Introduction to APIs

REST and GraphQL

Module Overview

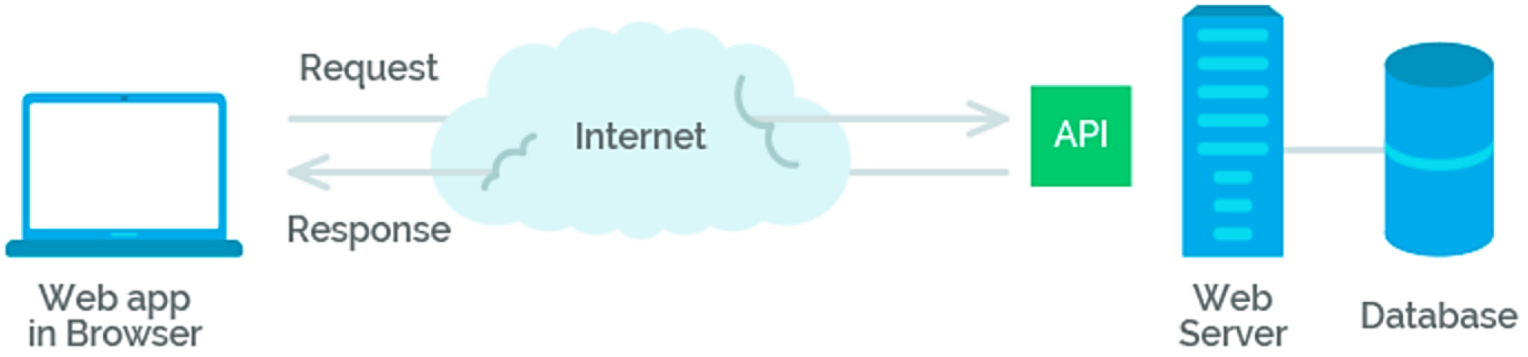
This module will cover the following:

- Getting Started with REST APIs
- REST Requests
- REST Responses
- REST Authentication
- GraphQL

Getting Started with REST APIs

What is an API?

- APIs (application programming interface) are a set of rules and mechanisms by which one application or component interacts with the others



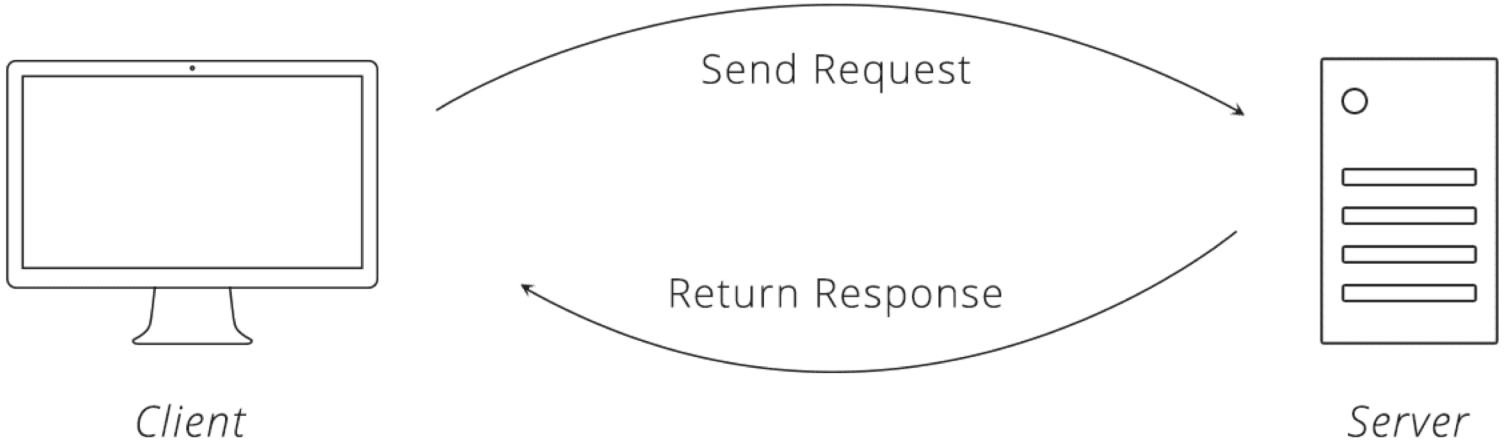
What is REST?

