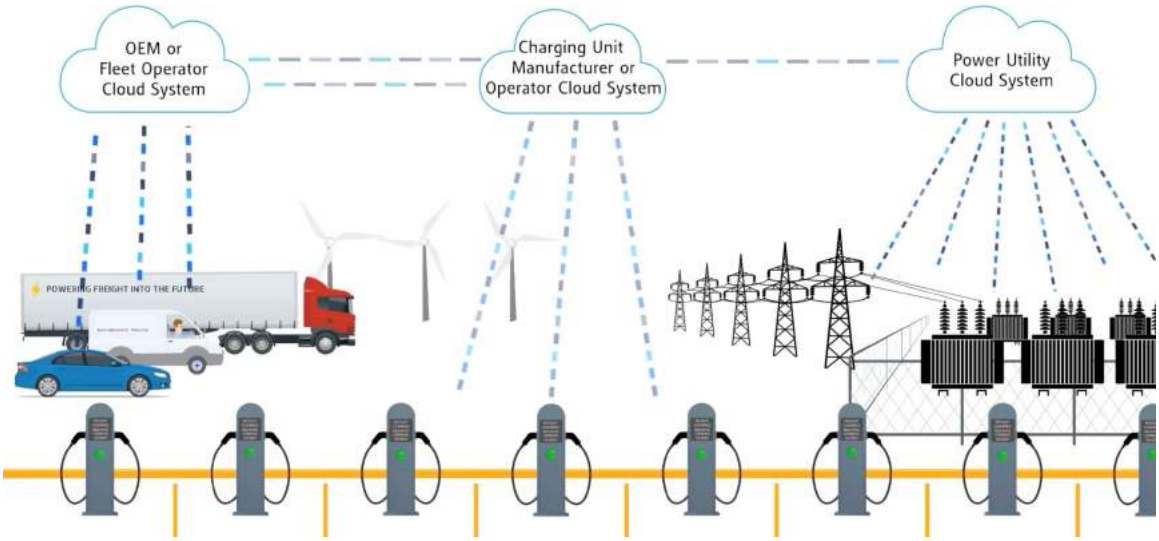
TERBINE®

TAKE THE PULSE OF THE EARTH™

TERBINE API SPECIFICATION



Last Revision: August 18, 2022
Version: 1.0.2

Contents subject to change. All rights reserved. www.terbine.com

TABLE OF CONTENTS

Contents

 Contents subject to change. All rights reserved. www.terbine.com

 Contents subject to change. All rights reserved. www.terbine.com

OVERVIEW

This document is a high-level overview of the TERBINE Application Programming Interface (API). This document is not intended to be an exhaustive list of available interfaces or services, but rather a smooth introduction to any client who intends to use the API to interface with the TERBINE marketplace services.

REFERENCED DOCUMENTS

This API Overview is built on information contained within the IoT Metadata Specification document. Example code is also provided along with a complimentary document titled TERBINE API Tutorial that has working examples showing usage of the TERBINE API.

API CATEGORIES

The TERBINE API can be broken down into the following categories:

- Organization / User Administration
- Organization / User Profile and Preferences
- Search
- Group / Metadata / Schema Administration
- Technical (Upload/Download)

SECURITY

API security will be implemented using the following security strategies:

- Java Web Token (JWT) is used for authentication. When a user successfully logs in, they receive a TOKEN from the server with a limited lifetime. This token is exclusive to that user, expires after a period of time, and must be passed in to all subsequent calls.
- Pre-defined TERBINE roles will provide authorization requirements for chosen functionality and resources.
- Wire-based security will be implemented using SSL/TLS.
- Disk based encryption will provide “data at rest” security as needed.
- All Personally Identified Information (PII) will be stored in an encrypted format and salted as needed. TERBINE policy is to avoid storage of any PII related data.
- Machine to machine (M2M) interaction will be supported. Identification for this will be accomplished with a combination of a provisioned API Key Identifier and Public Key.

The majority of the API will be secured, excluding basic browsing, login, and system overview services.

 Contents subject to change. All rights reserved. www.terbine.com

SERVICE MODEL

The TERBINE API is implemented using a REST model built on HTTP(s).
Within the REST model, the system is divided into groups of uniquely identified resources (users, groups, metadata, etc.) Generally, but not exclusively, resources use a Universally Unique Identifier (UUID) which is a 128 bit value. An example of a UUID is **`fb9967f1-eadf-465f-b11e-cbeda2c62bdf`**. These will be allocated and managed by the TERBINE backend.

