

CLEAN API Version 2.0

Summary

The CLEAN API Version 2.0 allows for specific data to be pulled from CLEAN by third party applications. Each API user is allowed 3000 requests per day and after 3000 requests all future requests are blocked until the next day.

Enabling API Access

Only CLEAN users with the *API* role can generate API tokens.

To create an *API* user, a CLEAN user with the *Account Administrator* role should login to CLEAN and visit the *Administration > Manage Users* area to create a new user with the *API* role.

Once the API user is created, they should login to CLEAN and can generate an API token.

For security reasons API tokens are not displayed after initial creation. If the API token is lost a new can be generated from CLEAN by logging as the API user.

Note on Access

To protect users and data, the CLEAN API uses a single origin policy for requests made to the server. To successfully retrieve data, users must make requests using a backend technology like a NodeJS Server, Apache, Python, or PHP script designed to act as an intermediary for the client. More details are available by contacting CLEAN API Support.

Types of Available Data

Accounts: Retrieve the list of accounts for the user

Ready to Collect: Retrieve the list of components that are READY to collect

Collection Summary: Retrieve a summary of collections for a specified time

Account Assets: Retrieve the list of assets for an account

Alerts: Retrieve the list of alerts

Requests

Any tool that can make an HTTPS GET request can communicate with the CLEAN server using this API by forming the appropriate request.

All requests should be made using HTTPS so that the traffic between the requestor and the CLEAN is encrypted.

The GET request must contain two components:

- Header containing the key "X-Token" with value of your API Token
- URL Parameters

URL GET Parameters

The following chart lists the (7) supported GET parameters:

Parameter	Required	Type
action	Yes	Predefined String
objectType	Yes	Predefined String
accountFilter	No	Comma Separated Value List
groupFilter	No	Comma Separated Value List
serialNumberFilter	No	Comma Separated Value List
startTime	No	UNIX Timestamp in milliseconds
endTime	No	UNIX Timestamp in milliseconds

There are (5) objectTypes that can be used with API v2.0. The chart below has the objectTypes in the right most columns matched to the (7) URL parameters (rows) which may be used with that objectType. Parameters that are not marked as Optional or Required are not supported.

Parameter		Supported Parameters for each objectType				
objectType	=	accounts	collectionSummary	assets	alerts	collectionReady
action	=	load	load	load	load	load
accountFilter	=	Optional	Optional	Optional	Optional	Optional
groupFilter	=	-	-	Optional	Optional	Optional
serialNumberFilter	=	-	-	Optional	Optional	Optional
startTime	=	-	Required	-	-	-
endTime	=	-	Required	-	-	-

Returned Results

Error Code

Each request will return an error code indicating whether the request contains any errors or was OK.

The set of possible returned values is shown in the chart below, along with the HTTP status code it is roughly analogous to.

errorCode	Equivalent HTTP status code
STATUS_OK	200
INVALID_REQUEST	400
FORBIDDEN	403
TOO_MANY_REQUESTS	429
STATUS_NOT_VALID	500
SERVICE_UNAVAILABLE	503

Responses

When a request to CLEAN is successful, a response body will be returned to the caller in the form of a set of JSON objects and arrays.

Note that it's possible a successful request may not contain any results, in that case the returned JSON objects/arrays may be empty. This can occur if there are no item(s) that meet the request criteria.

Request Examples

Accounts

Retrieve a list of accounts, and any groups in the accounts, that are accessible with the given API Token.

Required Parameters:

- *objectType=accounts*
- *action=load*

Optional Parameters:

- *accountFilter=[CSV list of account ids]*

Returned Results:

KEY	TYPE	DESCRIPTION
accountId	integer	The CLEAN identifier for the account
name	string	The name of the account
groupId	integer	The CLEAN identifier for the group
groupName	string	The name of the group

Example GET request:

<https://api.bigbelly.com/api/v2?action=load&objectType=accounts&accountFilter=5,1650>

