八爪鱼数据导出 API 通用教程

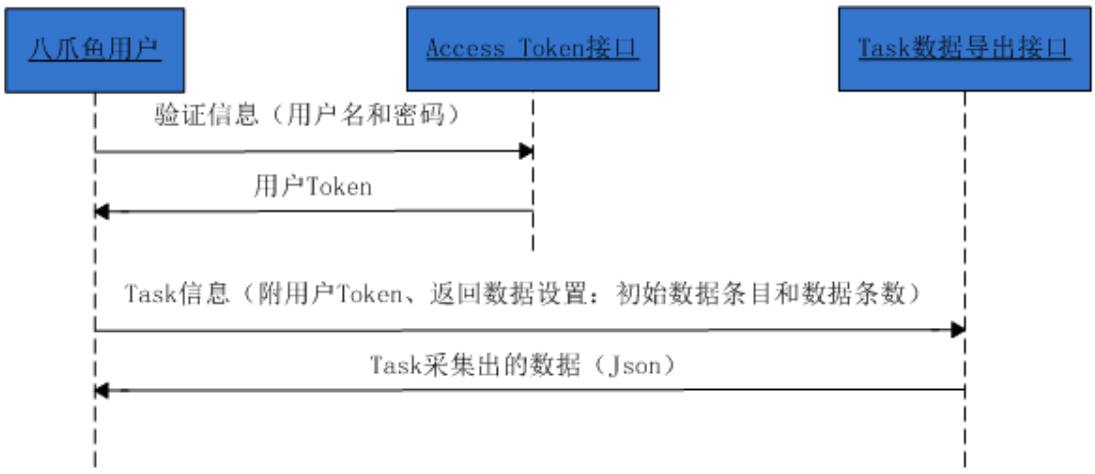
目录

1. 概述

开发者需要执行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。通过一系列步骤向八爪鱼数据 API 提供合法访问标志(access token)和采集任务的信息（taskId）即可得到任务所采集的所有或者部分数据。

- 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

基本流程如下：



下文介绍这些流程并给出代码示例。

2. 使用 token API 获取访问数据 API 所需的 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://dataapi.bazhuayu.com/token
POST 表单数据格式：application/x-www-form-urlencoded
POST 表单参数示例：username=test&password=test&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": "46b94a3ead204d1b84e101364a71c6d1"
}
```

access token 结构体各个字段的解释如下：

字段	解释
access_token	访问标识 access token
token_type	access token 类型
expires_in	access token 过期时间（秒），当前为 24 小时
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=test&password=test&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."
}
```

注：参考示例代码 5.1。

3. 使用 Data API 来获取任务数据

使用如下接口并在 header 中提供 access token 验证信息、任务 ID（taskid，下节将会讲解两种获取 taskid 的方法）和页面请求信息来获取数据。

目前提供了两个接口，AllData 和 NotExportData，前者可以获得所有任务的数据，后者可以获得未导出的任务的数据。

```
http 请求方式: GET
http://dataapi.bazhuayu.com/api/alldata?taskid=testtaskid&pageindex=1&pagesize=1
或者
http://dataapi.bazhuayu.com/api/notexportdata?taskid=testtaskid&pageindex=1&pagesize=1
Http 头文件参数 1:
    Name: Authorization
    Value: bear [access token]
Http 头文件参数 2:
    Name: Accept
    Value: application/json
```

Http 头文件数据例子：

Name	Value	说明
Authorization	bearer ABCD1234	Token 验证信息
Accept	application/json	表明接受返回数据的格式，默认采用 json

```
HTTP URL 参数示例: ?taskid=0e309d43-c4d6-4055-9461-a701a6e41894&pageindex=1&pagesize=2
```

其中 pageindex 指明从所有数据中的那一条开始获取，pagesize 指明获取多少条数据。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": 1316176,
        "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

注：一、参考示例代码 5.4。

二、获取任务数据时若提供的参数不合法，会返回如下错误结果，每种错误的原因和 json 格式解释如下。

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 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 task 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. pageSize 太大，pageSize 不允许超过 MaxPageSize 的值，此值目前系统设置为 1000


```
{
    "error": "export_pagesize_error",
    "error_Description": "pageSize 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！"
}
```