HTTP VERBS

REST built on HTTP(s) uses standard HTTP verbs for managing resources.

- GET – read a resource by id or group of resources by name
- PUT – update a resource
- POST – create a new resource
- DELETE – delete a resource
- PATCH – this will have limited use and is used when a portion of a resource will be updated

EXAMPLES

GET

- *GET* `https://api.terbine.io/user/fb9967f1-eadf-465f-b11e-cbeda2c62bdf`
- *GET* `https://api.terbine.io/user/fb9967f1-eadf-465f-b11e-cbeda2c62bdf/securityanswers`

POST

- *POST* `https://api.terbine.io/user`
- *POST* `https://api.terbine.io/user/fb9967f1-eadf-465f-b11e-cbeda2c62bdf/securityanswers`

PUT

- *PUT* `https://api.terbine.io/user/fb9967f1-eadf-465f-b11e-cbeda2c62bdf`
- *PUT* `https://api.terbine.io/user/fb9967f1-eadf-465f-b11e-cbeda2c62bdf /api/metadata/eb9967f2-eadf-465f-b11e-cbeda2c62bdf`

DELETE

- *DELETE* `https://api.terbine.io/user/fb9967f1-eadf-465f-b11e-cbeda2c62bdf`
- *DELETE* `https://api.terbine.io/user/fb9967f1-eadf-465f-b11e-cbeda2c62bdf /api/metadata/eb9967f2-eadf-465f-b11e-cbeda2c62bdf`

 Contents subject to change. All rights reserved. www.terbine.com

RETURN AND ERROR CODES

HTTP Verb	CRUD Operation	Entire Collection (e.g. /users)	Specific Item (e.g. /users/{id})
GET	Read	200 (OK), list of users. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single user, 404 (Not Found), if ID not found or invalid.
POST	Create	201 (Created) Request Body with mandatory information for all resources. Response with new resources, ids and associated data.	201 (Created) Request Body with mandatory resource information. Response with new resource, id and associated data.
PUT	Update	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid for all resources in body	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid for specific resource
DELETE	Delete	404 (Not Found), unless implemented delete the whole collection—not normally desirable.	200 (OK). 404 (Not Found), if ID not found or invalid for resource

As a general rule, success of any operation can be checked by looking for a return code in the range of 200 – 299.

Any return code in the 300 – 399 range will denote a moved resource and will be used sparingly. This will be documented thoroughly at the specific API level.

Any return code in the 400 range denotes a client error. Most common examples of these are:

- 400 – Bad Request: The correct method was used, but there was missing information in the body or a malformed entity was being passed.
- 401 – Unauthorized: User needs to be authenticated and use a correct API Key / Session token.
- 403 – User is Not Authorized: User is not allowed to carry out this operation.
- 404 – Not Found: Resource with this ID was not found.
- 405 – Method Not Allowed: A resource was called with a form not allowed (e.g. correct form for a GET but not a POST).

Server errors will be denoted by HTTP status codes in the 500 range. Most common are:

- 500 – Internal Server Error: Unexpected error at server level
- 503 – Service Unavailable: API or service not available.

Optionally, specific information may be passed back detailing what is missing or why the error occurred.

 Contents subject to change. All rights reserved. www.terbine.com

HEADER INFORMATION

REST calls will often require multiple header values to be set.

Within TERBINE, we have multiple common header values that may be required dependent on the value. For example, the majority of services are protected by requiring an **Authorization** header.

There is also the ability to set a tracking header value, **X-TRACKING-ID,** from the client**.** This id will be used for all logging on the server for that request. For any client logging, a new id would be created for each request made to the server. In this way, logging entries from client to server and back in response can be linked by this id. The tracking id is very useful for user error reporting and troubleshooting. If the client does not pass a tracking id, one will be created on the server and passed back in the response header.

DATA EXCHANGE FORMAT

All data is interchanged in the API via JSON. JSON is a simple text-based message format that is often used with RESTful Web services.

