CYBRAICS

nLighten™

nLighten 4 API Guide

CYBRAICS

Overview

The nLighten Application Programming Interface (API) is an interface to share nLighten data with other applications. It is implemented with REST and GraphQL and currently returns JavaScript Object Notation (JSON) objects that represent the requested data from nLighten. This API is protected using API Keys and token security model and IP address whitelisting. This document gives a developer who wants to access the API, the information needed to complete this task.

Assumptions

This document assumes that the user has already been given an nLighten account to access the desired instance, has activated this account, and has a whitelisted IP address to access the portal. If an account has not been created for nLighten access, it should be completed prior to using this document.

Access Token Authentication

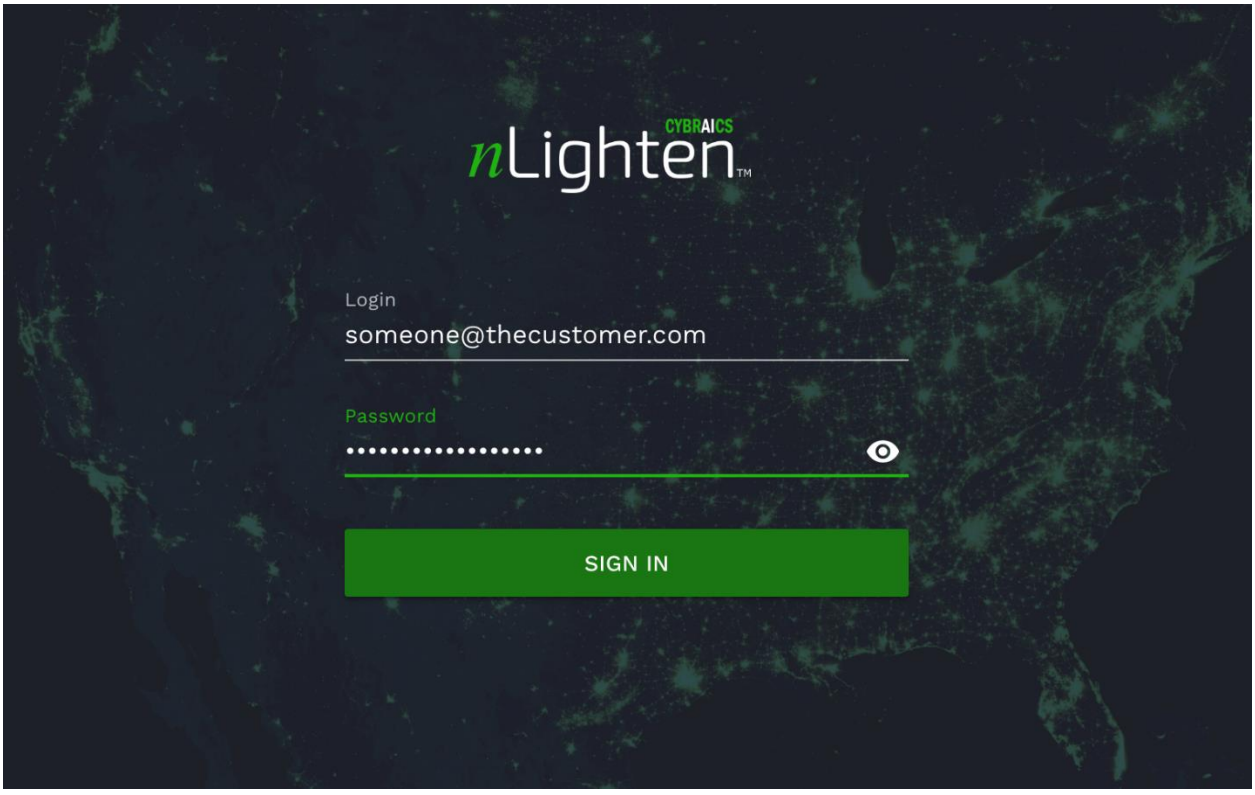
HTTP requests made to nLighten must be authenticated or the response will be 401 Unauthorized. If the request is authenticated but the requestor doesn't have the required permissions, the response will be 403 Forbidden.

To authenticate, the requestor must supply an access token in an Authorization HTTP header. Below is an example. The long string of characters is the access token.

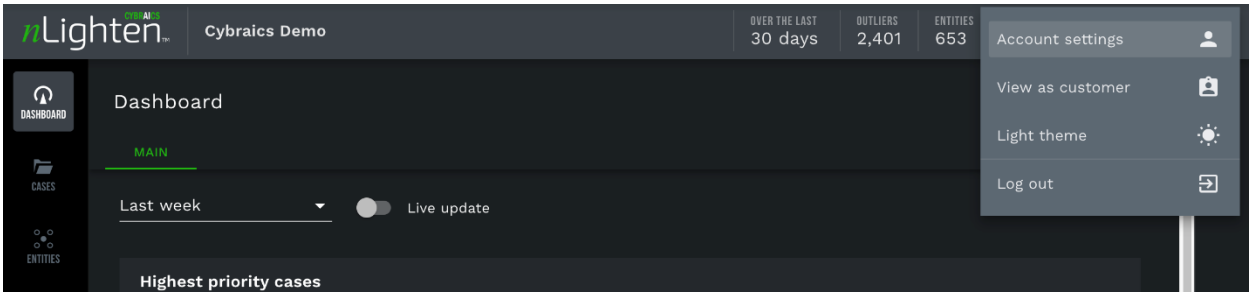
```
{ "Authorization": "Bearer 123456789012345678" }
```

You can create an access token for an instance of nLighten by following steps below.

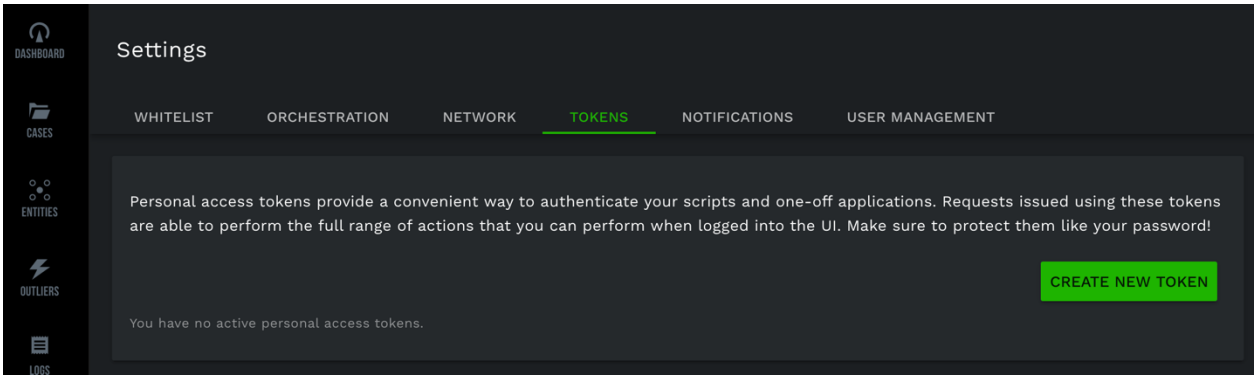
1. Sign into the nLighten instance (`https://<pea>.nl.cybraics.com`).



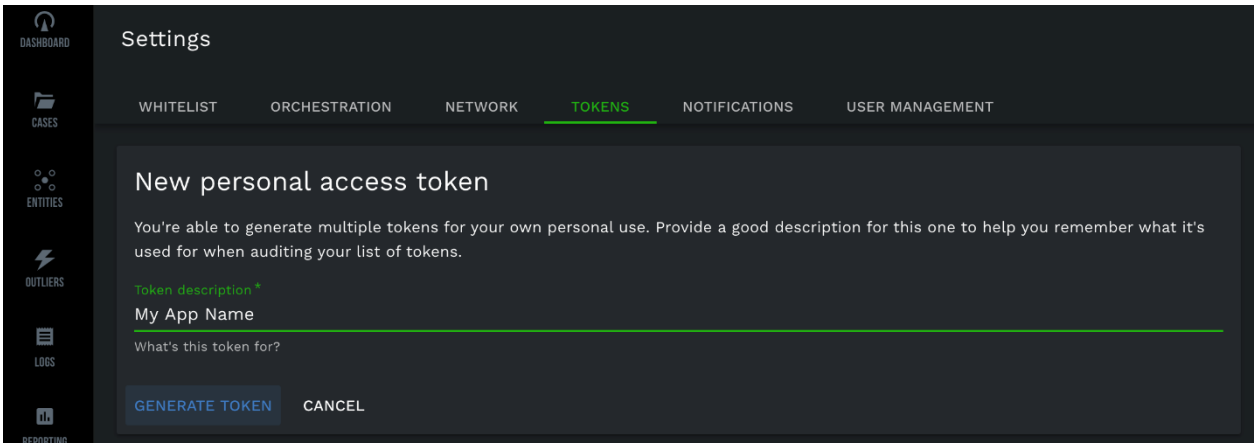
2. At the top right of nLighten, click on your name to get a dropdown menu, then click on "Account Settings".



