，请使用正确的 **taskId**

```
{
  "error": "taskId_error",
  "error_Description": "taskId 错误或该 task 不属于您"
}
```

4. 服务器暂时不可用

```
{
  "error": "server_error",
  "error_Description": "服务器错误，请稍后再试"
}
```

4. Task 操作和修改任务相关接口

4.1. 启动任务

使用 **StartTask** 接口并在 **Header** 中提供 **Access Token** 验证信息和任务 ID 来启动任务。

```
http 请求方式: POST
http://advancedapi.bazhuayu.com/api/task/starttask?taskid=taskid
Http 头文件参数 1:
  Name: Authorization
```

```
Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json
HTTP URL 参数示例:
http://advancedapi.bazhuayu.com/api/task/starttask?taskid=taskid
```

在以上请求中若 **access token** 合法，参数传递正确将会得到类似如下的任务组列表结构体文本：

```
{
  "error": "Success",
  "error_Description": "任务启动成功"
}
```

若没有返回正确的数据，那返回的数据有以下几种错误提示：

1. **Access Token** 无效，有可能过期，请使用正确的用户名和密码重新获取 **Access Token**

```
{
  "error": "unauthorized",
  "error_Description": "access_token 无效"
}
```

2. 采用了 **GET** 方法，请使用 **POST** 方法

```
{
  "message": "请求的资源不支持 http 方法“GET”。"
}
```

3. **taskId** 错误或该任务不属于 **Access Token** 对应的用户，请使用正确的 **taskId**

```
{
  "error": "taskId_error",
  "error_Description": "taskId 错误或该 task 不属于您"
}
```

4. 任务正在执行中，无需再次启动

```
{
  "error": "task_on_executing",
  "error_Description": "任务正在执行中，不能再次启动"
}
```

5. 任务还未配置完成，不能启动

```
{
  "error": "task_not_ready",
  "error_Description": "任务还未配置完成，不能启动"
}
```

6. 服务器暂时不可用

```
{
```

```
{
  "error": "server_error",
  "error_Description": "服务器错误, 请稍后再试"
}
```

4.2. 停止任务

使用 StopTask 接口并在 Header 中提供 Access Token 验证信息和任务 ID 来停止任务。

```
http 请求方式: POST
http://advancedapi.bazhuayu.com/api/task/stoptask?taskid=taskid
Http 头文件参数 1:
  Name: Authorization
  Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json
HTTP URL 参数示例:
http://advancedapi.bazhuayu.com/api/task/stoptask?taskid=taskid
```

在以上请求中若 access token 合法, 参数传递正确将会得到类似如下的任务组列表结构体文本:

```
{
  "error": "Success",
  "error_Description": "任务停止成功"
}
```

若没有返回正确的数据, 那返回的数据有以下几种错误提示:

1. Access Token 无效, 有可能过期, 请使用正确的用户名和密码重新获取 Access Token

```
{
  "error": "unauthorized",
  "error_Description": "access_token 无效"
}
```

2. 采用了 GET 方法, 请使用 POST 方法

```
{
  "message": "请求的资源不支持 http 方法 “GET” 。"
}
```

3. taskId 错误或该任务不属于 Access Token 对应的用户, 请使用正确的 taskId

```
{
  "error": "taskId_error",
  "error_Description": "taskId 错误或该 task 不属于您"
}
```

4. 服务器暂时不可用

```
{
```

```

    "error": "server_error",
    "error_Description": "服务器错误, 请稍后再试"
  }

```

4.3. 查看任务运行状态

使用 `GetTaskStatusByIdList` 接口并在 Header 中提供 Access Token 验证信息、Id 列表(Json 格式)来获取批量任务的运行状态。

```

http 请求方式: POST
http://advancedapi.bazhuayu.com/api/task/GetTaskStatusByIdList
Http 头文件参数 1:
  Name: Authorization
  Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json
POST 数据参数示例:
{"taskIdList":["taskid1","taskid2"]}

```

在以上请求中若 `access token` 合法, 参数传递正确将会得到类似如下的任务组列表结构体文本:

```

{
  "data": [
    {
      "taskId": "taskid1",
      "taskName": "任务 1",
      "Status":2
    },
    {
      "taskId": "taskid2",
      "taskName": "任务 2",
      "Status":2
    }
  ]
  "error":"success"
}

```

任务组列表结构体各个字段解释如下:

字段	解释
taskId	任务唯一标示符
taskName	任务名称
Status	任务的运行状态 0:正在运行, 1: 已停止, 2: 已完成, 3: 等待运行, 5: 未启动

若没有返回正确的数据, 那返回的数据有以下几种错误提示:

1. Access Token 无效, 有可能过期, 请使用正确的用户名和密码重新获取 Access Token

```

{

```

```
"error": "unauthorized",
"error_Description": "access_toekn 无效"
}
```

2. 采用了 GET 方法，请使用 POST 方法

```
{
  "message": "The requested resource does not support http method 'GET'."
}
```

3. 参数为空或参数格式错误

```
{
  "error": "param_error",
  "error_Description": "参数为空或参数格式错误"
}
```

4. 服务器暂时不可用

```
{
  "error": "server_error",
  "error_Description": "服务器错误，请稍后再试"
}
```

4.4. 修改任务采集流程中的配置参数

使用 `UpdateTaskRule` 接口并在 Header 中提供 Access Token 验证信息、任务 ID 和要修改的配置参数名称、需要修改成的值。

```
http 请求方式: POST
http://advancedapi.bazhuayu.com/api/task/UpdateTaskRule
Http 头文件参数 1:
  Name: Authorization
  Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json
POST 数据参数:
{
  taskId: {taskId},
  name: {parameterName},
  value: {parameterValue}
}
HTTP URL 参数示例: http://advancedapi.bazhuayu.com/api/task/UpdateTaskRule
POST 数据参数示例:
1. 修改循环步骤 [loopAction1] 的 URL 列表
{
  "taskId": "taskid1",
  "name": "loopAction1.UrlList",
```

```
    "value":["www.abc.com","www.def.com"]
  }
2. 修改打开网页步骤[navigateAction1]的Url
{
  "taskId":"taskid1",
  "name":"navigateAction1.Url",
  "value":"www.abc.com "
}
```

在以上请求中若 access token 合法，参数传递正确将会得到类似如下的任务组列表结构体文本：

```
{
  "error": "Success",
  "error_Description": "该任务的参数已经被修改"
}
```

若没有返回正确的数据，那返回的数据有以下几种错误提示：

1. Access Token 无效，有可能过期，请使用正确的用户名和密码重新获取 Access Token

```
{
  "error":"unauthorized",
  "error_Description":"access_toekn 无效"
}
```

2. 采用了 GET 方法，请使用 POST 方法

```
{
  "message":"The requested resource does not support http method 'GET'."
}
```

3. taskId 错误或该任务不属于 Access Token 对应的用户，请使用正确的 taskId

```
{
  "error": "taskId_error",
  "error_Description": "taskId 错误或该 task 不属于您"
}
```

4. 参数为空或参数格式错误

```
{
  "error":"param_error",
  "error_Description":"参数为空或参数格式错误"
}
```

5. 没有找到对应的配置参数

```
{
  "error":"task_notfind_configuration_parameter",
  "error_Description":"该任务没有找到对应的配置参数"
}
```

```
}
}
```

6. 服务器暂时不可用

```
{
  "error": "server_error",
  "error_Description": "服务器错误, 请稍后再试"
}
```

注:

一、配置参数名称解释如下:

流程中的每个步骤都会有个唯一标识, 同时部分配置设置是可以通过 API 来修改的, **配置参数名称=步骤唯一标识.设置名称**, 如打开网页的 Url 设置:navigateAction1.Url, 可以通过 API 修改的配置有

流程步骤	设置名称	配置参数名称示例	说明
打开网页 (Navigate)	Url	navigateAction1.Url	设置打开网页的【页面 Url】
循环 (Loop)	UrlList	loopAction1.UrlList	设置【Url 列表】, 每个连接用换行符分配 (\r\n)
	TextList	loopAction1.TextList	设置【文本列表】, 每个文本用换行符分配 (\r\n)
输入文本 (EnterText)	Text	enterTextAction1.Text	设置【要输入的文本】

二、获取步骤的唯一标识需要从客房端中获取, 先要打开的高级模式的编辑页面, 并打开 workflow 设置界面, 把鼠标放放到流程中的频上, 会有个弹出气泡, 我们可以从气泡中获取频的唯一标识, 如下图:

1. 打开网页的唯一标识



2. 输入文本的唯一标识



3. 循环步骤的唯一标识



4.5. 获取任务采集流程中的配置参数原始值

使用 `GetTaskRulePropertyByName` 接口并在 `Header` 中提供 `Access Token` 验证信息、任务 ID 和要读取的配置参数名称。

```

http 请求方式: POST
http://advancedapi.bazhuayu.com/api/task/GetTaskRulePropertyByName?task
id=taskid&name=name
Http 头文件参数 1:
  Name: Authorization
  Value: bearer {access token}
Http 头文件参数 2:
  Name: Accept
  Value: application/json
HTTP URL 参数示例:
http://advancedapi.bazhuayu.com/api/task/GetTaskRulePropertyByName?task
id=taskid&name=name
  
