Octoparse Data Export API

Content

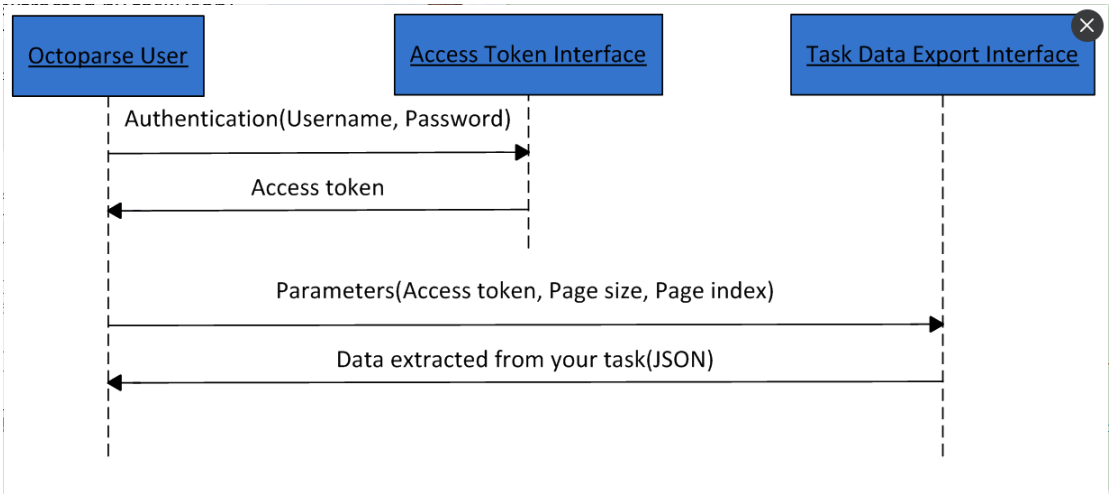
You must obtain an access token to use the Octoparse API. The access token is passed with each API request and is used to authenticate you access to the Octoparse API. It provides a secure access to the Octoparse API.

1. Overview

You can export data extracted using the Octoparse Data API by using the following procedure. It is worth mentioning that you have an Octoparse advanced account(Standard/Professional) and have obtained some data from at least one task that is running in the cloud before using the Octoparse API.

The basic flow to use the Octoparse API:

1. Get an access token by providing your user name and password.
2. Use Access Token and Task ID to get the data from a specific extraction task in Octoparse.



2. Get an Access Token

You can obtain an access token by making an HTTP POST request with your username and password.

```
HTTP Method: POST
http://dataapi.octoparse.com/token
POST Content Type: application/x-www-form-urlencoded
POST Example:
username={username}&password={password}&grant_type=password
The values of username and password should be URL-Encoded.
```

The successful HTTP response for the token request contains the access token that you can use to access the Octoparse API. The response is JSON-encoded and below is an example response.

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

The response includes the output properties as follows.

Property	Description
access_token	The access token that you can use to authenticate you access to the Octoparse API.
token_type	The format of the access toke. Currently, Octoparse uses a BEARER token.

expires_in	The number of seconds for which the access token is valid.Current default value is 86400(24 hours).
refresh_token	A token that can be sent to Octoparse API instead of an authorization code. (When the access token expires, send a POST request to the Octoparse API using this token instead of an authorization code. A new access token will be returned. A new refresh token might be returned too.)

An access token is a unique identifier of making an Octoparse API call, and is needed to add to the HTTP request Header to get the data from tasks via API.

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

Note:
The response would return some JSON-formatted strings if the request for access token failed. Below are some explanations for all error cases.

Case 1. The content of the POST is not formatted correctly.

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

Make sure that the format is like:

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

Case 2. The user name or password in the POST is incorrect.

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

3. How to Get Data through API

3.1. Get all data of a task using paging

Octoparse supports paging of data to retrieve only some data records by displaying a particular page of data, using the HTTP GET request. The parameters - taskID, pageindex, pagesize are needed for this API and the access token should be added to the HTTP Header.

```
HTTP Method: GET
http://dataapi.octoparse.com/api/alldata?taskid={taskid}&pageindex={pageIndex}&pagesize={pageSize}
HTTP Header parameter 1:
   Name: Authorization
   Value: bearer {access token}
HTTP Header parameter2:
   Name: Accept
   Value: application/json
HTTP URL sample:
http://dataapi.octoparse.com/api/alldata?taskid=taskid&pageindex=1&pagesize=2
```

Octoparse will page through the data of the current task based on the given pagesize(The maximum allowed page size is 1,000) and return the data of the index page; the number of data records returned is based on the page size you set.

For example, let's say there are 1,000 data records in a task. If the pageindex is 1 and pagesize is 2 (pagesize=2, pageindex=1), the data will be divided into 500 pages with 2 data records per page and Octoparse API would return the first page of 2 data records.