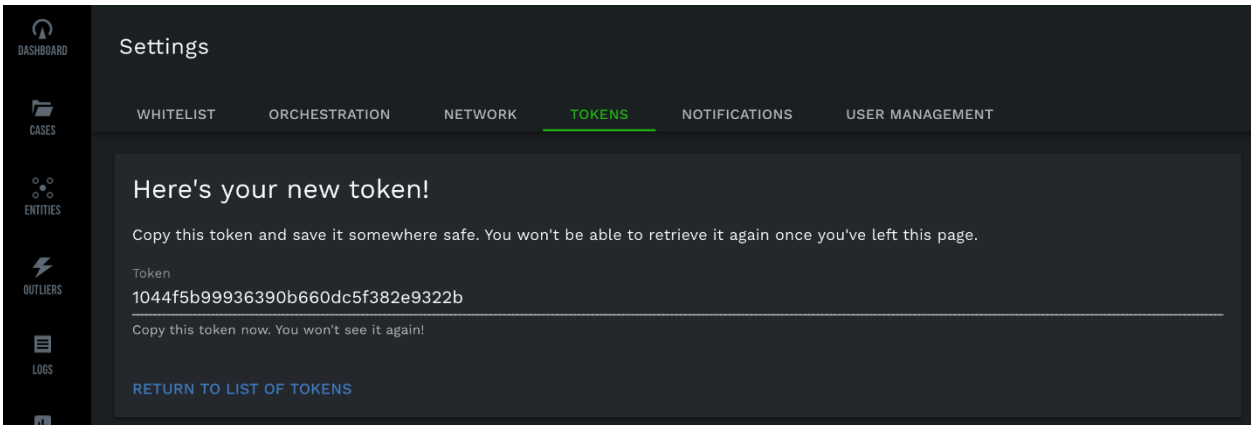
3. On the "Settings" page click on the "TOKENS" tab.



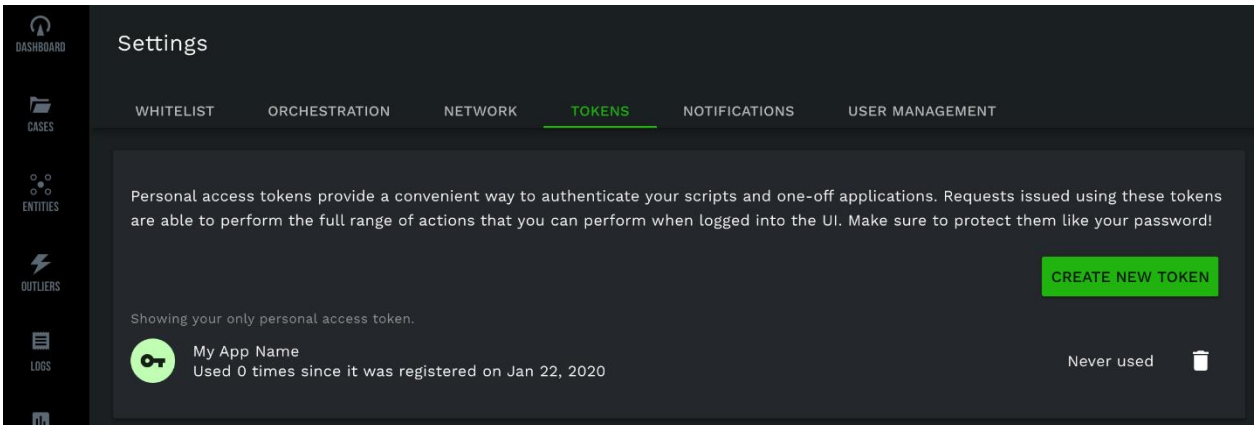
4. Under the "TOKENS" tab click on the "CREATE NEW TOKEN" button.
   Add a token description. A useful description is the name of the application that will use the token.



5. Click on the "GENERATE TOKEN" button and the token will be displayed. The long string of characters is the access token.
   Be certain to copy and save the access token somewhere safe! Some place that you will be able to remember/find when you need to use the token. *This is the only time nLighten will display the access token!*



6. If you click on "RETURN TO LIST OF TOKENS" you will see a list of all your access tokens and when they were lasted used. You can click on the trashcan icon at the right of each token to delete it. If you lost a token you previously created, you can always delete that token and create a new one.



REST API

Media Type

The REST API uses an `application/json` media type.

Request HTTP header:

```
{ "Accept": "application/json" }
```

Response HTTP header:

```
{ "Content-Type": "application/json; charset=utf-8" }
```

Response Body Structure

The REST API responds with the JSON:API v1.0 document structure. Note however, as stated above, the REST API does ***not*** adhere to the JSON:API requirement of media type `application/vnd.api+json`.

For example, a request for `/api/cases/477` will return an JSON object similar to the following.

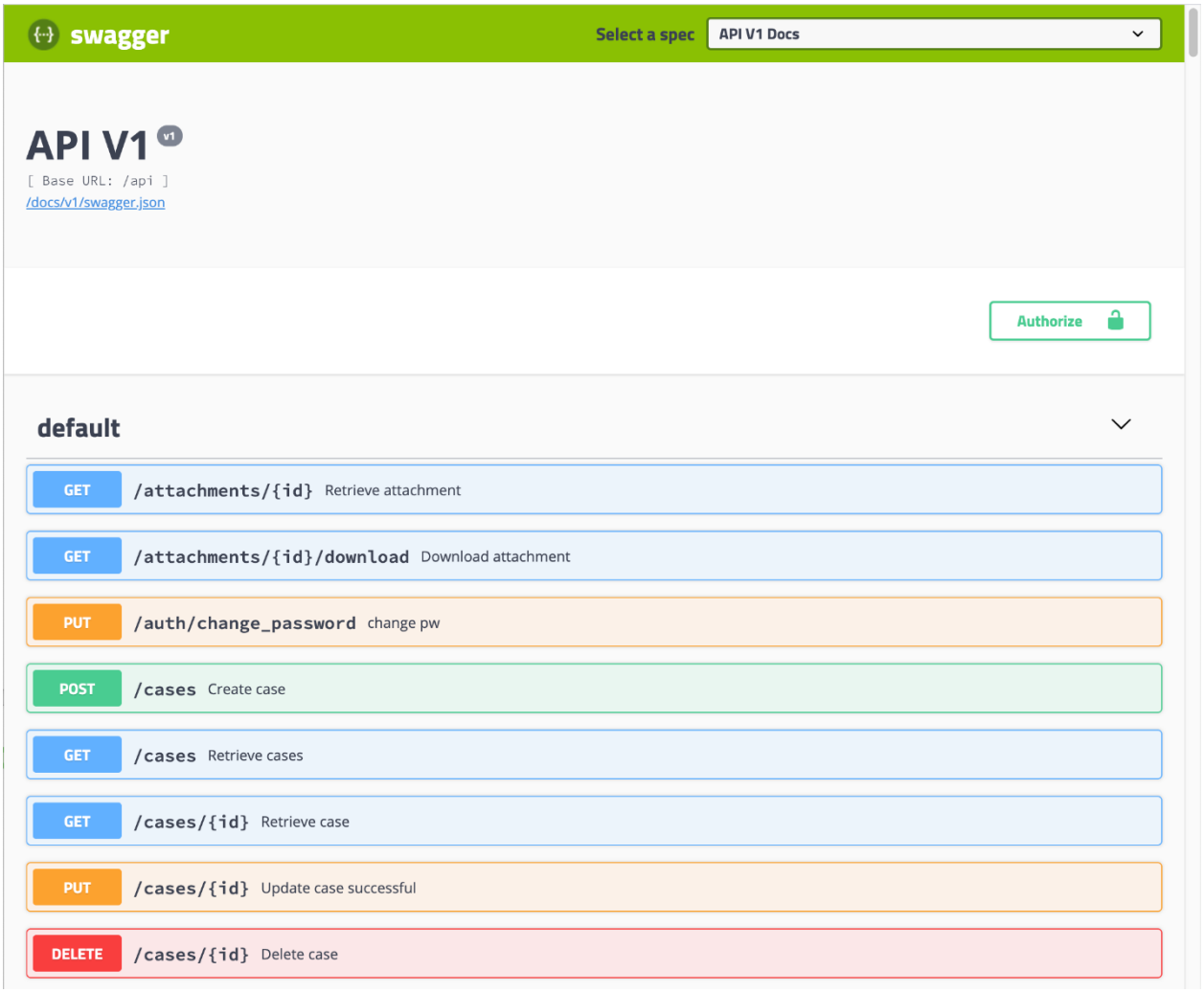
```
{
  "data": {
    "id": "477",
    "type": "cases",
    "attributes": {
      "id": 477,
      "title": "Suspicious VPN Connection Detected",
      "author_id": 4,
      // ... redacted remaining attributes
    },
    "relationships": {
      "author": {
        "data": {
          "id": "4",
          "type": "users"
        }
      },
      // ... redacted remaining relationships
    },
    "links": {
      "comments": "/api/cases/477/comments"
    }
  }
}
```

