# DataGator Specification: RESTful API v2*

| | |
|---|---|
| **author** | LIU Yu <liuyu@opencps.net> |
| **revision** | draft-05 |
| **license** | CC BY-NC-ND 4.0 |

## 1. Introduction

The RESTful API of DataGator is a JSON-based programming interface for accessing and manipulating DataGator's computing infrastructure. This document specifies web service endpoints and protocols for invoking the RESTful API of DataGator. Targeted readers of this document are developers experienced in web programming, esp., consuming web services through HTTP messages as specified in RFC 7231.

### Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

## 2. Overview

*API version*

This document describes the v2 version of DataGator's RESTful API. All API calls are over HTTPS. Service endpoints are absolute or relative URI templates as defined in RFC 6570. Unless otherwise specified, a relative service endpoint resolves to,

```
https://www.data-gator.com/api/v2{endpoint}
```

*JSON schema*

Data sent and received through API calls are HTTP messages with JSON objects as payload, which all conform to the draft-4 JSON schema available at,

```
https://www.data-gator.com/api/v2/schema#
```

For example, sending a **GET** request to the *root endpoint* (/) will receive an HTTP response with a *Message* object as its payload, e.g.,

```
GET /api/v2/ HTTP/1.1
Host: www.data-gator.com
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Wed, 03 Jun 2015 14:13:58 GMT
X-DataGator-Entity: Status
X-RateLimit-Limit: 200
X-RateLimit-Remaining: 188
X-RateLimit-Reset: 1433342782

{
    "kind": "datagator#Status",
    "code": 200,
    "version": "v2",
    "service": "datagator.wsgi.api"
}
```

*HTTP authentication*

Most service endpoints optionally perform client authentication and respond with personalized information matching each client's privileges. Some other resources are dedicated to authenticated clients with matching permissions[1]. Clients are RECOMMENDED to preemptively send the `Authorization` HTTP header when making API calls. The `v2` API supports the following two HTTP authentication schemes.

**Basic authentication**: Per RFC 7617, the client credentials sent with the HTTP request is the concatenation of the `username`, a single colon (`":"`) character, and the `password`, encoded into a string of ASCII characters using Base64.

```
GET /api/v2/ HTTP/1.1
Host: www.data-gator.com
Authorization: Basic {credentials}
```

**Token authentication**: The `access_token` sent with the HTTP request is an opaque string of ASCII characters issued to the client.

```
GET /api/v2/ HTTP/1.1
Host: www.data-gator.com
Authorization: Token {access_token}
```

*HTTP redirection*

API uses HTTP redirections where appropriate. Receiving an HTTP redirection is *not* an error, and clients SHOULD follow the redirection by default. Redirect responses will have a `Location` header containing the URI of the targeted resource.

*pagination*

Some *listing* service endpoints return a *paginated list* of entities encapsulated in a *Page* object. HTTP **`GET`** requests to these services can take an optional `?page` parameter in the query string to specify the *zero*-based page number of interest. A *Page* object contains 10 to 20 items by default. For some resources, the size of a *Page* object can be customized to contain up to 100 items with a `?page_size` parameter.

```
GET /api/v2/repo/Pardee/IGOs/data/?page=2&page_size=30 HTTP/1.1
Host: www.data-gator.com
Accept: */*
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Fri, 04 Sep 2015 06:16:41 GMT
Link:
  </api/v2/repo/Pardee/IGOs.1/data?page=0&page_size=30>; rel="first",
  </api/v2/repo/Pardee/IGOs.1/data?page=1&page_size=30>; rel="prev",
  </api/v2/repo/Pardee/IGOs.1/data?page=3&page_size=30>; rel="next",
```

[1] To prevent accidental leakage of private information, some service endpoints will return `404 Not Found`, instead of `403 Forbidden`, to unauthorized clients.

```
  </api/v2/repo/Pardee/IGOs.1/data?page=13&page_size=30>; rel="last"
X-RateLimit-Limit: 200
X-RateLimit-Remaining: 198
X-RateLimit-Reset: 1441350986

{
    "kind": "datagator#Page",
    "items": [
        {"kind": "datagator#Matrix", "name": "ASEF"},
        {"kind": "datagator#Matrix", "name": "ASPAC"},
        ...
        {"kind": "datagator#Matrix", "name": "CBI"}
    ],
    "startIndex": 60,
    "itemsPerPage": 30,
    "itemsCount": 20
}
```

Service endpoints that return *Page* objects MAY also provide RFC 5988 `Link` headers containing one or more of the following link relations.

| Relation | Description |
|---|---|
| `first` | link to the initial *Page* |
| `prev` | link to the immediate previous *Page* |
| `next` | link to the immediate next *Page* |
| `last` | link to the last non-empty *Page* |

When enumerating a *paginated* resource, clients are recommended to follow the `Link` relations instead of constructing URIs by themselves. Note that, the *pagination* of resource is open-ended. Querying a `?page` number beyond the `last` page is *not* an error, and will receive an empty *Page* object, instead of `404 Not Found`.

*rate limiting*

Authenticated clients can make up to 2,000 API calls per hour. For unauthorized clients, the limit is 200 calls per hour and is associated with the client's' IP address. The rate limit status is included in the `X-RateLimit-*` headers of HTTP responses.

| HTTP Header | Description |
|---|---|
| `X-RateLimit-Limit` | The hourly limit of API calls allowed for the current client. |
| `X-RateLimit-Remaining` | The number of API calls remaining in current rate limit window. |
| `X-RateLimit-Reset` | The UNIX time at which the current rate limit window resets. |

Exceeding the rate limit will receive a `429 Too Many Requests` response.

```
HTTP/1.1 429 TOO MANY REQUESTS
Content-Type: application/json
Date: Fri, 05 Sep 2015 03:10:56 GMT
X-DataGator-Entity: Error
X-RateLimit-Limit: 200
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 1441426202
Content-Length: 110
Retry-After: 3546

{
    "kind": "datagator#Error",
```

```
    "code": 429,
    "service": "datagator.rest.api",
    "message": "API request over-rate."
}
```

Note that, invoking some *expensive* services, such as *full-text search* and *dataset revision*, may be counted as multiple API calls.

*CORS*

The `v2` API accepts client-side requests from any origin.

```
GET /api/v2/ HTTP/1.1
Host: www.data-gator.com
Origin: http://example.com
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Date: Fri, 16 Oct 2015 13:25:57 GMT
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, HEAD, POST, PUT, PATCH, DELETE
Access-Control-Allow-Origin: http://example.com
Access-Control-Expose-Headers: ETag, Last-Modified, Link, Location,
    X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset,
    X-DataGator-Entity
X-DataGator-Entity: Status
Vary: Origin
```

**Know Issues**: Due to a known issue of the web server being used by the backend system, `304 Not Modified` responses do *not* currently contain CORS headers.