REST stands for **Representational State Transfer**

- Data presentation for a client in the format that is convenient for it to consume.
- REST is not a standard or protocol
  - An approach to writing an API.
  - Architectural style for writing an API.

REST Communication

- REST communication centers around a concept called the **Request-Response Cycle**.



API Documentation

- Real-time visibility into all available endpoints
- Includes parameter, query, path, and response details
- Professionally written and maintained with pseudo-code and code snippet examples
- Explains the **why** and the **how**

SLA Domains

Rubrik clusters provide automated data management and protection through SLA Domains. An SLA Domain defines the c
protection policies for their assigned snappables (virtual machines, file systems, and applications).

To provide policy based management and protection of a snappable, add the snappable to an SLA Domain, or to multiple

Retrieving SLA Domains

Before assigning snappables to SLA Domains, get a list of the SLA Domains that exist on a Rubrik cluster. For a new Rubr
only the default SLA Domains. When custom SLA Domains are added to the Rubrik cluster, the list is modified to include t

Example: Retrieving SLA Domains from a Rubrik cluster

Send a GET request to `/sla_domain`.

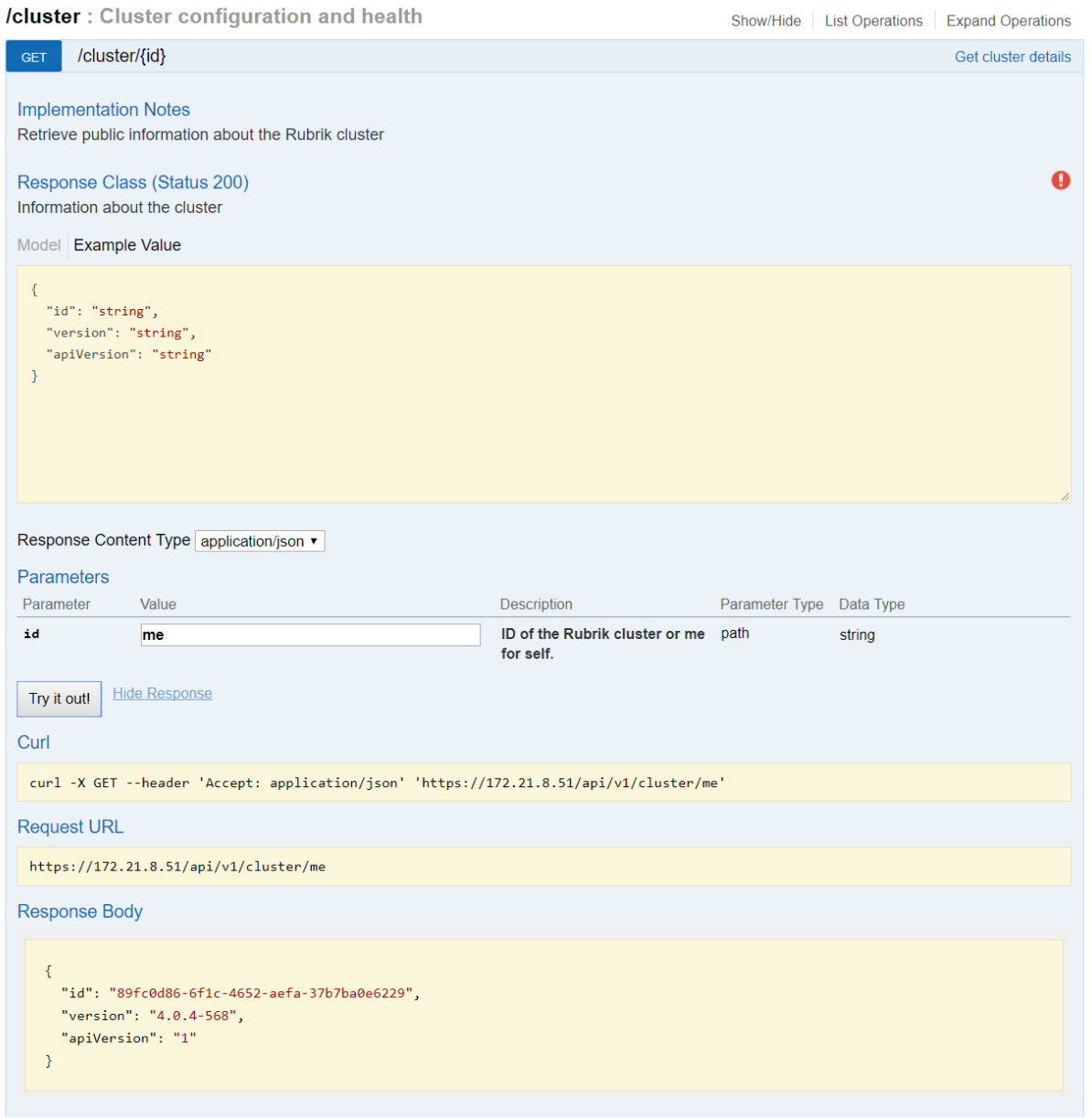
```
curl -X GET "https://$cluster_address/api/v1/sla_domain"
```