JSON EXAMPLE

```
{
   "id":"98931ef4-c423-42f5-869c-88eae5b39dfa",
   "orgId":"6d94ff88-4b41-41d0-b3a4-d14d89da0939",
   "identifier":{
      "extId":"A634321"
   },
   "meta":{
      "name":"Medical Test Metadata",
      "description":"Medical Test Metadata Description",
      "version":"v1"
   },
   "dataset":{
      "extId":"SRC_EXT_ID1",
      "type":30,
      "sensorInfo":{
         "id":null,
         "type":44,
         "make":"make 123",
         "model":"model 123",
         "comment":"Specific sensor comment"
      },
      "schemaInfo":{
```

```
        "format":6,
        "type":81,
        "property":";",
        "body":null
     },
     "comment":"Comment on sample dataset information"
},
... Remainder of response omitted.
```

EXPLANATION OF STRUCTURE

1. JSON in the TERBINE API is not wrapped (type indicator at the root of the data).
2. Identifiers are case insensitive in the API. For example, orgId and orgid are equivalent.
3. UUID – Widely used unique identifiers for resources within TERBINE. An example of a UUID value is **`6d94ff88-4b41-41d0-b3a4-d14d89da0939`**.
4. Standard Date Format – this is an example of a standard date format, ISO 8601 format and transmitted in UTC. It is up to the client to display these in any localized format. Date only format will be YYYY-MM-DD.
5. Embedded (Children) Data – This is a sub entity of the owning item, **metadata**, called **schema**. It has a child relationship to the parent. Dependent on context, it may also live as a top-level resource.
6. Null Values – How non-existent values are transmitted. If passed in a PUT or POST, they will overwrite the existing value with the NULL value.
7. Embedded Quotes – This shows a string with embedded quotes. Therefore, the \ character is preceding them.
8. Array of Embedded Objects – When there is a 0 to N (or 1 to N) list of objects, the embedded objects are placed in an array.
9. Array of Single Data Items – When there is a list of single items, they are embedded in an array.

CLIENT DEPENDENCIES

TERBINE API is built completely client agnostic. All API interfaces have neither client specific information nor variations of the definitions dependent on client type. Therefore, a Web client, iOS or Android client will find the use of the TERBINE API seamless. For analytics purposes, see below.

TERBINE has a variety of libraries and SDK's built including Tableau, PowerBI, Java, JavaScript and Python for integration into application or language specific custom libraries.

DEVICE CATEGORIES

TERBINE API allows passing of device info as part of the API. This is done through the **X-DEVICE-HDR** header. Note we do not use the user-agent header for this purpose.

The following categories are supported:

- WEB
- ANDROID
- ANDROID_TABLET
- IOS_IPHONE
- IOS_IPAD

Note as of writing, only WEB is supported.

TERBINE HTTP(S) REQUESTS

BASE URI

The Base Uniform Resource Identifier (URI) used within each HTTP(s) request is https://api.terbine.io**.** This base URI is combined with the Uniform Resource Locator (URL) Path to create the complete URL passed in the request.

PAGINATION

Paginated requests allow passing these items as query parameters in the request URL:

- pageNum = page to be returned, starting from 1
- pageSize = size of returned result set
- orderBy = column to sort by
- order = ASC or DESC

AUTHORIZATION

LOGIN

The Login request logs a user into the TERBINE system. The response provides the **token** field to be used in all subsequent requests requiring authentication. Clients should also note the **userid** and **orgid** fields in the response, as these are often needed for specific services. These are system wide unique identifiers for the login user and the user's organization.

- Request:
  `POST`

 Contents subject to change. All rights reserved. www.terbine.com

- URL Path:
  `/api/auth/login`

- Header:
  `Content-Type: application/json`

- Body:

```
{
    "username": "<user>",
    "password": "<password>"
}
```

Sample Response:


```
{
   "lastlogin": "2017-08-28T21:44:48Z",
   "username": "user name",
   "displayname": "user display name",
   "token":
   "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0ZXJiaW5lYWRtaW4iLCJleHAiOjE1MDM5Nj
   I1MzV9.z0KDmCubJmio7JsFGK8TE4yQyuzMGKCOLZMcrmJV_ME",
   "userid": "aab864d2-74fc-44c1-b0d3-632829719929",
   "orgid": "6d94ff99-4b41-42d0-b3a4-d14d89da0838",
   "orgname": "company name",
   "tokenlifeseconds": 4200,
   "numberNotices": 1,
   "tabs": [
        "ACCOUNT",
        "SUMMARY",
        "FINANCE",
        "LEGAL",
        "WORKSPACE",
        "METADATA",
        "TECHNICAL",
        "DATASETS",
        "POLICIES"
   ],
}
```

To utilize the token field in subsequent calls requiring authentication, add within a header as seen here:

```
Authorization:  bearer
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ0ZXJiaW5lYWRtaW4iLCJleHAiOjE1MDM5NjI1MzV9.z0KDmCubJmio7JsFGK8TE4yQyuzMGKCOLZMcrmJV_ME
```

SEARCH

SEARCH SERVICE

The Search Service request sends a query to search TERBINE's available metadata instances and allows for customization of displayed results by changing the pagination parameters within the URL. Clients should note the **id** field (the metadata instance's GUID) within each search result item, as this is required in requests for metadata by id.