4. 获取 Task ID 的两种方式

4.1. 使用 Task Group API 和 Task API 来获取 Task ID

1. 首先获取用户的 task group，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 task group，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 验证信息和任务组 ID 来获取任务列表。

```
http 请求方式: GET
http://dataapi.bazhuayu.com/api/task?taskgroupid=1234
Http 头文件参数:
    Name: Authorization
    Value: bearer [access token]
HTTP URL 参数示例：?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": "0e309d43-c4d6-4055-9461-a701a6e41894",
            "taskName": "任务 1"
        },
        {
            "taskId": "1d51212c-a837-442c-8614-1be29b8a02b3",
            "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所有或者部分数据。

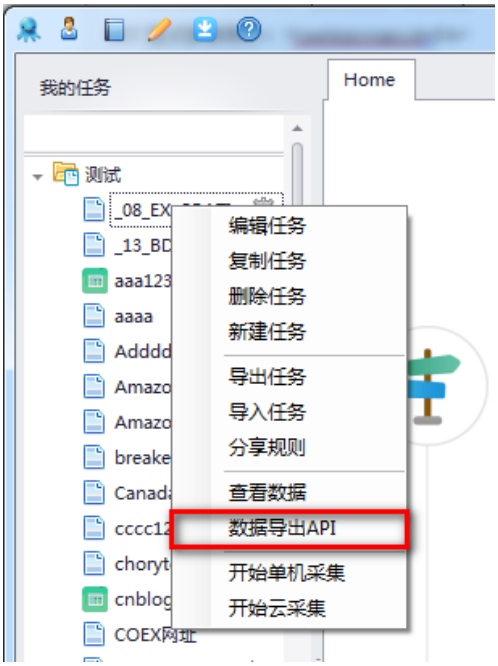
一般的使用场景：对于某一种采集任务，若用户因为采集量太大而将其分成了一个任务组中的多个任务来采集，此时通过本节两个 API 来获得任务组 ID 和该组下的所有任务 ID，然后编写程序使用数据 API 分别获取这些任务 ID 对应的任务采集来的数据，再进行合并，

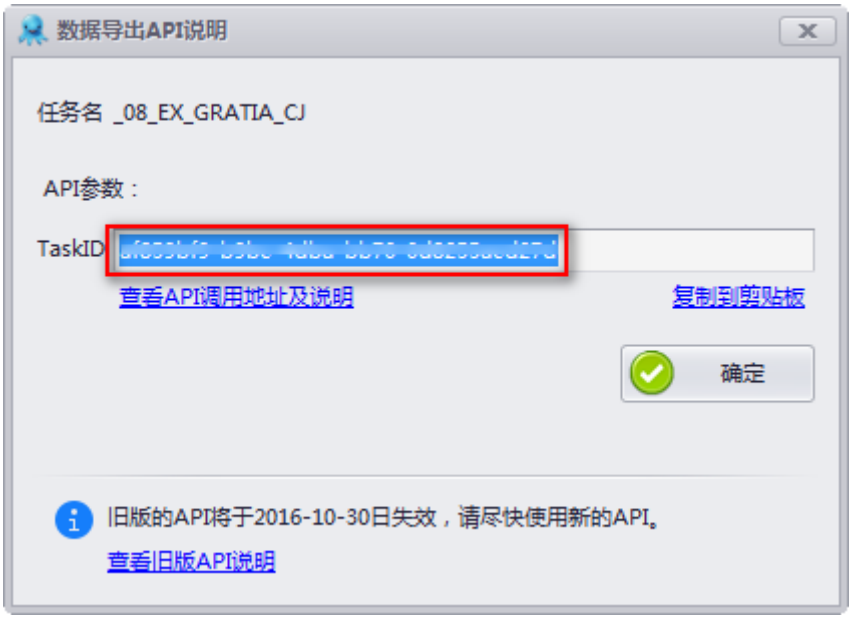
注：参考示例代码 5.1&5.2。

4.2. 使用八爪鱼客户端来获取 task ID

（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。





5. 示例代码

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

5.1. 获取用户的 Access Token

```
        /// <summary>
        /// 用HTTP的POST方法提交username和password，通过tokenUrl接口来获取token
        /// </summary>
        /// <param name="userName">八爪鱼用户名</param>
        /// <param name="password">登录密码</param>
        /// <param name="tokenUrl">token接口地址，一般为
http://ipaddress/token</param>
        /// <returns></returns>
        public string GetToken(string userName, string password, string
tokenUrl)
        {
            if (null != userName && null != password && null != tokenUrl)
            {
                string postdata =
string.Format("username={0}&password={1}&grant_type=password", userName,
password);
                string responseText = HttpHelper.Post(tokenUrl, postdata);
                if (responseText.Contains("access_token"))
                {
                    token =
JObject.Parse(responseText)["access_token"].ToString();
                }
            }
            return token;
```

