Requests

BigML.io uses the standard POST, GET, PUT, and DELETE HTTP methods to create, retrieve, update, and delete individual resources, respectively. You can also list all your resources for each resource type.

Jump to:

- Creating a Resource
- Retrieving a Resource
- Updating a Resource
- Deleting a Resource
- Listing Resources
- Paginating Resources
- Filtering Resources
- Ordering Resources
- Webhooks

Creating a Resource

To create a new resource, you need to POST an object to the resource's base URL. The content-type must always be **"application/json"**. The only exception is **source** creation which requires the "multipart/form-data" content type.

For example, to create a model with a dataset, you can use curl like this:

```curl
curl "https://bigml.io/andromeda/model/?$BIGML_AUTH" \
  -X POST \
  -H 'content-type: application/json' \
  -d '{"dataset": "dataset/4f66a80803ce8940c5000006"}'
```

The following is an example of what a request header would look like for the request:

```bash
POST /model?username=alfred&api_key=79138a622755a2383660347f895444b1eb927730 HTTP/1.1
```

```
Host: bigml.io
Content-Type: application/json
```

BigML.io will return a newly create resource document if the request is succeeded.

A number of required and optional arguments exist for each type of resource. You can see a detailed arguments list for each resource in their respective sections: projects, external connectors, sources, datasets, and etc.

Retrieving a Resource

To retrieve a resource, you need to issue a HTTP GET request to the **resource/id** to be retrieved. Each resource has a unique identifier in the form **resource/id** where resource is a type of the resource such as **dataset**, **model**, and etc, and id is a string of 24 alpha-numeric characters that you can use to retrieve the resource or as a parameter to create other resources from the resource.

For example, using **curl** you can do something like this to retrieve a **dataset**:

```curl
curl "https://bigml.io/andromeda/dataset/54d86680f0a5ea5fc0000011?$BIGML_AUTH"
```

The following is an example of what a request header would look like for a **dataset** GET request:

```bash
GET /dataset/54d86680f0a5ea5fc0000011?username=alfred&api_key=79138a622755a2383660347f89544
Host: bigml.io
```

Once a resource has been successfully created, it will have properties. A number of properties exist for each type of resource. You can see a detailed property list for each resource in their respective sections: projects, external connectors, sources, datasets, and etc.

Updating a Resource

To update a resource, you need to PUT an object containing the fields that you want to update to the resource's base URL. The content-type must always be: **"application/json"**.

If the request succeeds, **BigML.io** will respond with a **202 accepted** code and with the new updated resource in the body of the message.

For example, to update a **project** with a new **name**, a new **category**, a new **description**, and new **tags** you can use **curl** like this:

```curl
curl "https://bigml.io/andromeda/project/54d9553bf0a5ea5fc0000016?$BIGML_AUTH" \
  -X PUT \
  -H 'content-type: application/json' \
  -d '{"name": "My new Project",
       "category": 3,
       "description": "My first BigML Project",
       "tags": ["fraud", "detection"]}'
```

The following is an example of what a request header would look like for the request:

```bash
PUT /project/54d9553bf0a5ea5fc0000016?username=alfred&api_key=79138a622755a2383660347f89544
Host: bigml.io
Content-Type: application/json
```

Deleting a Resource

To delete a resource, you need to issue a HTTP DELETE request to the **resource/id** to be deleted.

For example, using curl you can do something like this to delete a **dataset**:

```curl
curl -X DELETE "https://bigml.io/andromeda/dataset/54d86680f0a5ea5fc0000011?$BIGML_AUTH"
```

If the request succeeds you will not see anything on the command line unless you executed the command in verbose mode. Successful DELETEs will return HTTP 204 responses with no body.

```
HTTP/1.1 204 NO CONTENT
Content-Length: 0
```

Once you delete a resource, it is permanently deleted. That is, a delete request cannot be undone.

For example, if you try to delete a **dataset** a second time, or a **dataset** that does not exist you will receive an error like this:

json

```json
{
  "code": 404,
  "status": {
      "code": -1201,
      "extra": [
          "A dataset matching the provided arguments could not be found"
      ],
      "message": "Id does not exist"
  }
}
```

The following is an example of what a request header would look like for a **dataset** DELETE request:

bash

```bash
DELETE /dataset/54d86680f0a5ea5fc0000011?username=alfred&api_key=79138a622755a2383660347f895
Host: bigml.io
```

Listing Resources

To list all the resources, you can use its base URLs. By default, only the 20 most recent resources will be returned. You can see below how to change this number using the **limit** parameter.