- Request:
  `POST`

- URL Path:
  `/api/search/v2/metadata?pageNum=1&pageSize=5&sort=DATE ADDED&order=DESC`

- Header:
  `Optional`

- Body:
```
{
    "text": "<query>"
}
```

Sample Response:


```
[
   {
      "metadata": null,
      "type": 2,
      "source": "Organization",
      "id": "90189158-8362-4df5-b147-f97b10151dfc",
      "orgId": "893528a7-5d5c-45ee-87a9-e35bf0635e27",
      "ownerOrgId": null,
      "alphaSort": "Vehicle-sampled Ambient Air Temperature / United States Nationwide",
      "title": "Vehicle-sampled Ambient Air Temperature / United States Nationwide",
      "description": "Timestamped & geolocated samplings of outside air temperature on vehicles equipped with the Voyomotive On Board Diagnostics (OBD-II standard) capture / transceiver device. Fleet sampling of over 500 vehicles in various geographies. \n",
```


 Contents subject to change. All rights reserved. www.terbine.com

```
"dateIndexed": "2017-08-28T01:13:58Z",
"dateAdded": "2017-07-01T18:52:13Z",
"createDate": "2017-07-01T18:52:13Z",
... Rest of response omitted.
```

Please note that the search service does not require authentication. Unauthenticated calls will result in any metadata not covered by a policy.

Authenticated searches will return all metadata owned by the user's organization, public metadata owned by other organizations, and metadata from other organizations allowed by a covered policy.

 Contents subject to change. All rights reserved. www.terbine.com

METADATA

Note Metadata Feeds are also known as Single Instances. This is usually within the context of when Metadata Feeds are added to Group Instances.

DOMAIN TYPES

The Domain Types request retrieves all metadata domain types. Clients should note the **id** field in the response, as this is needed to retrieve the Domain Values.

- Request:
  `GET`
- URL Path:
  `/api/domain/type`
- Header:
  `Optional`

Response:
JSON array with each type of domain that can be used within metadata.

DOMAIN VALUES

The Domain Values request retrieves the values of a chosen Domain Type. An **id** number associated with a domain is retrieved from the Domain Types request and used as a parameter.

- Request:
  `GET`
- URL Path:
  `/api/domain/<id>`
- Header:
  `Optional`

Response:
JSON with each value for passed domain type

 Contents subject to change. All rights reserved. www.terbine.com

INTERVAL TYPES

Returns list of interval types (e.g. second, minute, hours, day) for designating continuous feed interval and load frequency.

- Request:
  `GET`
- URL Path:
  `/api/metadata/intervaltype`
- Header:
  `Optional`

Response:
JSON array

METADATA TYPES

Returns list of metadata types (e.g. archival or continuous) with associated text and ids.

- Request:
  `GET`
- URL Path:
  `/api/metadata/type`
- Header:
  `Optional`

Response:
JSON array

METADATA STATUS

Returns list of metadata status codes (e.g. active, not active) with associated text and ids.

- Request:
  `GET`
- URL Path:

 Contents subject to change. All rights reserved. www.terbine.com

```
/api/metadata/status
```

- Header:
  `Optional`

Response:
JSON array

METADATA SERVICES

GET ALL METADATA FOR AN ORGANIZATION

The Get All Metadata for an Organization retrieves a list of all metadata associated with an organization. The request allows for pagination, so can take pageNum and pageSize as optional URL query parameters.

- Request:
  `GET`

- URL Path:
  `/api/metadata`

- Header:
  `Authorization: bearer <token>`

Response:
JSON array of organization metadata objects

GET METADATA BY ID

The Get Metadata by ID request returns the metadata field values for a certain metadata instance using the **id** field retrieved from the Search request. Clients should note the **id** field within the dataset array in the response, as this is required as a **dataset ID** used in various calls such as listing and downloading multiple files associated with that metadata instance.

- Request:
  `GET`

- URL Path:
  `/api/metadata/<id>`

 Contents subject to change. All rights reserved. www.terbine.com

- Header:
  `Authorization: bearer <token>`

Response:
JSON representation of a single metadata object.