The successful HTTP response with correct access token and taskID will get JSON -formatted data. Below is an example response.


```
{
"data": {
"total": 1000,
"currentTotal": 2,
"dataList": [
          {
"State": "Texas",
"City": "Plano",
"Date": "2013-1-1",
"Humidity": "34%",
"High Temperatures": "72.8F",
"Wind": "NW 8km/h",
"Low Temperatures": "24.8F"
          },
          {
"State": "Texas",
"City": "Plano",
"Date": "2013-1-2",
"Humidity": "32%",
```

```
"High Temperatures": "76F",
"Wind": "NNW 10km/h",
"Low Temperatures": "25F"
          }
       ]
   },
"error": "success"
}
```

The data returned includes fields as follows.

Data field	Description
total	The number of total data records of the current task
currentTotal	The number of data records requested
dataList	The list of data fields
error	Prompt information

3.2. Get Unexported Data from a Task

You can get all unexported data from a task in batches, using the HTTP GET request. The taskID and the number of data records returned per batch(size) are needed for the request, and the access token is needed to add to the HTTP Header. Octoparse API will return the data that were first collected.

```
HTTP Method: GET
http://dataapi.octoparse.com/api/notexportdata?taskid={testtaskid}&size
={size}
HTTP Header parameter 1:
   Name: Authorization
   Value: bearer {access token}
HTTP Header parameter 2:
   Name: Accept
   Value: application/json
          application/xml
HTTP URL sample:
http://dataapi.octoparse.com/api/notexportdata?taskid=testtaskid&size=1
00
```

The interface would return the unexported data (the amount of unexported data depends on the parameter: ‘size’) and then identify this data as exported data so that all exported data will be skipped next time you make a request.

For example, let's say there are 1,000 data records in a task. If the size is 2 (the number of data records returned per batch) for the first request, Octoparse API will return 2 data records that were first collected. Similarly, next time it will return another 2 data records that were first collected from the remaining 998 records.

The successful HTTP response with correct access token and taskID will get JSON -formatted data. Below is an example response.


```
{
"data": {
"total": 1000,
"currentTotal": 2,
"dataList": [
          {
"State": "Texas",
"City": "Plano",
"Date": "2013-1-1",
"Humidity": "34%",
"High Temperatures": "72.8F",
"Wind": "NW 8km/h",
"Low Temperatures": "24.8F"
          },
          {
"State": "Texas",
"City": "Plano",
"Date": "2013-1-2",
"Humidity": "32%",
"High Temperatures": "76F",
"Wind": "NNW 10km/h",
"Low Temperatures": "25F"

          }
       ]
   },
"error": "success"
}
```


The data returned includes fields as follows.

Data field	Description
total	The number of total unexported data records of the current task
currentTotal	The number of data records requested

dataList	The list of data fields
error	Prompt information

Note:
If the parameter provided is incorrect when getting data from task, Octoparse API will return the following errors. Below are some explanations for all error cases.

Case 1. Access token is invalid or has expired. Please use your username and password to obtain a new access token.

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

Case 2. The requested resource does not support HTTP method 'POST'. Please use GET method in this case.

```
{
"message": "Requested Resource Does Not Support HTTP Method 'POST'"
}
```

Case 3. The taskID is invalid or the task doesn't belong to the user indicated by the access token. Please use correct taskID.

```
{
"error": "taskid_error",
"error_Description": "TaskID is invalid or the task does not belong to
you."
}
```

Case 4. The size is too big and exceeds the maximum allowed size. The default size is 1000.

```
{
"error": "export_pagesize_error",
"error_Description": "Size range from 1 to 1000"
}
```

Case 5. The server is temporarily unavailable.

```
{
"error":"server_error",
"error_Description": "Server Error. Please try again later!"
}
```

4. Two Ways to Get a Task ID

4.1. Get a Task ID via API

You can get all data from a task via a task ID.

Generally, users will create a task group to extract large amounts of data and therefore will create many tasks that categorized into that group to extract this data separately. In this case you can obtain the task group ID and all the task IDs under the group via two APIs (One for task group ID, the other for task ID), then extract all the data from these tasks in the group by writing codes to work with the APIs.

4.1.1. Get a Task Group ID

First of all, you need to obtain task group ID by using the HTTP GET request and adding the access token to the HTTP Header.

```
HTTP Method: GET
http://dataapi.octoparse.com/api/taskgroup
HTTP Header parameter:
   Name: Authorization
   Value: bearer {access token}
```

If the access token you requested is accurate and could be used to get data, you will get a text-formatted task group list as follows.

```
{
"data": [
       {
"taskGroupId": 84,
"taskGroupName": "Task Group ID 1"
       },
       {
"taskGroupId": 527,
"taskGroupName": "Task Group ID 2"
       }]
   "error": "success"
}
```

The descriptions of data fields in the task group list are as follows.

Data Field	Description
taskGroupId	The unique identifier for the task group
taskGroupName	The name for the task group

4.1.2. Get a Task ID from the task group

For a task group, all the tasks under the task group can be obtained by providing the task group ID.