Interactive Documentation

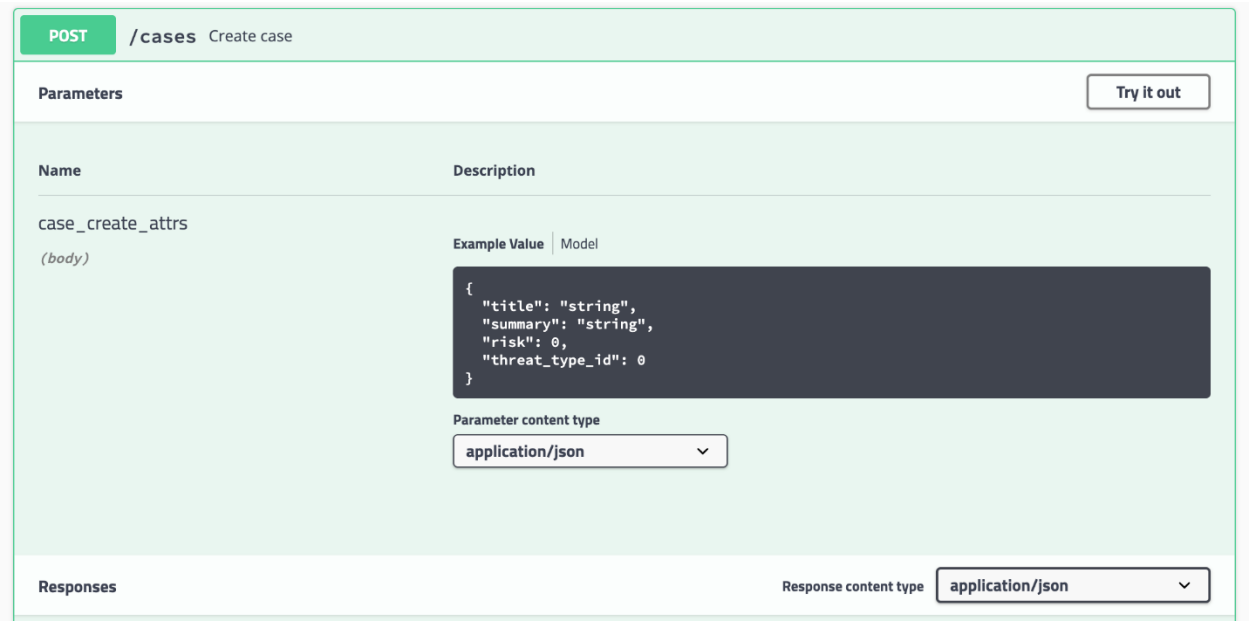
Interactive documentation for the REST API is generated by Swagger and is made available at the following URL.

```
https://<pea>.nl.cybraics.com/docs/swagger-ui/index.html
```

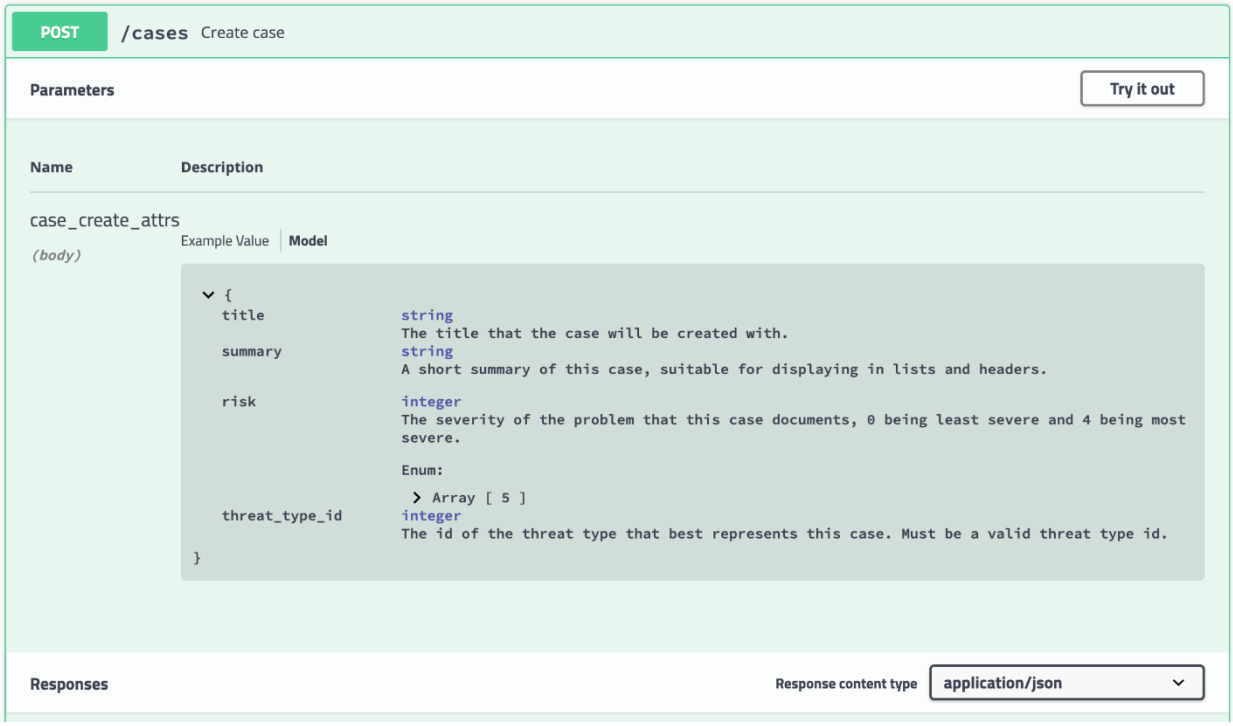
When loaded, the page should look something like the image below.



You can click on one of the listed actions to see the accepted parameters. For example, `POST /cases` looks like the image below. Accepted parameters are `title`, `summary`, `risk`, and `threat_type_id` as shown by the "Example Value".

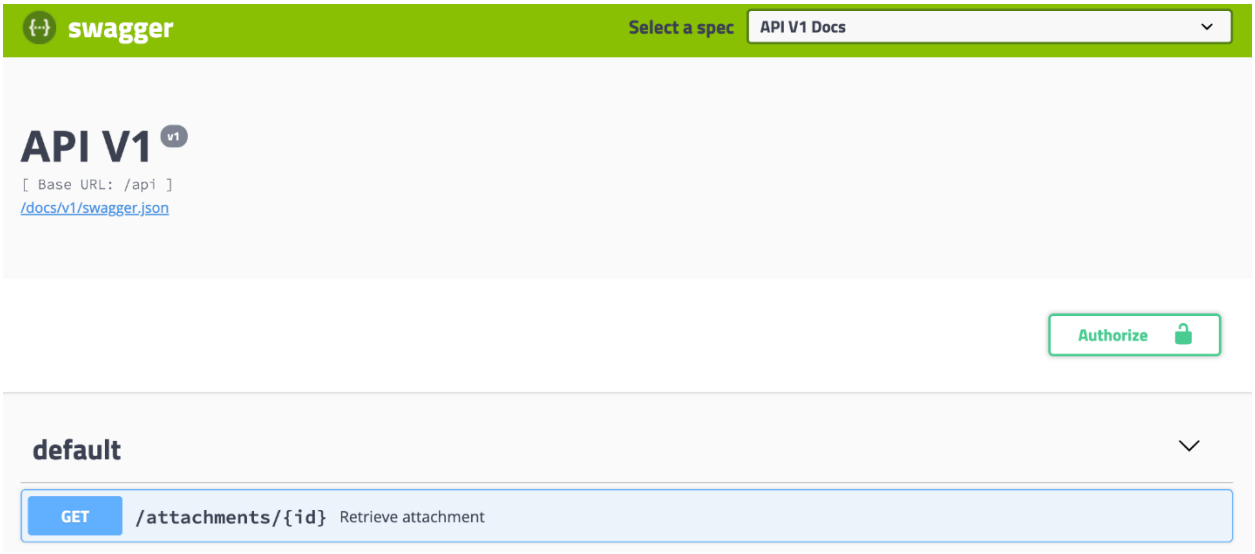


Click on the "Model" link and you will get a description of each parameter.