Example Results:

```
{
  "errorCode": "STATUS_OK",
  "accounts": [
    {
      "accountId": 1650,
      "accountName": "City Sixteen Fifty",
      "groups": [
        {
          "groupName": "All Parks",
          "groupId": 4322
        },
        {
          "groupName": "All Schools",
          "groupId": 432
        }
      ]
    },
    {
      "accountId": 5,
      "accountName": "City Five",
      "groups": [
        {
          "groupName": "1 Sanitation Area 1",
          "groupId": 1217
        },
        {
          "groupName": "2 Sanitation Area 2",
          "groupId": 1218
        },
        {
          "groupName": "3 Sanitation Area 3",
          "groupId": 1219
        },
        {
          "groupName": "4 Sanitation Area 4",
          "groupId": 1220
        }
      ]
    }
  ]
}
```

Example of Retrieving Accounts with Python:

```
import requests

url = "api.bigbelly.com/api/v2"

querystring = {"objectType":"accounts","action":"load"}

headers = {
    'X-Token': "n7q7V8mX4782xcTAn5Hc", // Replace with your API Key
    'Cache-Control': "no-cache"
}

response = requests.request("GET", url, headers=headers,
params=querystring)

print(response.text)
```

Ready to Collect

Retrieve summary totals for all components and those which are “READY” to be collected based on alert, age, or fullness threshold. Returned results are included for each waste stream type included in the dataset.

Required Parameters:

- *objectType=collectionReady*
- *action=load*

Optional Parameters:

- *accountFilter=[CSV list of account ids]*
- *groupFilter=[CSV list of group ids]*
- *serialNumberFilter=[CSV list of serial numbers]*

Returned Results:

KEY	TYPE	DESCRIPTION
accountId	integer	The CLEAN identifier for the account
accountName	string	The name of the account
allStreamsTotal	integer	Number of components in account
allStreamReady	integer	Number of components that are READY to collect
trashTotal	integer	Number of TRASH components
trashReady	integer	Number of TRASH components that are READY to collect
bottlesCansTotal	integer	Number of BOTTLES/CANS components
bottlesCansReady	integer	Number of BOTTLES/CANS components that are READY to collect
paperTotal	integer	Number of PAPER components
paperReady	integer	Number of PAPER components that are READY to collect
singleStreamTotal	integer	Number of SINGLE STREAM components
singleStreamReady	integer	Number of SINGLE STREAM components that are READY to collect
compostablesTotal	integer	Number of COMPOSTABLES components
compostablesReady	integer	Number of COMPOSTABLES components that are READY to collect
notReadyTotal	integer	Number of components that are NOT READY to collect
ageTotal	integer	Number of components that are AGE ready to collect
fullnessTotal	integer	Number of components that are FULLNESS ready to collect
alertTotal	integer	Number of components that are ALERT ready to collect

Example GET request:

<https://api.bigbelly.com/api/v2?action=load&objectType=collectionReady&accountFilter=5,1650>

Example Results:

```
{
  "collectionReady": [
    {
      "trashReady": 416,
      "allStreamsTotal": 1251,
      "allStreamsReady": 469,
      "ageTotal": 0,
      "accountName": "City Five",
      "fullnessTotal": 287,
      "trashTotal": 986,
      "bottlesCansTotal": 1,
      "bottlesCansReady": 0,
      "singleStreamReady": 53,
      "singleStreamTotal": 264,
      "accountId": 5,
      "notReadyTotal": 1072,
      "alertTotal": 211
    },
    {
      "accountId": 1650,
      "trashReady": 11,
      "allStreamsTotal": 346,
      "allStreamsReady": 17,
      "ageTotal": 0,
      "accountName": "City Sixteen Fifty",
      "fullnessTotal": 17,
      "trashTotal": 173,
      "singleStreamReady": 6,
      "singleStreamTotal": 173,
      "notReadyTotal": 329,
      "alertTotal": 0
    }
  ],
  "errorCode": "STATUS_OK"
}
```


Collection Summary

Retrieve a summary of collections for a specified time.