The Rubrik REST API server returns a `ListResponse` object of all SLA Domains. At a minimum, the `ListResponse`
SLA Domains: Gold, Silver, and Bronze.

```
{
  "data": [
    {
      "id": "$gold_sla_id",
      "name": "Gold"
    },
```

API Explorer

- Live access to API through industry standard interface familiar to developers
- Documents all available REST endpoints
- Facilitates quick integration development



cURL

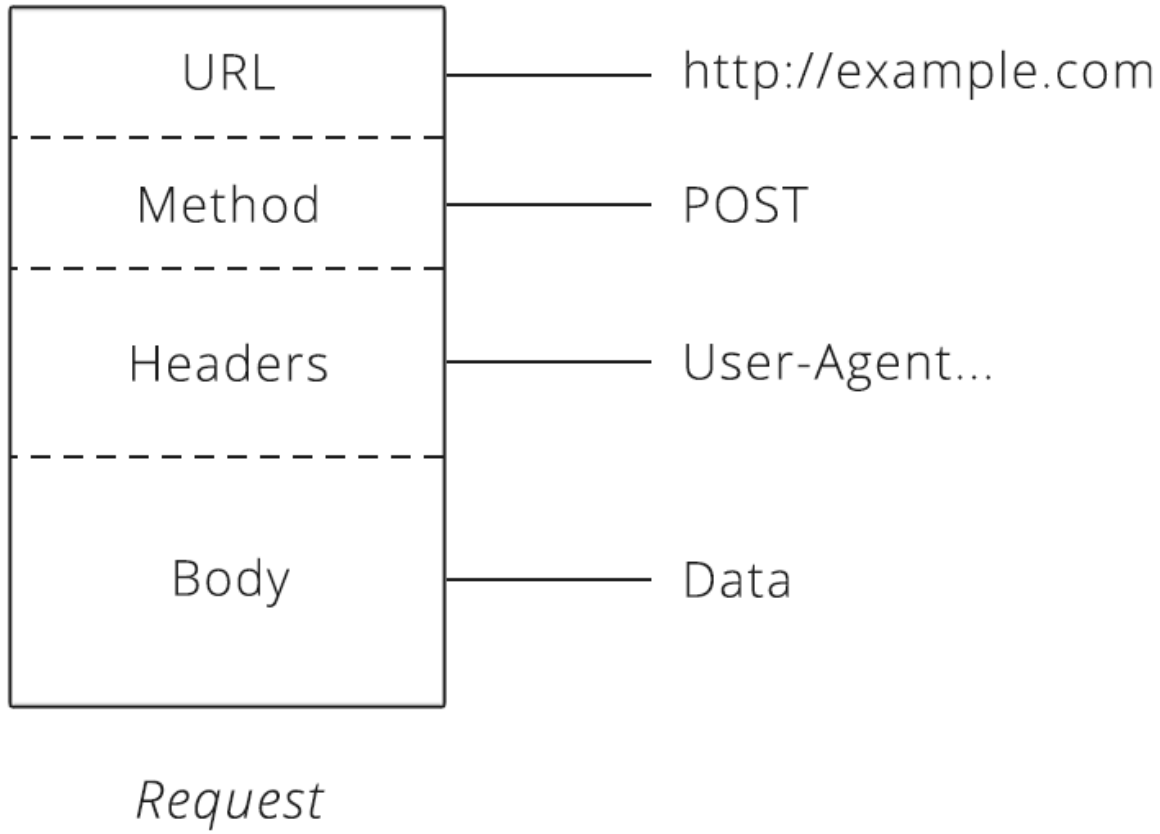
- cURL is an open source command line utility called cURL
- Most API documentation is written with reference to cURL, for example:

```
curl -k -u admin:pass -X GET
"https://$cluster_address/api/v1/cluster/me"
```