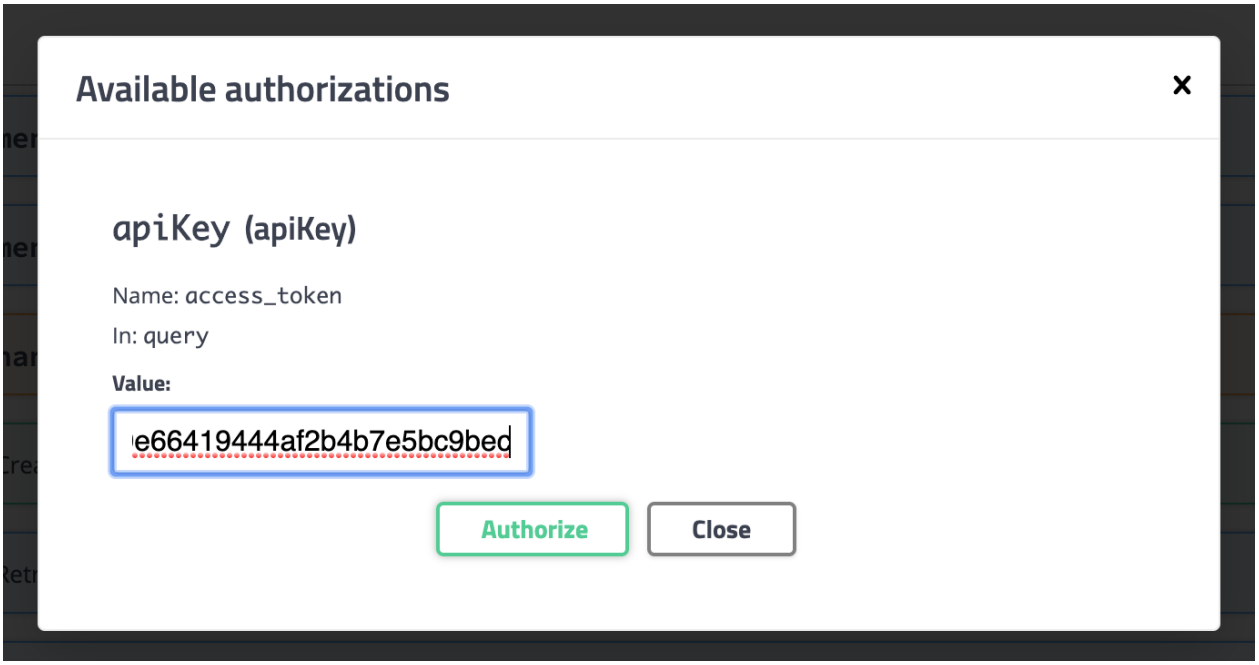


You can experiment with the REST API directly from this page using the following steps.

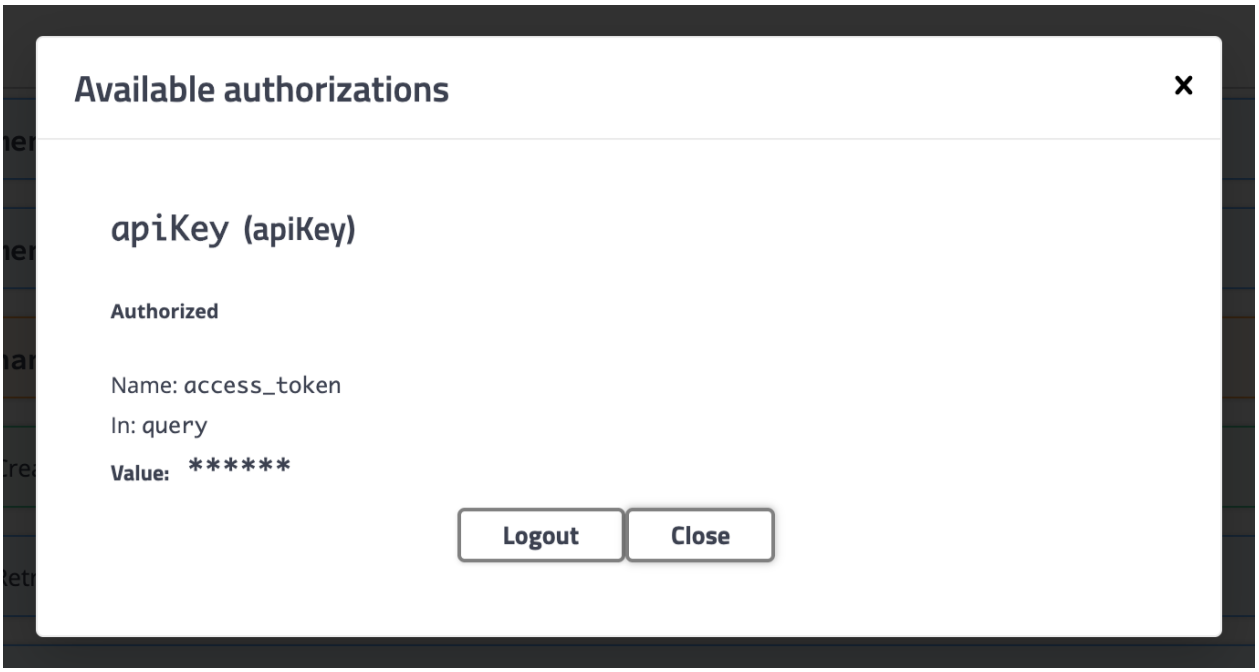
1. Click on the "Authorize" button at the top right.



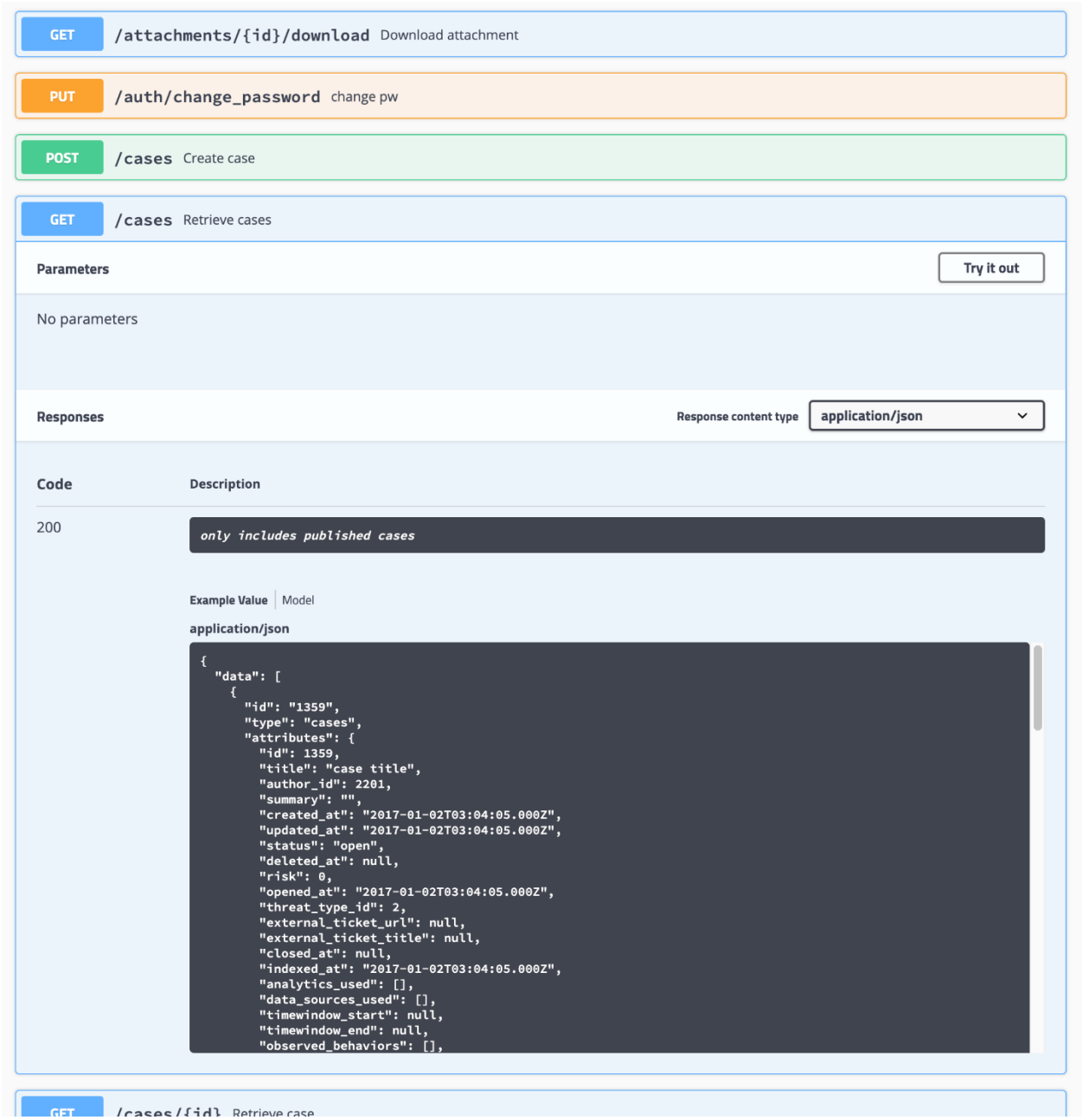
2. Enter the access token you generated earlier and click on the "Authorize" button.