Required Parameters:

- *objectType=collectionSummary*
- *action=load*
- *startTime=[Unix Timestamp (in milliseconds)]*
- *endTime=[Unix Timestamp (in milliseconds)]*

Optional Parameters:

- *accountFilter=[CSV list of account ids]*

Returned Results:

FIELD	TYPE	DESCRIPTION
accountId	integer	The CLEAN identifier for the account
accountName	string	The name of the account
daysInPeriod	integer	The total number of days in the period specified
Period	integer	The first day of the interval specified (Unix Timestamp in milliseconds)
componentsInPeriod	integer	The total components collected in the period
Alert	integer	The total number of collections for components with a reason of ALERT
Fullness	integer	The total number of collections for components with a reason of FULLNESS
notReady	integer	The total number of collections for components with a reason of NOT READY
Age	integer	The total number of collections for components with a reason of AGE

Example GET request:

<https://api.bigbelly.com/api/v2?objectType=collectionSummary&action=load&accountFilter=5,1650&endTime=1530373527000&startTime=1530114279000>

Example Results:

```
{
  "errorCode": "STATUS_OK",
  "collectionSummary": [
    {
      "accountId": 5,
      "daysInPeriod": 4,
      "period": 1530158400000,
      "componentsInPeriod": 615,
      "accountName": "City Five",
      "alert": 117,
      "fullness": 575,
      "notReady": 398
    },
    {
      "accountId": 1650,
      "daysInPeriod": 4,
      "period": 1530072000000,
      "componentsInPeriod": 117,
      "accountName": "City Sixteen Fifty",
      "fullness": 91,
      "notReady": 36
    }
  ]
}
```

Assets

Retrieve the individual details for a set of assets

Required Parameters:

- *objectType=assets*
- *action=load*

Optional Parameters:

- *accountFilter=[CSV list of account ids]*
- *serialNumberFilter=[CSV list of account ids]*
- *groupFilter=[CSV list of account ids]*

Returned Results:

FIELD	TYPE	DESCRIPTION
accountId	integer	The CLEAN identifier for the account
accountName	string	The name of the account that owns the asset
latestFullness	string	The current fullness level for the asset
reason	string	The reason that we are saying the asset should be collected (NOT_READY/AGE/FULLNESS/ALERT)
serialNumber	integer	The numeric serial number of the asset. Note: If this asset is not a hub, and its labeled serial number is unknown, this value will usually be set to the stationSerialNumber prefixed with either a 1 or 2 digital number, as needed to make this value an 8 digit number. <ul style="list-style-type: none">- '1' or '10' prefix indicates the asset is at the left position relative to its hub- '2' or '20' prefix indicates the asset is at the right position relative to its hub
latitude	decimal	The latitude for the asset
description	string	The location description for the asset
position	string	The position within the station where the asset resides (<i>left/center/right</i>)
stationSerialNumber	integer	The numeric serial number for the hub this asset is attached to (this will match serialNumber if the asset is a hub)
ageThreshold	integer	The max number of days that are allowed between collections (0 means disabled) that which the asset field reason will change to AGE
fullnessThreshold	string	The fullness level at which the asset field reason will change to FULLNESS, this value should be multiplied by 10 to provide a scale of 0 to 100% of bin capacity
longitude	decimal	The longitude for the asset

NOTE:

- To determine which assets make up a station, the stationSerialNumber field can be used.
- To determine which stations/assets are at the same location use the latitude/longitude fields.

Example GET request:

<https://api.bigbelly.com/api/v2?objectType=assets&action=load&accountFilter=5,1650&serialNumberFilter=1503361,21503361,174864,10174864>