You can get the list of each resource type directly in your browser using your own **username** and **API key** with the following links.

bash

```bash
https://bigml.io/andromeda/project?$BIGML_AUTH
https://bigml.io/andromeda/externalconnector?$BIGML_AUTH
https://bigml.io/andromeda/source?$BIGML_AUTH
https://bigml.io/andromeda/dataset?$BIGML_AUTH
https://bigml.io/andromeda/sample?$BIGML_AUTH
https://bigml.io/andromeda/correlation?$BIGML_AUTH
https://bigml.io/andromeda/statisticaltest?$BIGML_AUTH
```

```
https://bigml.io/andromeda/configuration?$BIGML_AUTH
https://bigml.io/andromeda/composite?$BIGML_AUTH
```

You can also easily list them from the command line using **curl** as follows:

curl

```
curl "https://bigml.io/andromeda/project?$BIGML_AUTH"
curl "https://bigml.io/andromeda/externalconnector?$BIGML_AUTH"
curl "https://bigml.io/andromeda/source?$BIGML_AUTH"
curl "https://bigml.io/andromeda/dataset?$BIGML_AUTH"
curl "https://bigml.io/andromeda/sample?$BIGML_AUTH"
curl "https://bigml.io/andromeda/correlation?$BIGML_AUTH"
curl "https://bigml.io/andromeda/statisticaltest?$BIGML_AUTH"
curl "https://bigml.io/andromeda/configuration?$BIGML_AUTH"
curl "https://bigml.io/andromeda/composite?$BIGML_AUTH"
```

The following is an example of what a request header would look like when you request a list of **models**:

```
GET /model?username=alfred&api_key=79138a622755a2383660347f895444b1eb927730
Host: bigml.io
```

When you list resources, the JSON response that you get back is a meta object and a list of resource objects. The meta object paginates all the resources returned in the response.

List Response Object Properties

Property	Type	Description
meta	Object	Specifies in which page of the listing you are, how to get to the previous page and next page, and the total number of resources.
objects	Array of resources	A list of resources filtered and ordered according to the criteria that you supply in your request. See the filtering and ordering options for more details.

Meta objects have the following properties:

Meta Object Properties

Property	Type	Description
limit	Integer	Maximum number of resources that you will get in the objects field.
next	String	Path to get the next page or null if there is no next page.
offset	Integer	How far off from the first resource in your resources is the first resources in the objects field.
previous	String	Path to get the previous page or null if there is no previous page.
total_count	Integer	Total number of resources.

For example, when you list your **projects**, they will be displayed as below:

```json
{
  "meta": {
    "limit": 20,
    "next": "/?username=alfred&api_key=79138a622755a2383660347f895444b1eb927730&offset=20"
    "offset": 0,
    "previous": null,
    "total_count": 54
  },
  "objects": [
    {
      "category": 0,
      "code": 200,
      "created": "2015-01-27T22:51:57.488000",
      "description": "",
```

Paginating Resources

There are two parameters that can help you retrieve just a portion of your resources and paginate them.

Pagination Parameters

Parameter	Type	Description
limit optional	Integer, default is **20**	Specifies the number of resources to retrieve. Must be less than or equal to 200.
offset optional	Integer, default is **0**	The order number from which the resource listing will start.

If a **limit** is given, no more than that many resources will be returned but possibly less, if the request itself yields less resources.

For example, if you want to retrieve only the third and forth latest **projects**:

curl

```
curl "https://bigml.io/andromeda/project?$BIGML_AUTH&limit=2&offset=2"
```

To paginate results, you need to start off with an **offset** of zero, then increment it by whatever value you use for the **limit** each time. So if you wanted to return resources 1-10, then 11-20, then 21-30, etc., you would use "**limit**=10&**offset**=0", "**limit**=10&**offset**=10", and **limit**=10&**offset**=20", respectively.

Filtering Resources

The listings of resources can be filtered by any of the fields that we labeled as **filterable** in the table describing the properties of a resource type. For example, to retrieve all the **projects** tagged with "fraud":

bash

```
https://bigml.io/andromeda/project?$BIGML_AUTH&tags__in=fraud
```

Using **curl**:

curl

```
curl "https://bigml.io/andromeda/project?$BIGML_AUTH&tags__in=fraud"
```

In addition to exact match, there are more filters that you can use. To add one of these filters to your request you just need to append one of the suffixes in the following table to the name of the property that you want to use as a filter.