INSERT METADATA

The Insert Metadata request creates a new metadata instance using information placed in the body of the request.

- Request:
  `POST`

- URL Path:
  `/api/metadata`

- Header:
  `Authorization:  bearer <token>`

- Body:
  `Metadata JSON Representation`

Response:
JSON representation of new inserted metadata object with IDs created.

UPDATE METADATA

The Update Metadata request updates an existing metadata instance using information placed in the body of the request.

- Request:
  `PUT`

- URL Path:
  `/api/metadata`

- Header:
  `Authorization:  bearer <token>`

- Body:

 Contents subject to change. All rights reserved. www.terbine.com

```
Metadata JSON Representation
```

Response:
JSON representation of newly updated metadata object.

DELETE METADATA

The Delete Metadata request disables an existing metadata instance.

- Request:
  `DELETE`
- URL Path:
  `/api/metadata/<id>`
- Header:
  `Authorization: bearer <token>`

Response:
HTTP 200 with message metadata was disabled.

CLONE METADATA

The Clone Metadata request copies an existing metadata instance for simplification of creating a new related metadata instance.

- Request:
  `POST`
- URL Path:
  `/api/metadata/<id>/clone`
- Header:
  `Authorization: bearer <token>`

Response:
HTTP 200 with newly created (cloned) metadata

UPDATE METADATA STATUS

The Update Metadata Status uses the numeric id from the **/metadata/status** request..

 Contents subject to change. All rights reserved. www.terbine.com

- Request:
  `PUT`

- URL Path:
  `/api/metadata/status/{statusid}`

- Header:
  `Authorization: bearer <token>`

- Body:
  `N/A`

Response:

JSON message if updated.

WORKSPACE SERVICES

LIST WORKSPACE

The List Workspace function displays a list of metadata instances currently added to a client's workspace. Similar to the Search function, optional pagination parameters can be used to customize the display of the resulting list (pageNum, pageSize, order, and orderBy). Clients should note the **id** for each list item, as this is needed as a **workspace ID** used in other workspace calls.

- Request:
  `GET`

- URL Path:
  `/api/dashboard/organization/<orgid>/workspace`

- Header:
  `Authorization: bearer <token>`

Response:
JSON array of all items added to the client workspace

ADD TO WORKSPACE

The Add to Workspace request adds a chosen metadata instance to the user's workspace where it can be later flagged, locked in for download, or deleted. The necessary **id** can be retrieved from an item in the Search request and the **orgid** is retrieved from the Login request.

- Request:
  `POST`

- URL Path:
  `/api/workspace/metadata/<id>?orgId=<orgid>`

- Header:
  `Authorization: bearer <token>`

Response:
JSON representation of metadata instance object with a status of "1" if added successfully.

LOCK TO WORKSPACE

 Contents subject to change. All rights reserved. www.terbine.com

The Lock to Workspace request locks a metadata instance found in the workspace and activates the dataset/stream for use in the user's dashboard. The necessary **workspaceid** can be retrieved from an item in the List Workspace request and the **orgid** is retrieved from the Login request.

- Request:
  `PUT`

- URL Path:
  `/api/dashboard/organization/<orgID>/workspace/<workspaceid>/locked`

- Header:
  `Authorization: bearer <token>`

Response:
JSON representation of metadata instance object with a status of "3" if locked successfully.

FLAG IN WORKSPACE

The Flag in Workspace request flags a metadata instance found in the workspace to distinguish it from other added workspace items. The necessary **workspaceid** can be retrieved from an item in the List Workspace request and the **orgid** is retrieved from the Login request.

- Request:
  `PUT`

- URL Path:
  `/api/dashboard/organization/<orgid>/workspace/<workspaceid>/flagged`

- Header:
  `Authorization: bearer <token>`

Response:
JSON representation of metadata instance object with a status of "2" if flagged successfully.

 Contents subject to change. All rights reserved. www.terbine.com

DELETE FROM WORKSPACE

The Delete from Workspace request removes a metadata instance located in the workspace. The necessary **workspaceid** can be retrieved from an item in the List Workspace request and the **orgid** is retrieved from the Login request.

- Request:
  `DELETE`

- URL Path:
  `/api/dashboard/organization/<orgid>/workspace/<workspaceid>`

- Header:
  `Authorization:  bearer <token>`

Response:
JSON representation of metadata instance object with a status of "1" if added successfully.