Example Results:

```
{
  "assets": [
    {
      "accountId": 5,
      "latestFullness": 2,
      "reason": "ALERT",
      "serialNumber": 174864,
      "accountName": "City Five",
      "latitude": 38.0244,
      "stationSerialNumber": 174864,
      "description": "4647 N Broad St @ Wyoming, SE",
      "position": "center",
      "ageThreshold": 0,
      "fullnessThreshold": 4,
      "longitude": -72.4177
    },
    {
      "accountId": 5,
      "latestFullness": 0,
      "reason": "NOT_READY",
      "serialNumber": 10174864,
      "accountName": "City Five",
      "latitude": 38.0244,
      "stationSerialNumber": 174864,
      "description": "4647 N Broad St @ Wyoming, SE",
      "position": "left",
      "ageThreshold": 0,
      "fullnessThreshold": 8,
      "longitude": -72.4177
    },
    {
      "accountId": 1650,
      "latestFullness": 0,
      "reason": "NOT_READY",
      "serialNumber": 1503361,
      "accountName": "City Sixteen Fifty",
      "latitude": 40.349210,
      "stationSerialNumber": 1503361,
      "description": "344 Walnut St (Brewer's Coalition)",
      "position": "center",
      "ageThreshold": 0,
      "fullnessThreshold": 6,
      "longitude": -69.206
    },
    {
      "accountId": 1650,
      "latestFullness": 6,
      "reason": "NOT_READY",
      "serialNumber": 21503361,
      "accountName": "City Sixteen Fifty",
      "latitude": 40.349210,
      "stationSerialNumber": 1503361,
      "description": "344 Walnut St (Brewer's Coalition)",
      "position": "right",
      "ageThreshold": 0,
      "fullnessThreshold": 8,
      "longitude": -69.206
    }
  ],
  "errorCode": "STATUS_OK"
}
```

Alerts

Retrieve the set of assets which are currently under alert.

Required Parameters:

- *objectType=alerts*
- *action=load*

Optional Parameters:

- *accountFilter=[CSV list of account ids]*
- *serialNumberFilter=[CSV list of account ids]*
- *groupFilter=[CSV list of account ids]*

Returned Results:

FIELD	TYPE	DESCRIPTION
accountId	integer	The CLEAN identifier for the account
accountName	string	The name of the account
alertType	string	The descriptive name of the alert
serialNumber	integer	The numeric serial number of the asset
Latitude	decimal	The latitude for the asset
stationSerialNumber	integer	The numeric serial number for the parent of the asset
Description	string	The location description for the asset
startTime	long	The time (in UNIX format) that the alert was generated
Position	string	The position within the station where the asset resides (LEFT/CENTER/RIGHT)
alertCategory	string	The level of the alert for the asset (INFORMATIONAL/MINOR/URGENT)
Longitude	decimal	The longitude for the asset

Example GET request:

<https://api.bigbelly.com/api/v2?objectType=alerts&action=load&accountFilter=5&serialNumberFilter=174864>

Example Results:

```
{
  "alerts": [
    {
      "accountId": 5,
      "alertType": "ENCODER_PROBLEM",
      "serialNumber": 174864,
      "accountName": "City Five",
      "latitude": 40.0244,
      "stationSerialNumber": 174864,
      "description": "4647 N Broad St @ Wyoming, SE",
      "startTime": 1530363948000,
      "position": "CENTER",
      "alertCategory": "URGENT",
      "longitude": -75.1477
    }
  ],
  "errorCode": "STATUS_OK"
}
```

Common Terms:

Asset / Component – A single high capacity or standard capacity Bigbelly waste receptacle, it may be a hub or a companion

Hub – A Bigbelly waste receptacle which contains a power source and is capable of communicating with CLEAN servers

Companion – A Bigbelly waste receptacle which is designed to be connected to a Hub, it is not capable of communicating directly with CLEAN servers

Station – A logical group of between one and three components consisting of at least one hub, and up to two companions

Position – Left, Center, Right which refer to the relative positions of components within a Station. The hub of a Station is always Center. Companions may be Left or Right. *Note that this does not necessarily indicate the physical position of components at a collection point, only the position of companions relative to hubs.*

Location / Collection Point – A geographic point which containing one or more Stations. The Stations may be physically connected together to appear as one kiosk

Ready to Collect – The indicator that a component needs to be collected. The reason the component is ready may be due to its fullness, age since last transition, or the component being under alert which may prevent its fullness from being accurate.

Alert – An issue that may be impairing the correct operation of a component. Alerts are split into categories intended to express their urgency: Urgent, Minor, Informational.

Group – A logical association of various components to one another within CLEAN.