

Vettd Data Observatory (VDO) Application Program Interface(API)

April 5th, 2018



STEP 1	STEP 2	STEP 3	STEP 4	STEP 5	STEP 6
					
Authenticate	List Workflows	Upload Files	Call Workflow	Retrieve Job Results	Download Results
Get an authorization token for all subsequent calls	Choose which workflow to use. This will determine what input files are required to perform the job.	Upload all the files that will be used in the workflow.	Call a workflow and tell it which files/models to use as input to process. In doing so you are creating a new job.	Query your job to see the status of the job. If complete it will provide you with a reference to the output file to download	The output results will download as 1 file. If the results contain more than 1 file it will be returned as a zip file.
	*NOTE: If you already know which workflow to call you can skip this step	*NOTE: If you already have the files uploaded you can skip this step			

Overview

The VettD Data Observatory (VDO) Application Program Interface(API) is a RESTful API. Authentication is done by the OAuth 2.0 protocol.

This document will walk you through a simple scenario to get a client up and running with the VDO API with the intention of getting your feet wet with the power of the API.

The online API documentation can be located in the table below. This documentation will describe what actions can be taken, but the purpose of this document is to describe the order to take the actions. Take a few moments to get an overview of the **Testing** documentation below before continuing.

Environment	URL
Testing	https://get.vettd.com/vdo/test/docs/
Production	https://get.vettd.com/vdo/v1/docs/ <i>*Note v1 is subject to increase when new version are released</i>

This document will make all calls using the popular cURL syntax over HTTPS – but you can use whatever language/method you prefer that supports RESTful calls. The response is defaulted to JSON but may be returned as XML as an alternative.

Onboarding

You as a client will need Six values issued to you by Vettd to begin working with the API. Those values are below with different values to access our **TEST** and **PRODUCTION** environments:

Value	Description
Client Id	Your unique Id
Client Secret	Your unique secret/password
API Key	A value passed into every call (except authentication) as second factor of authentication

NOTE: The client secret and API Key can be reset on demand when needed by contacting Vettd.

The above information should be treated like you would treat a username and password.

We can also lock down the access even further if you provide a while list of valid IPs that will be allowed to access the VDO API. This list can be as a range and/or single IPs.

High Level API Concepts

Value	Definition
Asset	A physical file.
Dataset	A reference to 1 or more assets.
Workflow	A series of 1 or more tasks to perform. Workflows can call other workflows (building complex series of tasks)
Job	An instance of workflow. When you tell the API to do perform some workflow it creates a JobId which is used to check on the status of the job and retrieve the output (an asset or a dataset) of what the job produced. When you create a job you also specify what

Concept 1: What is data

The system was built to be able to handle a variety of incoming data types (text files, pdfs, word docs, etc....). These files are saved as assets and often time converted down to text for further processing. Each step that is taken during this processing will produce either an Asset or a Dataset (a group of assets) for easy review.

Concept 2: What is a Workflow

Workflows are just a series of steps you are asking our system to perform. You don't care that the work is broken down into X number of steps, all that you need to be concerned about is that you ask it to perform a high-level task by giving in a reference to some input data and as the result it produces some output data. In the background workflows can reference other workflows so you can build very complex processing, but still the consuming user only cares about the input and the output. Vettid will work with you and create custom workflows for your company, then all you need to do is call and consume the results

Concept 3: What is a Job

A job is just an instance of workflow in action. This allows you to call a workflow multiple times and still keep track of each individual job. We keep track of the instance by a Job Id. The time it takes to process the job depends on the complexity of the workflow and the volume of data the tasks have to process.

Step 1 Authentication

The Client Id and a Client Secret above will be used for step 1 of authentication only. This information is passed into as the first step of the process and exchanged for a Bearer token. The token is added to all subsequent calls for authentication from this point forward. The time to live (TTL) for this token can be modified if needed but we default to a 4 hours TTL.