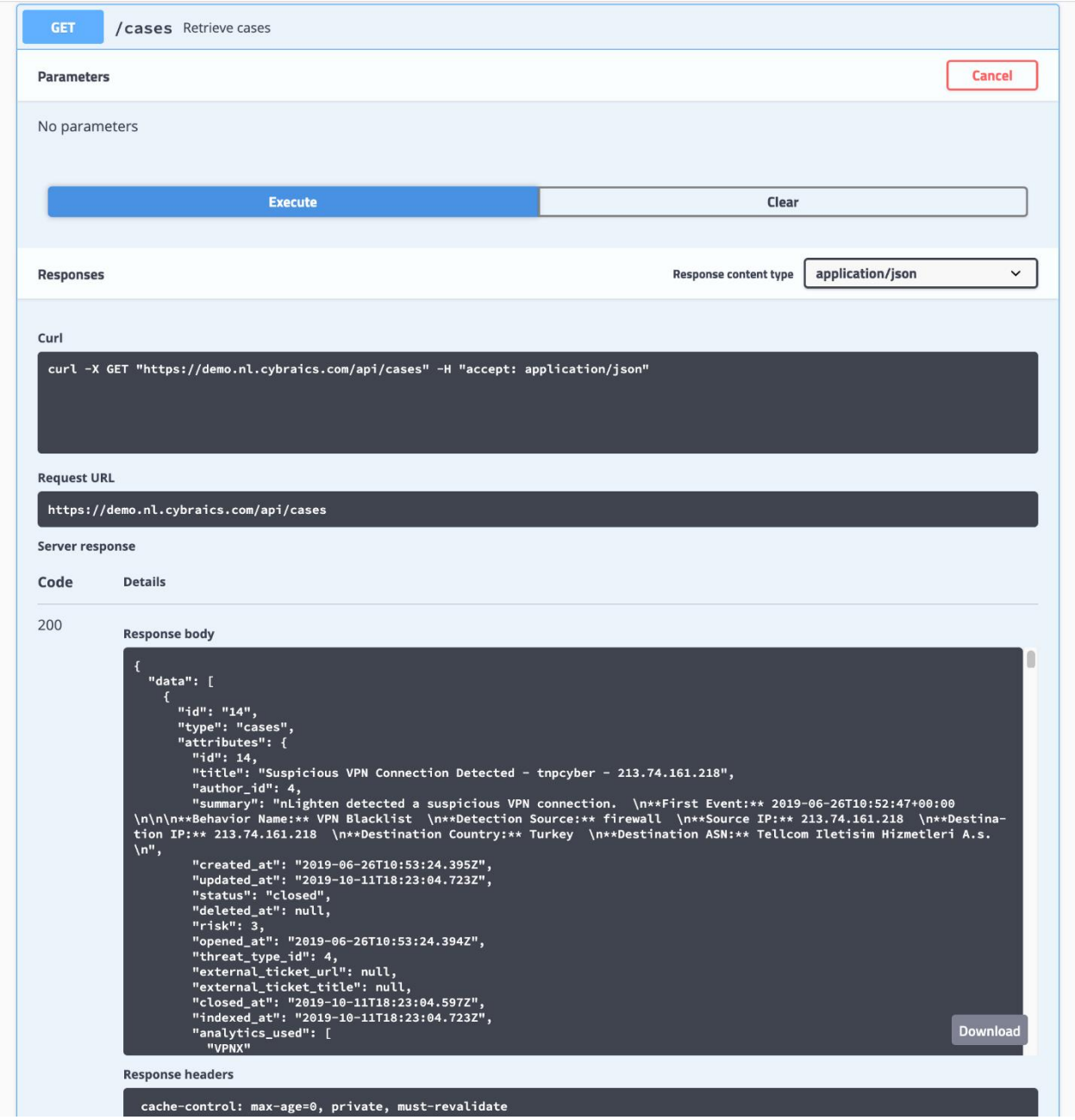
3. Click on the "Close" button to get back to the REST API.
   Whenever you are done experimenting with the API, you can again click the "Authorize" button to display this window, then click on the "Logout" button to remove your token.



4. Click on one of the listed actions, for example, `GET /cases`.



5. Click on the "Try it out" button, then click on the "Execute" button that appears. The request will be made, and the results will be displayed in the page.



GraphQL API

Media Type

The GraphQL API uses an `application/json` media type.

Request HTTP header:

```
{ "Accept": "application/json" }
```

Response HTTP header:

```
{ "Content-Type": "application/json; charset=utf-8" }
```

Response Body Structure

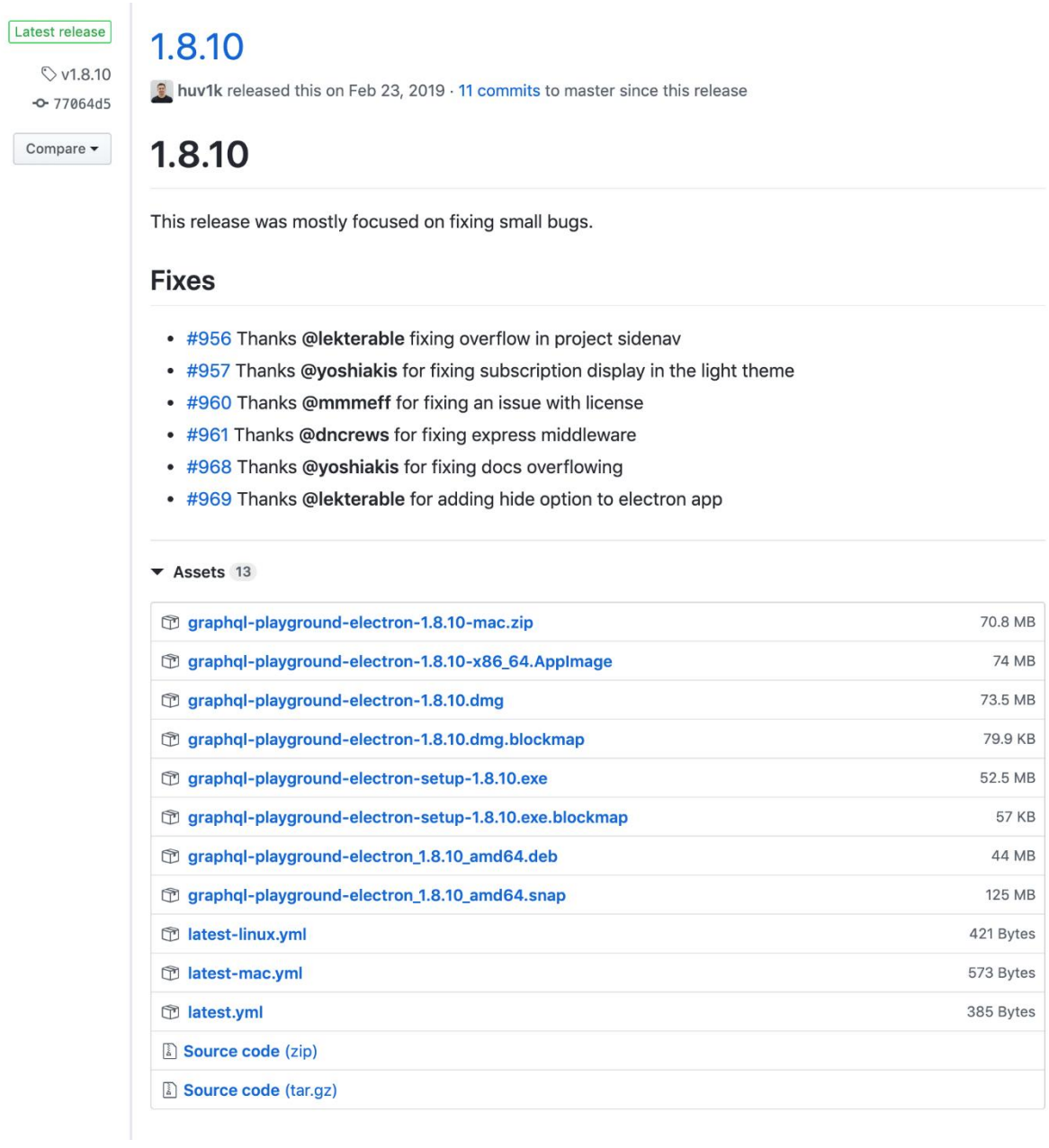
The GraphQL API responds with the standard GraphQL response format.