```

在以上请求中若 `access token` 合法，参数传递正确将会得到类似如下的任务组列表结构体文本：

```

{
  "data": ["www.baidu.com"],
  "error": "success"
}
  
```

```
}
```

若没有返回正确的数据，那返回的数据有以下几种错误提示：

1. Access Token 无效，有可能过期，请使用正确的用户名和密码重新获取 Access Token

```
{
  "error": "unauthorized",
  "error_Description": "access_toekn 无效"
}
```

2. 采用了 GET 方法，请使用 POST 方法

```
{
  "message": "The requested resource does not support http method 'GET'."
}
```

3. taskId 错误或该任务不属于 Access Token 对应的用户，请使用正确的 taskId

```
{
  "error": "taskid_error",
  "error_Description": "taskId 错误或该 task 不属于您"
}
```

4. 参数为空或参数格式错误

```
{
  "error": "param_error",
  "error_Description": "参数为空或参数格式错误"
}
```

5. 没有找到对应的配置参数

```
{
  "error": "task_notfind_configuration_parameter",
  "error_Description": "该任务没有找到对应的配置参数"
}
```

6. 服务器暂时不可用

```
{
  "error": "server_error",
  "error_Description": "服务器错误，请稍后再试"
}
```

5. 获取 TaskId 的两种方式

5.1. 通过 API 获取 TaskId

1. 首先获取用户的任务组，使用如下接口并在 Header 中提供 Access Token 验证信息来获取任务组列表。

```
http 请求方式：GET
```

```
http://dataapi.bazhuayu.com/api/taskgroup
```

Http 头文件参数:

Name: Authorization

Value: bearer {access token}

在以上请求中若 **Access Token** 合法且成功查询到数据将会得到类似如下的任务组列表结构体文本:

```
{
  "data": [
    {
      "taskGroupId": 84,
      "taskGroupName": "任务组 1"
    },
    {
      "taskGroupId": 527,
      "taskGroupName": "任务组 1"
    }
  ]
  "error": "success"
}
```

任务组列表结构体各个字段解释如下:

字段	解释
taskGroupId	任务组唯一标示符
taskGroupName	任务组名称

2. 对于一个任务组, 可以通过提供任务组 ID 来获取该任务组下的所有任务列表。

使用如下接口并在 **Header** 中提供 **Access Token** 验证信息, 同时传入任务组 ID 作为参数来获取任务列表。

http 请求方式: GET

```
http://advancedapi.bazhuayu.com/api/task?taskgroupid={taskgroupid}
```

Http 头文件参数:

Name: Authorization

Value: bearer {access token}

HTTP URL 参数示例:

```
http://advancedapi.bazhuayu.com/api/task?taskgroupid=?taskgroupid=84
```

以上请求中若 **Access Token** 合法并且 **taskgroupid** 的值是用户拥有的某个任务组 ID, 即是第 3 节中查询到的多个 ID 之一, 将会得到类似如下任务列表结构体文本。

```
{
  "data": [
    {
      "taskId": "taskid1",
```

```

        "taskName": "任务 1"
    },
    {
        "taskId": "taskid2",
        "taskName": "任务 2"
    }
]
"error": "success"
}
    
```

任务列表结构体各个字段解释如下：

字段	解释
taskId	采集任务唯一标示符
taskName	采集任务名称

可以通过某个任务 ID 来获取该任务所采集的数据。

一般的使用场景：对于某一种采集任务，若用户因为采集量太大而将其分成了一个任务组中的多个任务来采集，此时通过本节两个 API 来获得任务组 ID 和该组下的所有任务 ID，然后编写程序使用数据 API 分别获取这些任务 ID 对应的任务采集来的数据，再进行合并。

若没有返回正确的数据，那返回的数据有以下几种错误提示

1. Access Token 无效，有可能过期，请使用正确的用户名和密码重新获取 Access Token

```

{
    "error": "unauthorized",
    "error_Description": "access_token 无效"
}
    
```

2. 采用了 POST 方法，请使用 GET 方法

```

{
    "message": "请求的资源不支持 http 方法“POST”。”
}
    
```

3. 服务器暂时不可用

```

{
    "error": "server_error",
    "error_Description": "服务器错误，请稍后再试"
}
    
```

5.2. 通过八爪鱼客户端获取 TaskId

（仅旗舰版和私有云用户可使用此功能）

登录八爪鱼客户端之后在“我的任务”中展开一个任务组，右击其中一个任务并单击“数据导出 API”，在对话框中可得到 TaskId。



6. 示例代码

完整可运行的示例代码见 GitHub: <https://github.com/octopus-dev/AdvancedApi>