Filters

Filter	Description
! optional	Not **Example**: !size=1048576 (<>1MB)
__gt optional	Greater than **Example**: size__gt=1048576 (>1MB)
__gte optional	Greater than or equal to **Example**: size__gte=1048576 (>=1MB)
__contains optional	Case-sensitive word match **Example**: name__contains=test
__icontains optional	Case-insensitive word match **Example**: name__icontains=test
__startswith optional	Case-sensitive startswith match **Example**: name__contains=test
__istartswith optional	Case-insensitive startswith match **Example**: name__icontains=test
__in optional	Case-sensitive list word match **Example**: tags__in=fraud,test
__lt optional	Less than **Example**: created__lt=2022-03-20T00:00:00.000000 (before 2022-03-20)
__lte optional	Less than or equal to **Example**: created__lte=2022-03-20T00:00:00.000000 (before or on 2022-03-20)

Ordering Resources

A list of resources can also be ordered by any of the fields that we labeled as **sortable** in the table describing the properties of a resource type.

Ordering Parameters

Parameter	Type	Description
order_by optional	Name of a sortable field, default is **"-created"**	Specifies the order of the **resources** to retrieve. Must be one of the sortable fields. If you prefix the field name with "-", they will be given in descending order.

For example, you can list your **projects** ordered by descending **name** directly in your browser, using your own **username** and **API key**, with the following link.

```bash
https://bigml.io/andromeda/project?$BIGML_AUTH&order_by=-name
```

You can do the same thing from the command line using **curl** as follows:

```curl
curl "https://bigml.io/andromeda/project?$BIGML_AUTH&order_by=-name"
```

Webhooks

Webhooks allow you to build or set up apps which subscribe to the events triggered when the resource creation is complete or halted with an error. When the **finished** or **error** event is triggered, **BigML.io** can send an HTTP POST payload to the webhook's configured URL.

When you create a resource, you can specify the webhook parameter in the POST payload. For example, to create a **model** with a **dataset**, you can use **curl** like this:

```curl
curl "https://bigml.io/andromeda/model/?$BIGML_AUTH" \
  -X POST \
  -H 'content-type: application/json' \
  -d '{"dataset": "dataset/4f66a80803ce8940c5000006",
       "webhook":{"url": "http://myhost/path/to/webhook"}}'
```

When the resource creation is complete, **BigML.io** calls the provided URL and send an HTTP POST payload, and expects to receive an HTTP 201 status code. Optionally, you can provide the **secret** parameter to secure your webhook. The value of **secret** will be used as the key to generate the HMAC hex digest value of the request body in the **X-BigML-Signature** header if provided. It uses the **sha1** hash function and the value will always have the prefix of **sha1=**. The headers also contain **X-BigML-Delivery**, a GUID to identify the delivery, and **User-Agent**, which is **BigML.io**. The payload of the POST request is in the JSON format, so be sure to accept **Content-Type: application/json**.

```json
"webhook": {
    "url": "http://myhost/path/to/webhook",
    "secret": "mysecret"
}
```

The following is an example of a POST request to the webhook server. Note that the headers contain **X-BigML-Signature** when **secret** is provided.

```bash
POST /path/to/webhook HTTP/1.1
Host: localhost:800
X-BigML-Delivery: dd04ace6-c2c7-4c62-afff-d6514c016ad7
X-BigML-Signature: sha1=b7f0e0b9401f85ab00c8c8c575a5d71006788eec
User-Agent: BigML.io
Content-Type: application/json;charset=utf-8
Content-Length: 162
{
    "event": "finished",
    "message": "The model has been created",
    "resource": "model/5ba2ccc54e172745a0000000",
    "timestamp": "2018-09-19 22:25:11 GMT"
}
```

When **secret** is provided, it is convenient to validate the webhook call authenticity comparing received sha1 in the **X-BigML-Signature** header with a new generate HMAC hex digest value of the POST request body using same **secret** value sent in the creation request. In order to do this validation, you must first format the json payload of the POST request to remove line breaks and other special characters.

The following is an example of the json to use to generate the hash. Mind that the keys in the JSON string must be alphabetically sorted.

```json
{ "event": "finished", "message": "The model has been created", "resource": "model/5ba2ccc5
```

You have an example of the implementation in Python in the function that validates this in our bindings.

The following is an example of the **webhook** property in the response body of a model resource.

```json
"webhook": {
    "delivery": {
      "confirmation_id": "dd04ace6-c2c7-4c62-afff-d6514c016ad7",
      "method": "queue",
      "status": "delivered"
    },
    "event": "finished",
    "secret": "mysecret",
    "signature": "sha1=b7f0e0b9401f85ab00c8c8c575a5d71006788eec",
    "timestamp": "2018-09-19T22:25:11.536000",
    "url": "http://myhost/path/to/webhook"
}
```