For example, using the following query:

```
query {
 outlierById(id: 68100) {
   id
   analyticName
   timewindowStart
   timewindowEnd
   behavior {
    name
    category
   }
 }
}
will return a structure as shown below.
{
 "data": {
  "outlierById": {
   "id": "68100",
   "analyticName": "PAX",
   "timewindowStart": "2019-07-25T05:13:45.000000Z",
   "timewindowEnd": "2019-07-25T05:14:13.000000Z",
   "behavior": {
    "name": "Palo Alto High Alert",
    "category": "unusual_user_behavior"
r   }
  }
 }
}
```

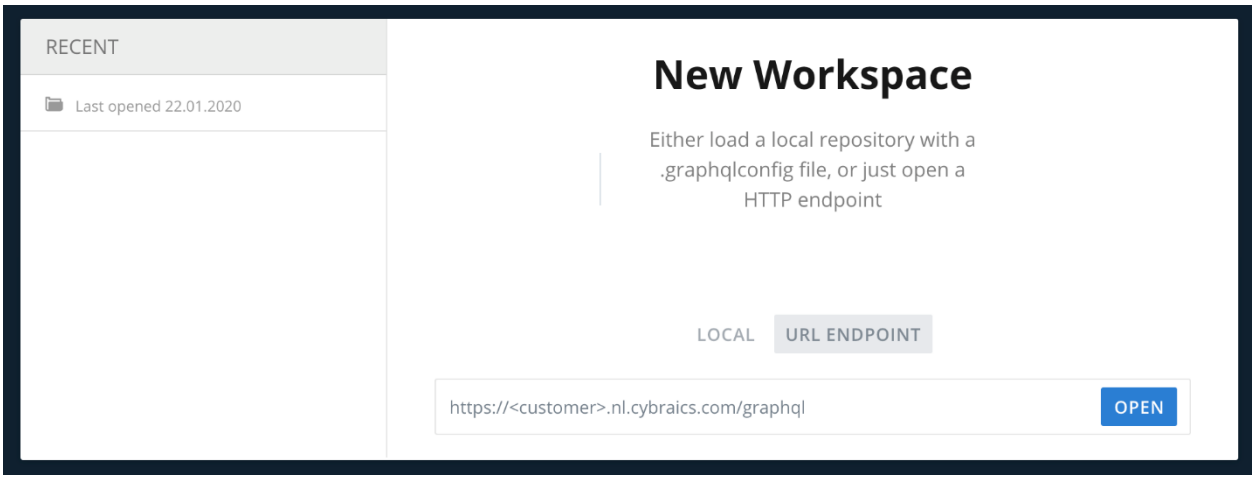
The best way to learn the GraphQL API is to browse it using a GraphQL tool, such as GraphQL Playground or GraphiQL. The following steps describe how to browse the API using GraphQL Playground.

1. Download the GraphQL Playground distribution for your platform from the URL below. The distributions are listed under "Assets" for the release you want to use. Install the distribution.
   https://github.com/prisma-labs/graphql-playground/releases

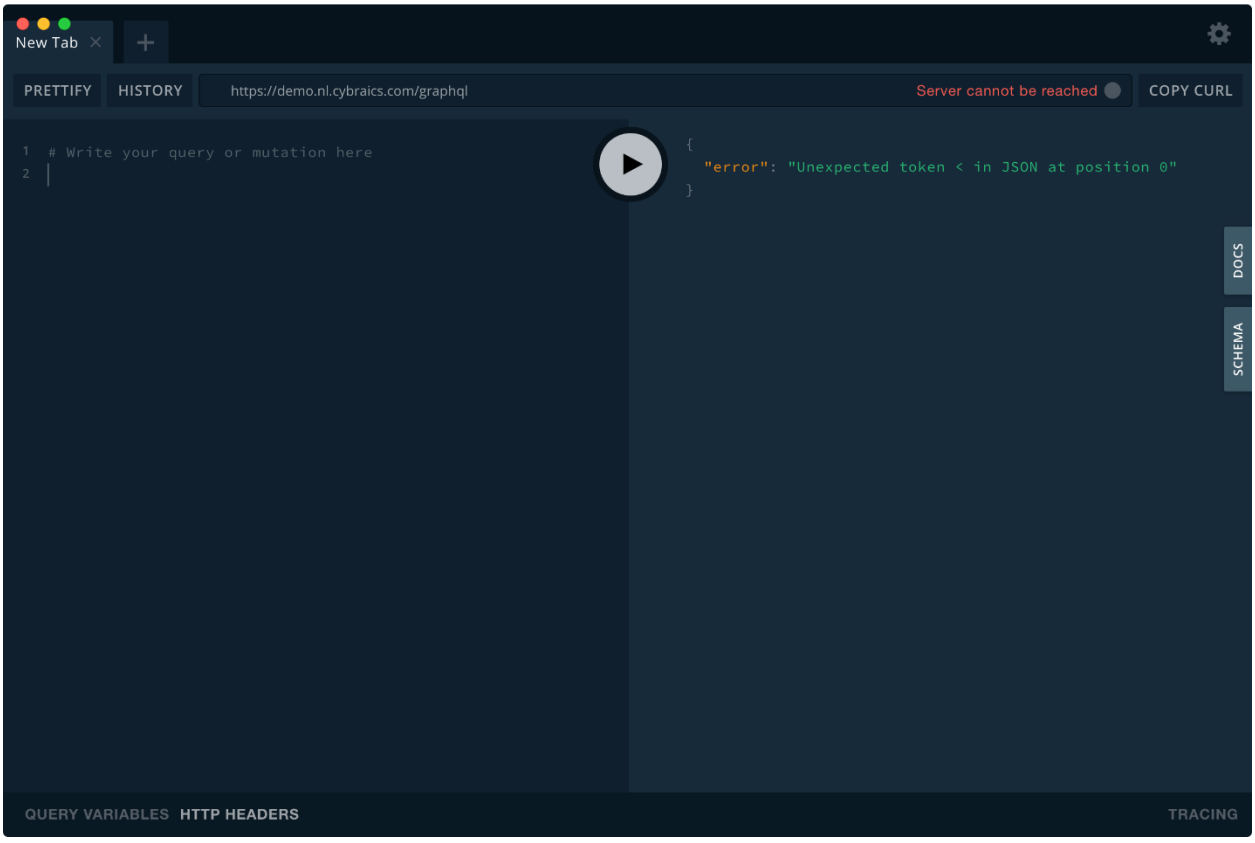


2. Run GraphQL Playground and you should see a window like the image below. Select "URL ENDPOINT" and enter the URL for the nLighten instance plus `/graphql` on the end.

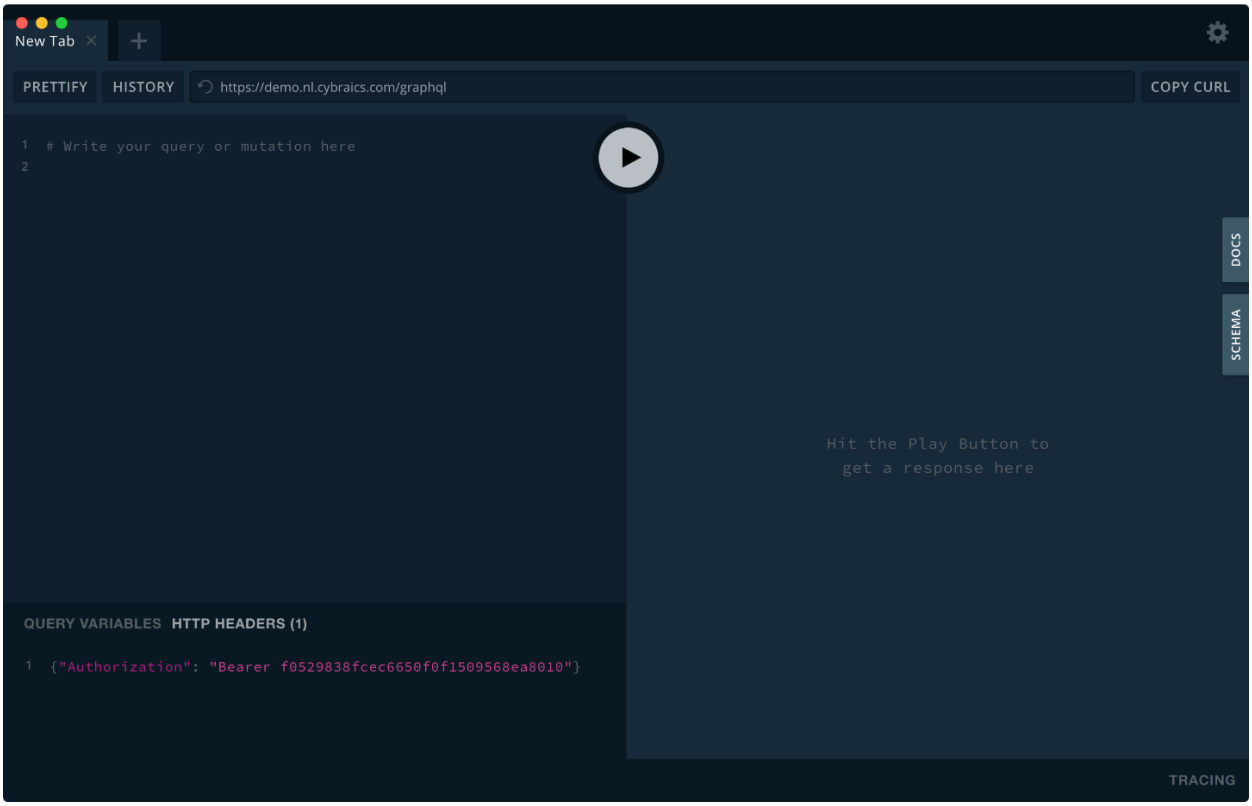
```
https://<pea>.nl.cybraics.com/graphql
```