You can get the list of all the tasks by using the HTTP GET request, adding the access token to the HTTP Header and using the task group ID as the parameter.

```
HTTP Method: GET
http://dataapi.octoparse.com/api/task?taskgroupid={taskgroupid}
HTTP Header parameter:
   Name: Authorization
   Value: bearer {access token}
HTTP URLsample:
http://dataapi.octoparse.com/api/task?taskgroupid=?taskgroupid=84
```

If the access token you requested is accurate and the task group belongs to you, you will get a text-formatted task list as follows.

```
{
"data": [
       {
"taskId": "taskid1",
"taskName": ""
       },
       {
"taskId": "taskid2",
"taskName": "Task 2"
       }]
   "error": "success"
}
```

The descriptions of data fields in the task list are as follows.

Data field	Description
taskId	The unique identifier for the task
taskName	The name for the task

Note:
If the parameter provided is incorrect when getting data from task, Octoparse API will return the following errors. Below are some explanations for all error cases.

Case 1. Access token is invalid or has expired. Please use your username and password to obtain a new access token.

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

Case 2. The requested resource does not support HTTP method 'POST'. Please use GET method in this case.

```
{
"message": "Requested Resource Does Not Support HTTP Method 'POST'"
}
```

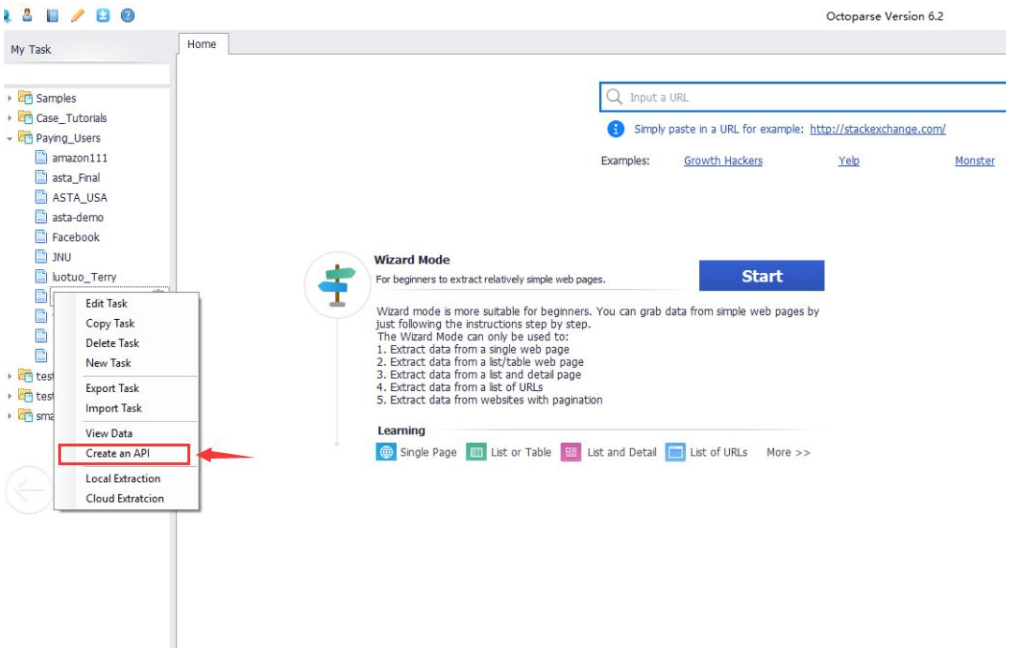
Case 3. The server is temporarily unavailable.

```
{
"error": "server_error",
"error_Description": "Server Error. Please try again later!"
}
```

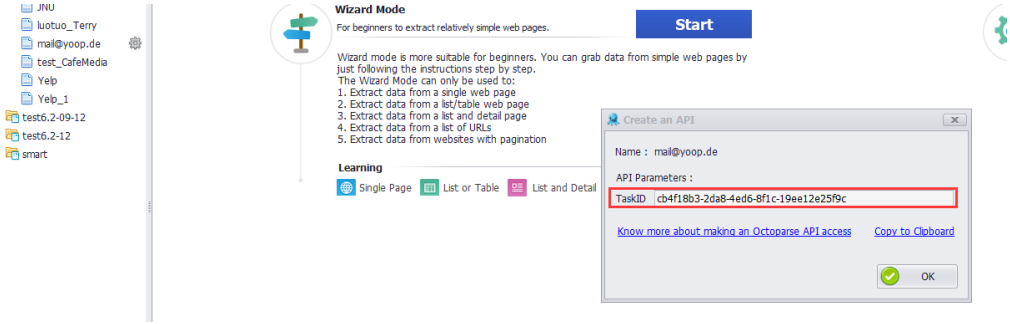
4.2. Get a task ID via Octoparse client

This function is only available for Standard and Professional Plans.

After you login to Octoparse, right click a task and choose “Create an API”.(Only available for Standard and Professional Plan).



Then you will get the Task ID on the pop-up window.



5. Sample Code

GitHub：https://github.com/octopus-dev/DataExportApi