REST Requests

Anatomy of a Request

- A request is made up of four things:
  - Endpoint
  - Method
  - Headers
  - Body (or data)



Endpoint

- **The endpoint** (or route) is the URL you request for. It follows this structure:
  - `root-endpoint/path`
- The **root-endpoint** is the starting point of the API you're requesting
  - `https://api.github.com`
  - `https://$Rubrik_cluster_IP/api/`
- The **path** determines the resource you are requesting
  - `https://api.github.com/orgs/{org}`
  - `https://$Rubrik_cluster_IP/api/v1/cluster/{id}/api_version`

Method

REST relies heavily on HTTP.

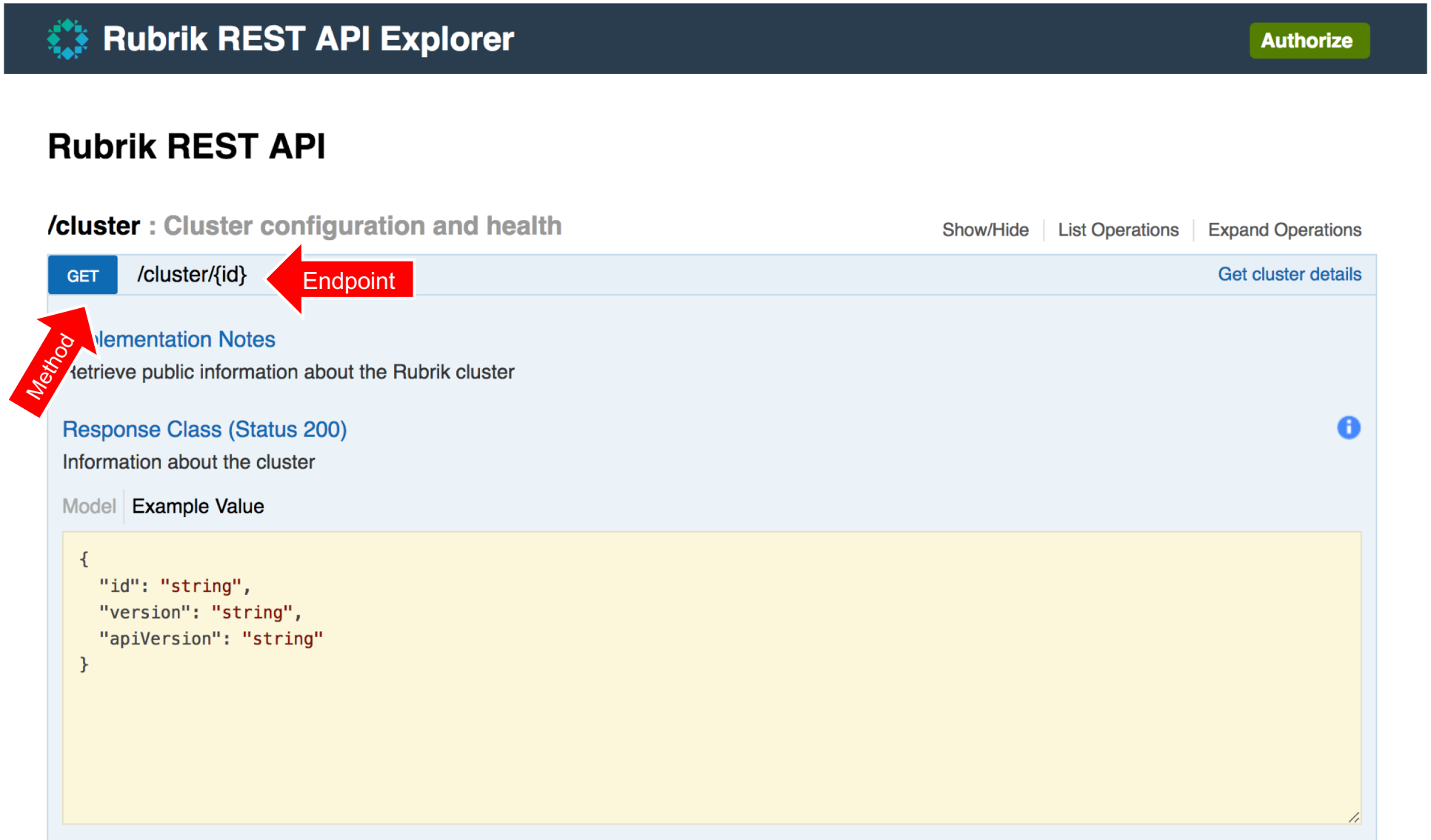
Each operation uses its own HTTP method:

- GET – getting
- POST – creation
- PUT – full object update (modification)
- PATCH – partial object update (modification)
- DELETE – removal

Headers and Body

- **Headers** provide meta-information about a request
- The request **body** contains the data the client wants to send the server

Request Example

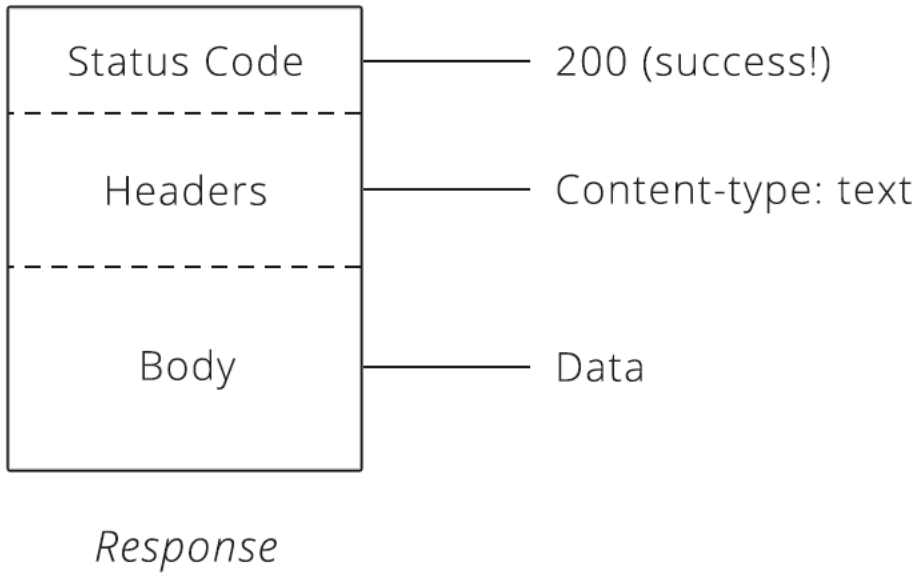


REST Response

Anatomy of a Response

APIs respond with:

- Status Code
- Headers
- Body or Data

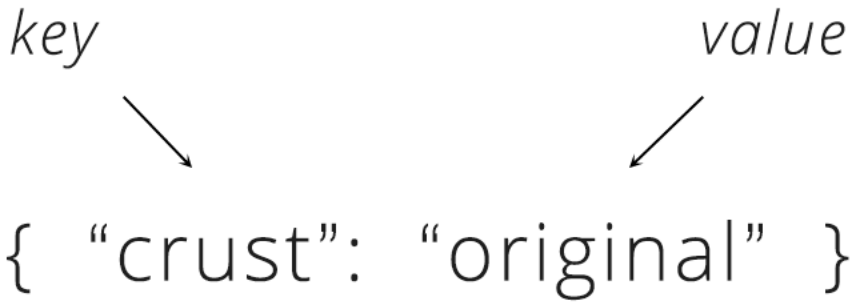


HTTP Status

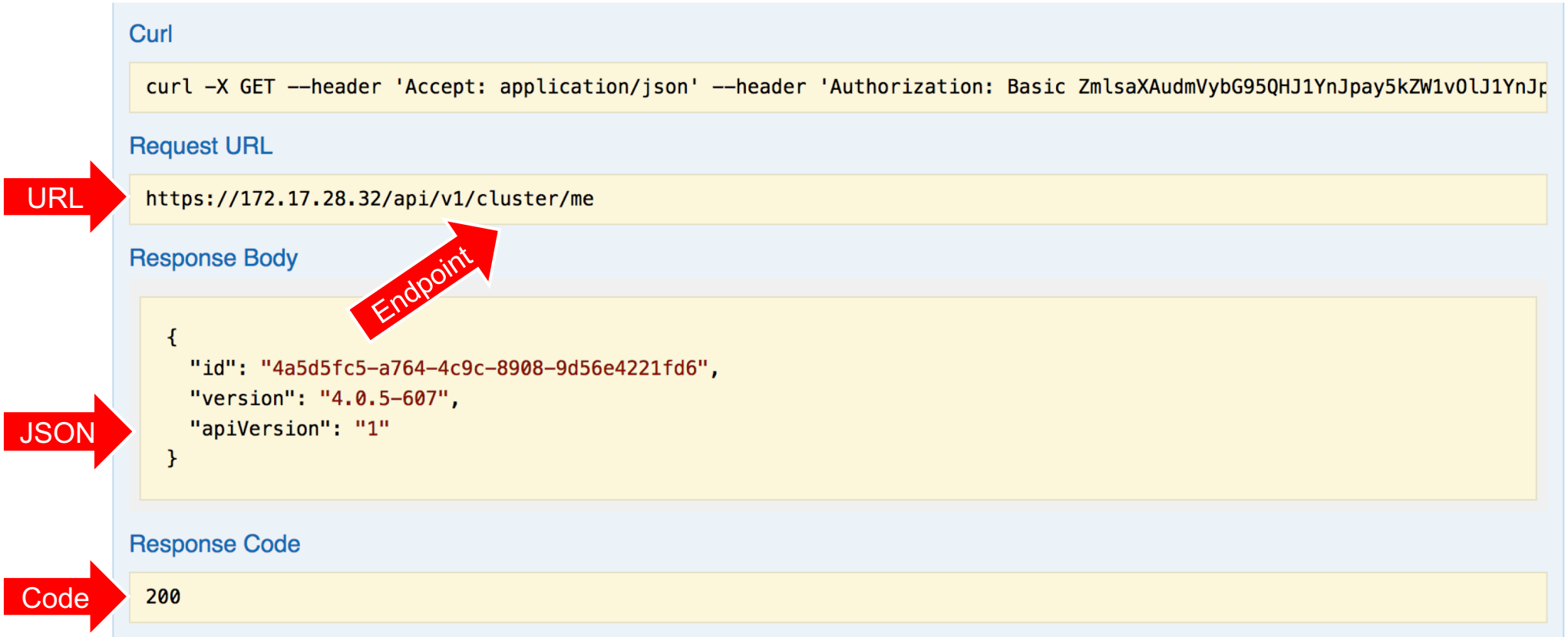
- 1xx - Informational
- 2xx - Success
- 3xx - Redirection (client needs to take additional action)
- 4xx - Client error
- 5xx - Server error