Request:

```
curl -X POST https://vettd.auth0.com/oauth/token
-H 'Cache-Control: no-cache'
-H 'Content-Type: application/x-www-form-urlencoded'
-d'grant_type=client_credentials&audience=api.vettd.com%2Fvdo&client_id={CLIENT_ID_HERE}&client_secret={CLIENT_SECRET_HERE}
```

Example Response:

```
{
  "access_token": "eyJ0eXAiOi...DhgE-Q3-cycw",
  "scope": "",
  "expires_in": 14400,
  "token_type": "Bearer"
}
```

Step 2 Review list of WorkFlows available to use

Online Docs: <https://get.vettd.com/vdo/v1/docs/#!/Workflow/VdoV1WorkflowGet>

Request:

```
curl -X GET https://get.vettd.com/vdo/v1/workflow
-H 'Authorization: Bearer {ACCESS_TOKEN_HERE}'
-H 'x-api-key: {API_KEY_HERE}'
-H 'Cache-Control: no-cache'
-H 'Content-Type: application/json'
```

Example Response:

```
[
  {
    "id": "c2442327-fe44-4c78-cfb1-08d58de3f70a",
    "name": "Email Classify",
    "workflow": {
      "id": "9155cdbbc-c7f0-4a5a-1010-08d58de3f6f1",
      "name": "Classify Email",
      "inputs": [
        {
          "id": "3742ae35-d42c-4148-b70f-738bbc093d71",
          "name": "ModelId",
          "typename": "staticasset",
          "iocounter": 1
        },
        {
          "id": "1f26f5cc-2699-4d16-9932-73f73dd08edb",
          "name": "AssetId of Email File",
          "typename": "asset",
          "iocounter": 2
        }
      ],
      "outputs": [
        {
          "id": "a96af789-9540-4ab1-a6bc-d642537d8bd8",
          "name": "Classify Email",
          "typename": "asset",
          "iocounter": 6
        }
      ]
    },
    "version": 1
  },
  {... more workflows suppressed}
]
```

How to interpret the response

The response will return a list of workflows you have access to call.

1. Start by reviewing the name section of the results and locate the workflow you would like to call. Example above we will call "Email Classify".
2. Next locate the inputs section. This is where you will find the requirements of what inputs are required to start the workflow. The **typename** is important because it tells you if the input need is:

TypeName	Input Ignored	Description
asset		A reference to a single file
dataset		A reference to a group of files
staticasset	X	Predefined reference to a single file
staticdataset	X	Predefined reference to a group of files
static	X	Predefined variable

In the example above the workflow requires two inputs.

- The first one is the id of the model to run against which was already predefined when the workflow was created. This is for informational purposes only.
 - The second is telling us when need to reference an asset already in the system before we can call this workflow. It also tells us the Id of that variable so that when more than one is required as input the system can differentiate between them. We will need this information when calling the workflow in a few steps.
3. Next locate the outputs section. At the conclusion of the workflow it will produce either an asset or a dataset. This section tells you which type to expect when you download the results in the last step.

Step 3 Upload Data

We learned in the previous step that sample workflow will require an asset to process. We could reference an existing file and skip this step, but since this is our first time in the system we have no assets in the system yet. To change that lets upload a file.

Online Docs: <https://get.vettd.com/vdo/v1/docs/#!/Storage/VdoV1StorageAssetsPost>

Request:

```
curl -X POST https://vettd.auth0.com/vdo/v1/storage/assets
-H 'Authorization: {TOKEN_HERE}'
-H 'x-api-key: {API_KEY_HERE}'
-H 'Cache-Control: no-cache'
-H 'content-type: multipart/form-data;'
-F 'files=@C:\SampleEmail.txt'
-F 'datatype=EmailText'
-F 'datasetname=EmailContent'
```

At this point just create a text file using Notepad and save it locally to your computer. Type or paste in some sample text in the document and save. In the request your will reference the file in -F section, taking note the content type is `multipart/form-data`. The **datatype** and **datasetname** is used to help organize. The **datatype** can be what every you want, and the **datasetname** can be thought of as a folder to drop it into.

NOTE 1: If the **datasetname**(folder) does not exists it will be automatically created.

NOTE 2: If the file already exists inside the **datasetname**(folder) it will be overwritten.

Example Response:

```
{
  "datasetid": "6fd9c0f3-2bae-4478-9be4-08d59bea002a",
  "datasetname": "EmailContent",
  "assets": [
    {
      "id": "11112222-3333-4444-5555-666677778888",
      "createdate": "2018-04-06T18:12:47.8919207Z",
      "updatedate": "2018-04-06T18:12:47.8919207Z",
      "datatype": "EmailText",
      "name": "SampleEmail.txt "
    }
  ]
}
```

Take note of the id returned. This value will be used as the AssetId when calling the workflow.