```
        }
```

5.2. 获取用户任务组

```
        /// <summary>
        /// 通过taskgroup接口来获得某个用户的所有taskgroup
        /// </summary>
        /// <param name="token">某个用户的access token</param>
        /// <param name="taskGroupUrl">taskgroup接口地址，一般为
http://ipadress/api/taskgroup</param>
        public void GetTaskGroups(string token, string taskGroupUrl)
        {
            if (null != token && null != taskGroupUrl)
            {
                Dictionary<string, string> headers = new Dictionary<string,
string>(1);
                headers.Add("Authorization", string.Format("bearer {0}",
user.token));
                string TaskGroupsStr = HttpHelper.GetWithHeaders(taskGroupUrl,
headers);
                if (!TaskGroupsStr.Contains("\"data\":"))
                {
                    //DelegGroupProgressTextChange(string.Format("获取任务组失败!
"));
                    return;
                }
                JObject jsonTaskGroupsAll = JObject.Parse(TaskGroupsStr);
                JArray jsonTaskGroupsJarray = jsonTaskGroupsAll["data"] as
JArray;
                user.taskGroups = new
List<TaskGroup>(jsonTaskGroupsJarray.Count);
                foreach (JToken jtokenTaskGroup in jsonTaskGroupsJarray)
                {
                    user.taskGroups.Add(new TaskGroup()
                    {
                        taskGroupID = jtokenTaskGroup["taskGroupId"].ToString(),
                        taskGroupName =
jtokenTaskGroup["taskGroupName"].ToString(),
                        tasks =
GetTasks(token,jtokenTaskGroup["taskGroupId"].ToString(),taskUrl)
                    });
                }
            }
        }
```

5.3. 获取用户任务组的所有任务

```
        /// <summary>
        /// 通过task接口来获得某个任务组的所有tasks，请求格式
http://ipadress/api/task?taskgroupid=123
        /// </summary>
        /// <param name="token">access token</param>
        /// <param name="taskGroupID">任务组标识</param>
        /// <param name="taskUrl">task接口地址，一般为
http://ipadress/api/task</param>
        /// <returns>传入任务组ID的任务组下的所有任务</returns>
        public List<Task> GetTasks(string token,string taskGroupID, string
taskUrl)
        {
            List<Task> tasks = null;
            if (null != user && null != taskUrl)
            {
                Dictionary<string, string> headers = new Dictionary<string,
string>(1);
                headers.Add("Authorization", string.Format("bearer {0}",
token));
                string URL = string.Format("{0}?taskgroupid={1}", taskUrl,
taskGroupID);
                string taskStr = HttpHelper.GetWithHeaders(URL, headers);
                if (taskStr.Contains("\"data\":"))
                {
                    JObject jsonTasks = JObject.Parse(taskStr);
                    JArray jsonTasksJarray = jsonTasks["data"] as JArray;
                    tasks = new List<Task>(jsonTasksJarray.Count);
                    foreach (JToken jtokenTask in jsonTasksJarray)
                    {
                        tasks.Add(new Task()
                        {
                            taskID = jtokenTask["taskId"].ToString(),
                            taskName = jtokenTask["taskName"].ToString()
                        });
                    }
                }
                else
                {
                    //elegAddProgressInfoIntoListView(string.Format("获取Tasks失
败！返回文本：{1}", taskStr));
                }
            }
            return tasks;
```

```
        }
```

5.4. 获取某个任务的采集数据

```
        /// <summary>
        /// 根据TaskID获取该任务采集的数据
        /// </summary>
        /// <param name="token">用户Token</param>
        /// <param name="dataUrl">数据接口地址</param>
        /// <param name="taskID">任务ID</param>
        /// <param name="pageIndex">初始数据条目位置</param>
        /// <param name="pageSize">数据条目个数</param>
        /// <returns>任务采集的数据</returns>
        public string GetDataByTask(string token,string dataUrl,string
taskID,int pageIndex,int pageSize)
        {
            string taskData = "";
            if (null != token && null != dataUrl)
            {
                string URL = "";
                Dictionary<string, string> headers = new Dictionary<string,
string>(1);
                headers.Add("Authorization", string.Format("bearer {0}",
user.token));
                URL = string.Format("{0}?taskid={1}&pageindex={2}&pagesize={3}",
dataUrl, taskID, pageIndex, pageSize);
                taskData = HttpHelper.GetWithHeaders(URL, headers);
            }
            return taskData;
        }
```