Data Formats

- Data is returned in a conveniently formatted header and body using JSON or XML



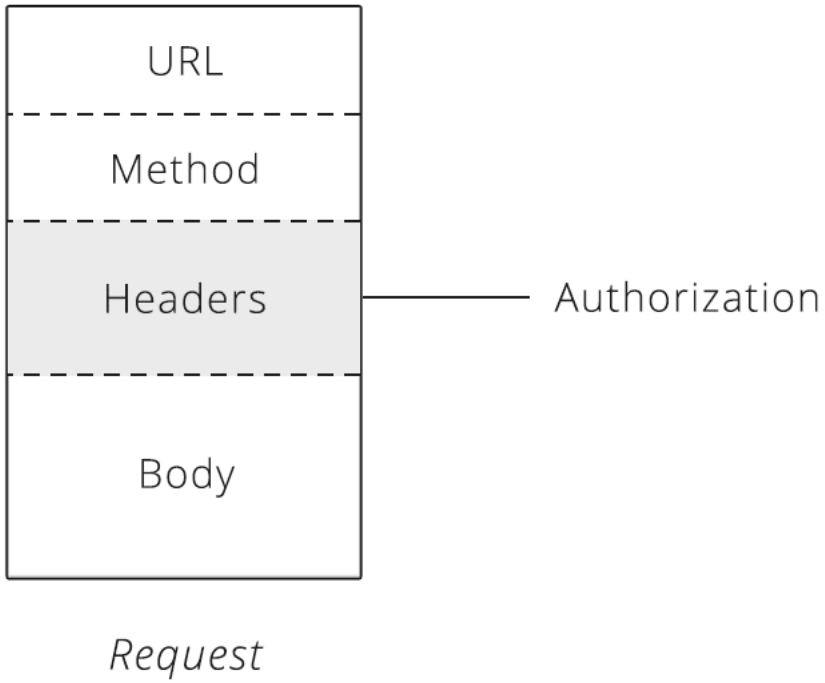
Response Example



REST Authentication

Anatomy of Authentication

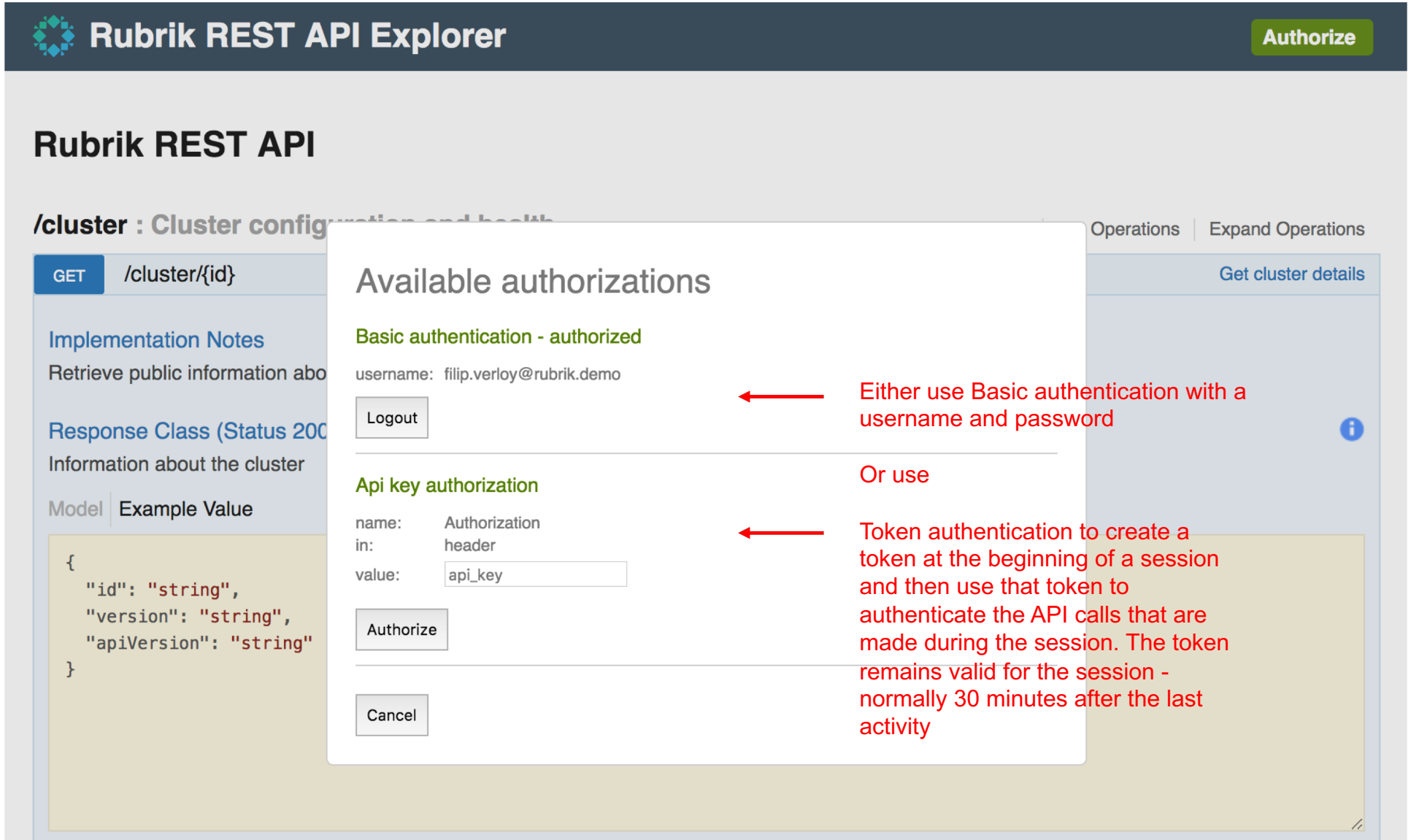
- There are two main ways to authenticate:
  - Basic authentication (username and password)
  - Token-based
    - Includes oAuth, which lets you to authentication with social media networks such as GitHub, Google, Twitter, Facebook, etc.