 Contents subject to change. All rights reserved. www.terbine.com

DATASET

OVERVIEW

This overview provides API descriptions for Dataset and Dataset Content (files).

The fundamental part of understanding dataset and content comes from the explanation of their terminology. A **dataset** (avoiding use of the word metadata) is a description of the physical content that is uploaded for a given **metadata** configuration. The id field in the dataset tab is the **"dataset id"**. The **metadata** owns the dataset. A dataset can be **inbound** or **outbound**. These terms are always used in relation to the user, so inbound is purchased datasets and outbound is provided datasets.

A metadata configuration, per the data model, can have more than one dataset. A dataset can contain multiple files. A dataset can therefore be thought of as a container for a group of same structured files.

The actual physical data might be addressed with any of these terms: **content, file,** or **attachment**. This is still a bit nebulous as we explore different use cases within the system (e.g. streaming data, external storage, etc).

For our case, when a user enters metadata it also adds a dataset as an outbound type.

DATASET SERVICES

GET LIST OF FILES FOR DATASET

The Get List of Files for Dataset request retrieves an array of content descriptions with information about each file. Note the **id** in this response is used in various calls as a **content ID**. Additionally, fields such as dateUploaded, dateStart, and originalName can be used for display purposes.

- Request:
  `GET`

- URL Path:
  `/api/dataset/<datasetid>/content`

- Header:
  `Authorization: bearer <token>`

 Contents subject to change. All rights reserved. www.terbine.com

Sample Response:


```
 [
    {
      "id": "e72ab85d-bc44-4b41-9c5b-0c6f0ae54efc",
      "metadataId": "d2fb8dd5-9151-40a0-a794-3a35bf893103",
      "datasetId": "b233d348-963a-45b4-a04b-c46becb846c0",
      "externalId": "be-292922432",
      "organizationId": "6d94ff88-4b41-41d0-b3a4-d14d89da0939",
      "type": 1,
      "status": 1,
      "fileStoreLocation": "6d94ff88-4b41-41d0-b3a4-
      d14d89da0939/d2fb8dd5-9151-40a0-a794-
      3a35bf893103/UItzTNVeTy7AKZt4AerYv2Ru.terbine.json",
      "fileExt": "json",
      "size": 22742,
      "fileHash": "b6aa9e273e8de94948a582b78f33207b6b9890b4",
      "lastModifyTime": "2017-06-26T20:42:46Z",
      "dateUploaded": "2017-06-26T20:42:45Z",
         "dateStart": "2017-06-24T17:45:40Z",
      "dateEnd": null,
      "deleted": 0,
      "originalName": "voyo-2017-04-20-18-30-01.json"
    },
... Rest of response omitted.
```


UPLOAD SINGLE FILE

The Upload Single File request returns a record with all information related to the file to upload, including its new unique ID.

- Request:
  `POST`

- URL Path:
  `/api/dataset/<datasetid>/content/upload`

- Header:
  `Content-Type: application/x-www-form-urlencoded`
  `Authorization: bearer <token>`

- Body:
  File info is sent as file.
  Below example is sent as a form with key info. Note all fields except filename are optional, though dataStart is recommended as it can inform when data is valid.

```
{
    "fileName": "filetoupload.csv",
    "externalId": "be-232323",
    "dataStart": "2017-06-24T21:45:40Z",
    "dataEnd": "2017-06-26T21:45:40Z",
}
```

Sample Response:


```
{
    "id": "0d5d81c0-dc77-4d85-8bd7-09462df1dfb8",
    "metadataId": "bca81462-e586-4fa4-ab72-37d3306b93ee",
    "datasetId": "47bce565-36c6-46e4-956a-c7cd14e9520a",
    "externalId":
    "766f796f2d323031372d30342d32302d31382d33302d30312e6a736f6e",
    "organizationId": "6d94ff88-4b41-41d0-b3a4-d14d89da0939",
    "type": 1,
    "status": 1,
    "fileStoreLocation": "6d94ff88-4b41-41d0-b3a4-
    d14d89da0939/bca81462-e586-4fa4-ab72-
    37d3306b93ee/bLp0GuBvFnMJJISiAq7cU3lu.terbine.json",
    "fileExt": "json",
    "size": 22742,
    "fileHash": "b6aa9e273e8de94948a582b78f33207b6b9890b4",
    "lastModifyTime": "2017-06-29T22:33:39Z",
    "dateUploaded": "2017-06-29T22:33:21Z",
    "dateStart": "2017-06-24T21:45:40Z",
    "dateEnd": null,
    "deleted": 0,
    "originalName": "voyo-2017-04-20-18-30-01.json"
}
```


DOWNLOAD SINGLE FILE

The Download Single File request provides a download of a single file located within a dataset.