NOTE: You can upload ZIP files as well. The API will unzip these files and add them to the same dataset on upload processing.

Step 4 Call Workflow/ Create a Job

To call a workflow you just need three pieces of information.

TypeName	Description
name	A name of your choosing for this instance of the workflow
workflowid	The id of the workflow you chose in the response of step 2
parameters	An array of input parameters which made up of the input id and the input value

In this example we just have one variable so we are passing in the id of it from Step 2 and the value(id) from Step 3

NOTE 1: The highlighted colors below match with the location it was pulled from above.

Online Docs: <https://get.vettd.com/vdo/v1/docs/#!/Job/VdoV1JobsPost>

Request:

```
curl -X POST https://vettd.auth0.com/vdo/v1/jobs
-H 'Authorization: Bearer {ACCESS_TOKEN_HERE}'
-H 'x-api-key: {API_KEY_HERE}'
-H 'Cache-Control: no-cache'
-H 'Content-Type: application/json'
-d '{
  "name": "Email Test 1",
  "workflowid": "c2442327-fe44-4c78-cfb1-08d58de3f70a",
  "parameters" : [{
    "id": "f26f5cc-2699-4d16-9932-73f73dd08edb",
    "value": "11112222-3333-4444-5555-666677778888"
  }]
}'
```

Example Response:

```
{
  "id": "99998888-7777-6666-5555-44443332222",
  "datatype": "jobid",
  "name": "Email Test 1"
}
```

A success response will contain a JobId. This means the processing has been put on the que to process. The processing time will vary depending on:

- Number of jobs and complexity of those jobs already existing in the que
- Complexity of the current workflow
- Internet and hardware communication health

Once you have an JobId you can check the status of that job at anytime.

Step 5 Check Job Status

Online Docs: <https://get.vettd.com/vdo/v1/docs/#!/Job/VdoV1JobsByJobidGet>

Request:

```
curl -X GET https://vettd.auth0.com/vdo/v1/jobs/99998888-7777-6666-5555-444433332222
-H 'Authorization: Bearer {ACCESS_TOKEN_HERE}'
-H 'x-api-key: {API_KEY_HERE}'
-H 'Cache-Control: no-cache'
-H 'Content-Type: application/json'
```

Example Response:

```
{
  "job": {
    "id": "7f6ea111-51d1-4b54-c5d5-08d59bea1c0",
    "name": "Email Test 1",
    "inputvalues": [
      {
        "id": "3742ae35-d42c-4148-b70f-738bbc093d71",
        "value": "26e117e1-4a38-4d22-61f5-08d59bea00af"
      },
      {
        "id": "1f26f5cc-2699-4d16-9932-73f73dd08edb",
        "value": "11112222-3333-4444-5555-666677778888"
      }
    ],
    "status": "Created",
    "createdate": "2018-04-06T18:19:05.4900101",
    "workflowid": "c2442327-fe44-4c78-cfb1-08d58de3f70a ",
    "ouputypename": "asset",
    "ouputvalue": "99999999-9999-9999-9999-999999999999"
  },
  "steps": [
    ..supressed
  ]
}
```

Review the **status** and it changes to "Complete" you will notice that the **ouputypename** and **ouputvalue** is also filled out as well. The **ouputvalue** will be the AssetId of the file created that contains your output results. Use this value to download the file in the next step.

Step 6 Download Job Output

Online Docs: <https://get.vettd.com/vdo/v1/docs/#!/Download/VdoV1StorageAssetsByAssetidDownloadGet>

Request:

```
curl -X GET https://vettd.auth0.com/vdo/v1/storage/assets/99999999-9999-9999-9999-999999999999/download
-H 'Authorization: Bearer {ACCESS_TOKEN_HERE}'
-H 'x-api-key: {API_KEY_HERE}'
-H 'Cache-Control: no-cache'
```

Example Response:

```
This will be a filestream of the asset
```

Congratulations you completed a full lifecycle of calling the Vettd Data Observatory

We realize this was a simple example but Vettd will work with you closely to create your company its own unique workflows tailored to your requirements and needs.

Have a suggestion for this document? Please email any suggestions on how to make this document better. We look forward to your feedback.