Authenticating with cURL

- To perform a basic authentication with cURL, you can use the `-u` option, followed by your username and password. For example:
- `curl -x POST -u "username:password" https://api.github.com/user/repos`

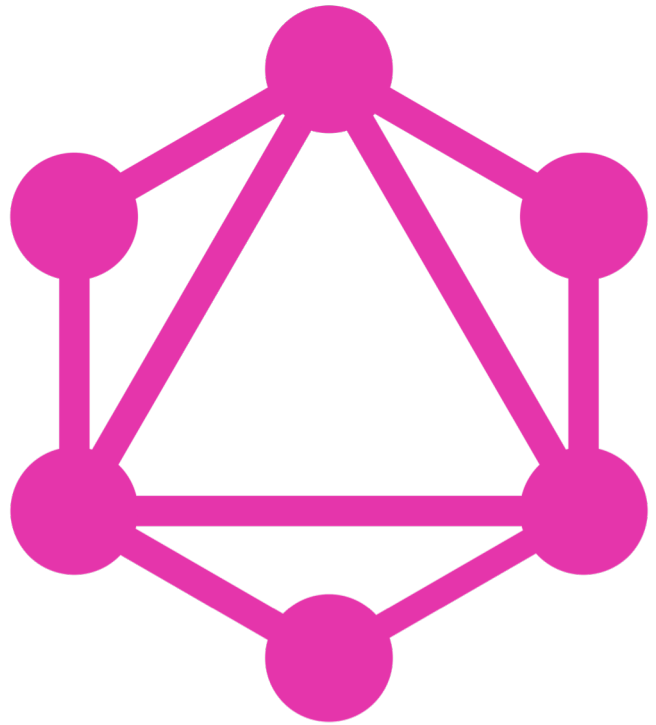
Authenticating using API Explorer



GraphQL

Why GraphQL?

- GraphQL is an API query and manipulation language built by Facebook in 2012, and publicly released in 2015.
- It uses HTTP as its transport method and leverages POST calls to provide increased efficiency around GET requests to REST APIs.
  - Request information you want via payload
  - Receive only the information you requested back



GraphQL Advantages

- Advantages of GraphQL over REST API:
  - Complex queries return spuriously-related properties of objects without having to issue multiple requests
  - Return only the data needed, so response times and processing are reduced

build

Building the Future of Data Management