- Request:
  `GET`

- URL Path:
  `/api/dataset/content/<contentid>/download`

- Header:
  `Authorization: bearer <token>`

Response:
File to download streamed to client.

DOWNLOAD MULTIPLE FILES

The Download Multiple Files request will download, in a zip file, the number of files specified by the maximum parameter. If this parameter is not provided, it will default to 50. The files will be in descending order from the most recent. Zip files will have the current date and time as their names.

- Request:
  `GET`

- URL Path:
  `/api/dataset/<datasetid>/content/download?maximum=10`

- Header:
  `Authorization: bearer <token>`

Response:
Zip file containing multiple files

DOWNLOAD SINGLE FILE FOR PREVIEW

This request is similar to downloading a single file, only a length parameter can be included to allow downloading of a certain number of bytes using a length parameter. If the length query parameter is not provided, it defaults to a value of 1024. *(note within Terbine this will currently only work for text files, and not types such as PDF, XLS, XLSX etc.)*

- Request:
  `GET`

- URL Path:
  `/api/dataset/content/<contentid>/download/preview?length=1024`

- Header:
  `Authorization: bearer <token>`

Response:
Portion of the file to download is streamed to the client. If the file is smaller than requested, the full length of the file will download.

 Contents subject to change. All rights reserved. www.terbine.com

DELETE SINGLE ITEM

The Delete Single Item request deletes a single file from a dataset.

- Request:
  `DELETE`

- URL Path:
  `/api/dataset/content/<contentid>`

- Header:
  `Authorization: bearer <token>`

Response**:**
Success message if deletion was completed.

DELETE ALL ITEMS IN DATASET

This request deletes all files within a dataset.

- Request:
  `DELETE`

- URL Path:
  `/api/dataset/<datasetid>/content`

- Header:
  `Authorization: bearer <token>`

Response:
Success message if deletion was completed.

 Contents subject to change. All rights reserved. www.terbine.com

GROUP INSTANCES

OVERVIEW

A Group Feed type of feed instance that aim to provide improved organization and control over **Single** Instances and their clones. Group Instances will have a schema that matches its associated Single Instances. A Group Instance is also properly indexed within the Terbine system so that it is searchable by both humans and machines, allowing them to be returned in the Search results along with Single Instances.

GET ALL GROUP FOR AN ORGANIZATION

The Get All Group for an Organization retrieves a list of all groups associated with an organization. The request allows for pagination, so can take pageNum and pageSize as optional URL query parameters. Clients should note that within the **metadatas** section there is a list of all Single Instance feeds associated with the group.

- Request:
  `GET`
- URL Path:
  `/api/metadata`
- Header:
  `Authorization: bearer <token>`

Response:
JSON array of organization's Group objects

GET GROUP BY ID

The Get Group by ID request returns the group field values for a certain group instance using the **UUID** field retrieved from the Search request. Clients should note that within the **metadatas** section there is a list of all Single Instance feeds associated with the group.

- Request:
  `GET`
- URL Path:
  `/api/metadata/<id>`
- Header:
  `Authorization: bearer <token>`

 Contents subject to change. All rights reserved. www.terbine.com

Response:
JSON representation of a single Group object.

INSERT GROUP

The Insert Metadata request creates a new metadata instance using information placed in the body of the request.

- Request:
  `POST`
- URL Path:
  `/api/metadata`
- Header:
  `Authorization:  bearer <token>`
- Body:
  `Group JSON Representation`

Response:
JSON representation of new inserted Group object with IDs created.

UPDATE GROUP

The Update Metadata request updates an existing metadata instance using information placed in the body of the request.

- Request:
  `PUT`
- URL Path:
  `/api/metadata`
- Header:
  `Authorization:  bearer <token>`
- Body:
  `Group JSON Representation`

 Contents subject to change. All rights reserved. www.terbine.com

Response:
JSON representation of newly updated Group object.

DELETE GROUP

The Delete Metadata request disables an existing metadata instance.

- Request:
  `DELETE`

- URL Path:
  `/api/metadata/<id>`

- Header:
  `Authorization:  bearer <token>`

Response:
HTTP 200 with message metadata was disabled.