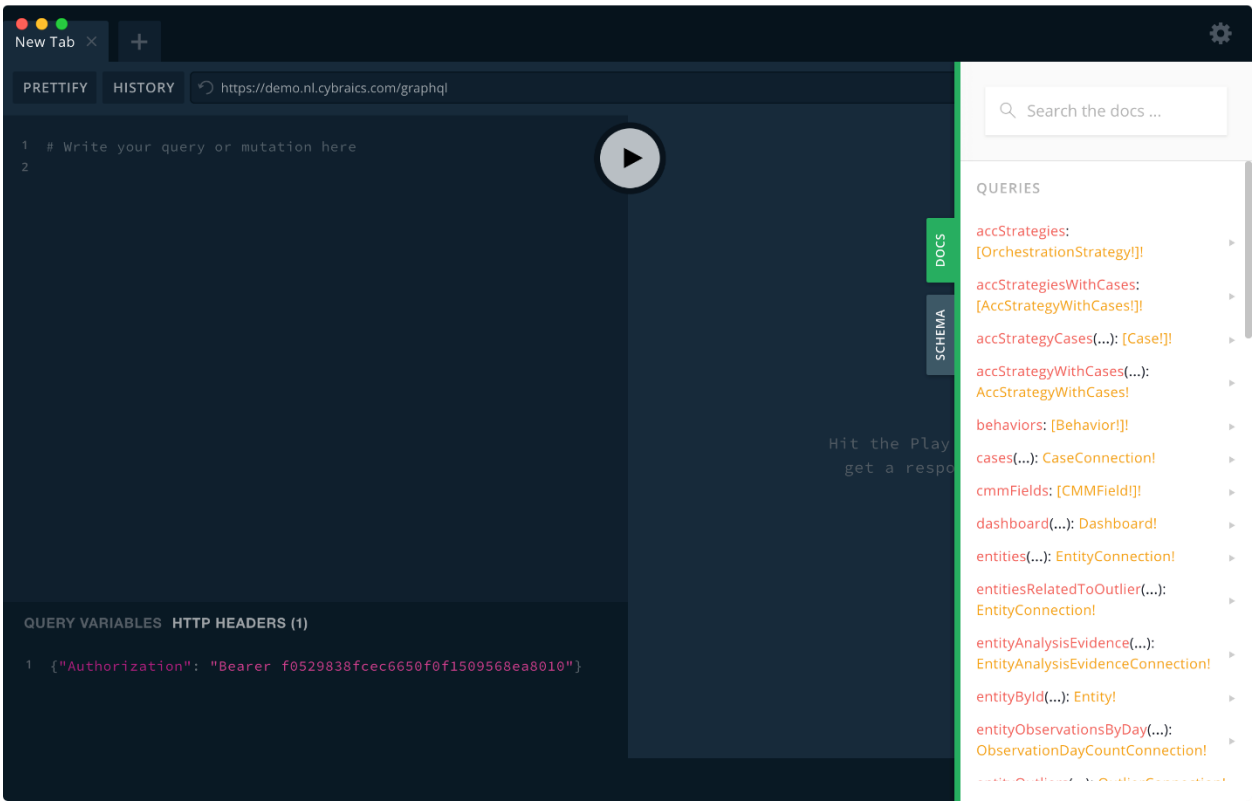
3. Click the "OPEN" button and you will see a screen that looks like the following image. There will be an error message displayed because you haven't added your access token yet so the server is responding with 401 Unauthorized.



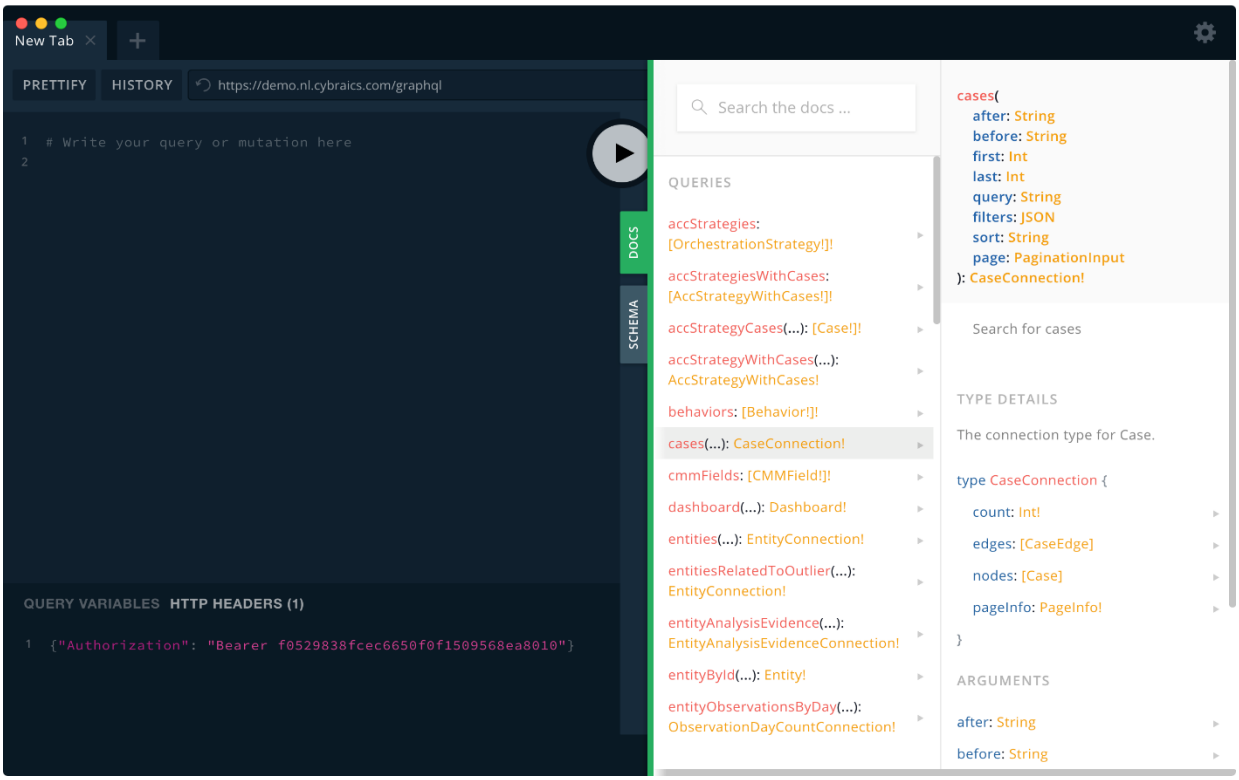
4. Click on "HTTP HEADERS" at the lower left of the window. Enter your access token as shown in the image below.