UPDATE GROUP STATUS

The Update Metadata Status uses the numeric id from the **/metadata/status** request.

- Request:
  `PUT`

- URL Path:
  `/api/metadata/group/status/{statusid}`

- Header:
  `Authorization:  bearer <token>`

- Body:
  `N/A`

Response:
JSON message if updated.

GET CLONES NOT IN GROUP

This returns all qualifying Single Instance feeds that are not already assigned to a given group. Can be used for updating a group (e.g. adding new single instance feeds). This can only be used for a Group Instance that exists.

- Request:
  `GET`
- URL Path:
  `/api/metadata/group/diff/clones/<group id>`
- Header:
  `Authorization: bearer <token>`

Response:

JSON representation of a Single Instance metadata object.

GET CLONES FOR GROUP

This returns all qualifying Single Instance feeds that can be assigned to a group based on a specific **Master Single Instance).** The UUID for the master is passed and all Single Instances that conform to the same attributes are returns. This is used when a Group is being created.

- Request:
  `GET`
- URL Path:
  `/api/metadata/group/clones/<master id>`
- Header:
  `Authorization: bearer <token>`

Response:

JSON representation of a Single Instance metadata objects.

 Contents subject to change. All rights reserved. www.terbine.com

CONTINUOUS FEEDS

OVERVIEW

Continuous Feeds (aka live feeds) can be accessed via the REST API by passing in the feed id and a date from or a date range to return data. Other methods such as via Kafka, s3 bucket or even other queuing products like Pulsar or RabbitMQ can be set up to access continuous feed data.

ACCESSING CONTINUOUS FEEDS

Note most continuous feeds within Terbine are not generally live as in a continuous stream but live in the context of being continually updated. The update frequency is dictated by the provider of the data and may be anywhere from 5 minutes, every hour or even once per day.

The update and load frequency can be found in the Technical Tab for a continuous feed. The update frequency is the interval on how often and in what interval the sensor updates the feed. The load frequency is how often and in what interval the data is delivered to Terbine. These values are often not the same.

The Rest API allows for data to be retrieved using a from date or date range. The returned data, dependent on how the individual data is sent to Terbine by the provider may require some manipulation by the consumer. Example is CSV files may have multiple header records returned, JSON may require an array section to be added, etc. We are constantly updating this as we go.

In addition to the data, the following important response header values will be returned:

- X-RECORDS-RAW – number of raw sensor event records that were stored. Note this is not records as in lines in a CSV, but number of deliveries from the sensor, which may contain (and usually do) multiple sensor records.
- X-LOWER-DATE – this is the first date within the range passed in that data was delivered for this feed id. Example is 2020-08-01T00:00:00Z. This will not be returned if X-RECORDS-RAW is 0
- X-UPPER-DATE – this is the most recent date where data was delivered for this feed id within the range passed in. Example is 2020-08-15T00:00:00Z. This will not be returned if X-RECORDS-RAW is 0.

  This value can be used when gathering data continuously in a loop in some interval by incrementing the value by 1 second and use as the next **datefrom** value in the REST call.

 Contents subject to change. All rights reserved. www.terbine.com

GET DATA WITH FROM DATE

This retrieves data starting from a certain date. The date passed must be in must be in ISO-8601 format (**2020-07-31T09:23:05Z**). This call takes a **limit** query parameter that sets the maximum number to return in one call, the default is 25.

- Request:
  `GET`
- URL Path:
  `/api/continuous/metadata/<metadataid>/datefrom/<datefrom>/download?limit=25`
- Header:
  `Authorization: bearer <token>`

Response:
Data returned in either JSON, CSV, TXT or XML starting from a certain date.

If no data is found for current feed id, the request will still return a 200, but a JSON message stating "No Data Found".

.

GET DATA WITH FROM AND TO DATE

This retrieves data between a date range. The dates passed must be in must be in ISO-8601 format (**2020-07-31T09:23:05Z**). This call takes a **limit** query parameter that sets the maximum number to return in one call, the default is 25.

- Request:
  `GET`
- URL Path:
  `/api/continuous/metadata/<metadataid>/datefrom/<datefrom>/dateto/<dateto>/download?limit=25`
- Header:
  `Authorization: bearer <token>`

Response:
Data returned in either JSON, CSV, TXT or XML within a date range.

If no data is found for current feed id, the request will still return a 200, but a JSON message stating "No Data Found".

 Contents subject to change. All rights reserved. www.terbine.com