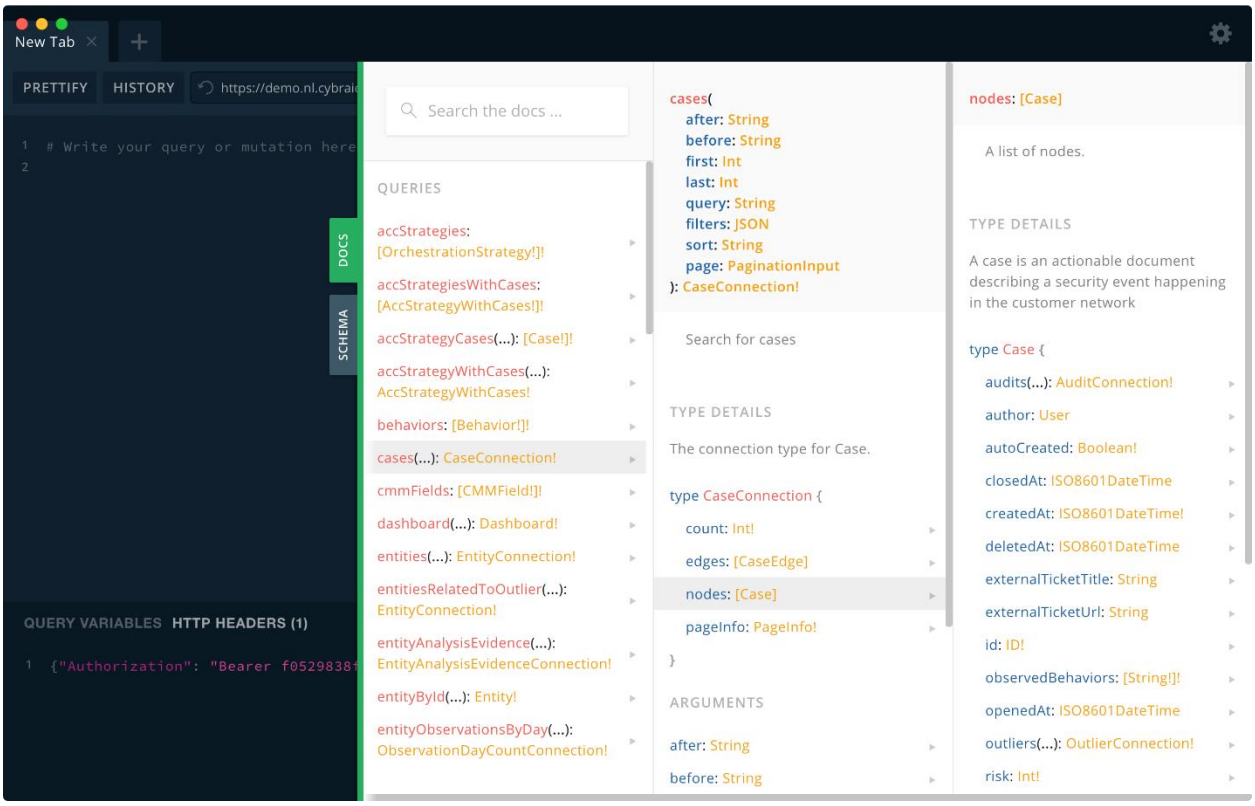
5. Click on the "DOCS" tab on the right to see the list of available queries and mutations.



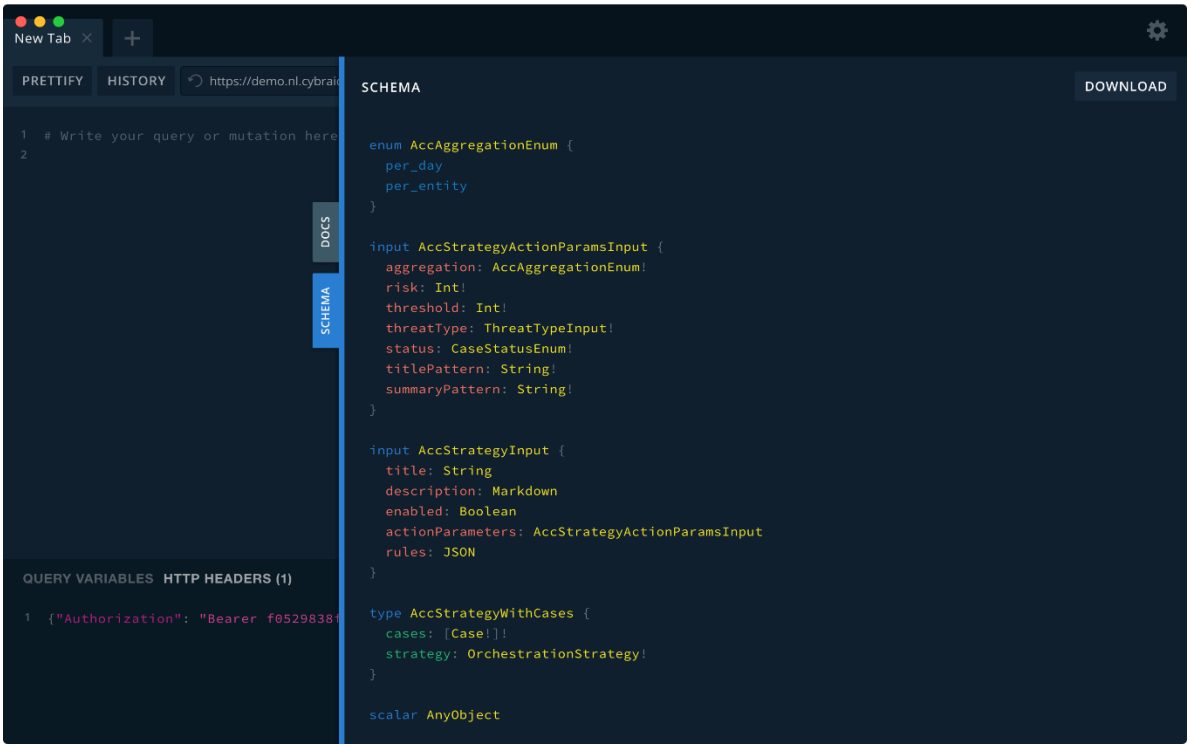
6. If you click on a query, it will expand to show details about that query. For example, the image below shows the `case` query.



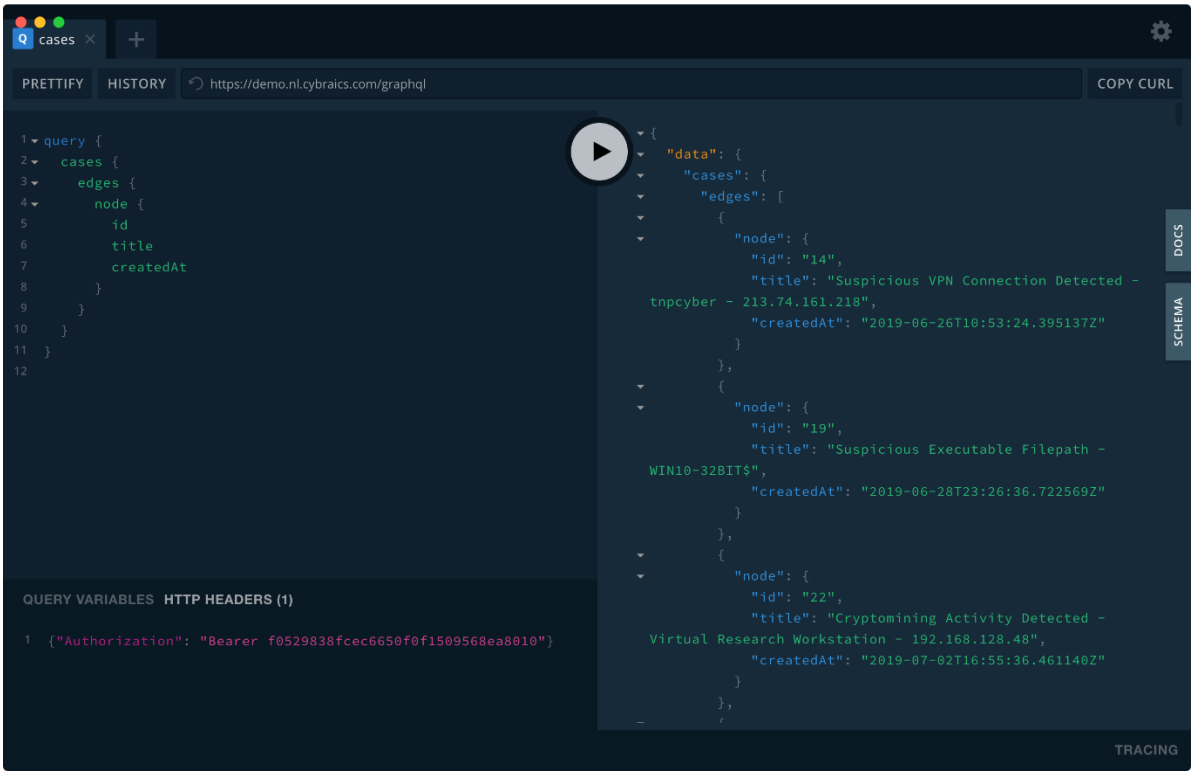
If you click on a type in the query, it will expand further to show you information about that type. You can continue to click on types to drill down further.



7. You can also click on the "SCHEMA" tab to see the GraphQL schema.



8. Finally, you can run a query, such as listing cases.



You can run both queries and mutations from GraphQL Playground but be ***careful*** because *mutations change data in the database!*