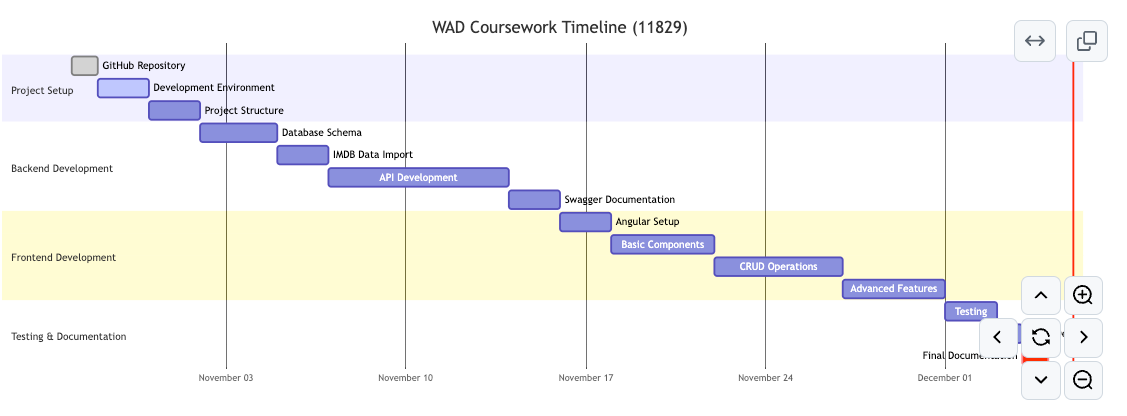
This document is a coursework report for the Web Applications Development (WAD) module by the student with ID 00011829 for Semester 1 at Westminster International University in Tashkent (WIUT).

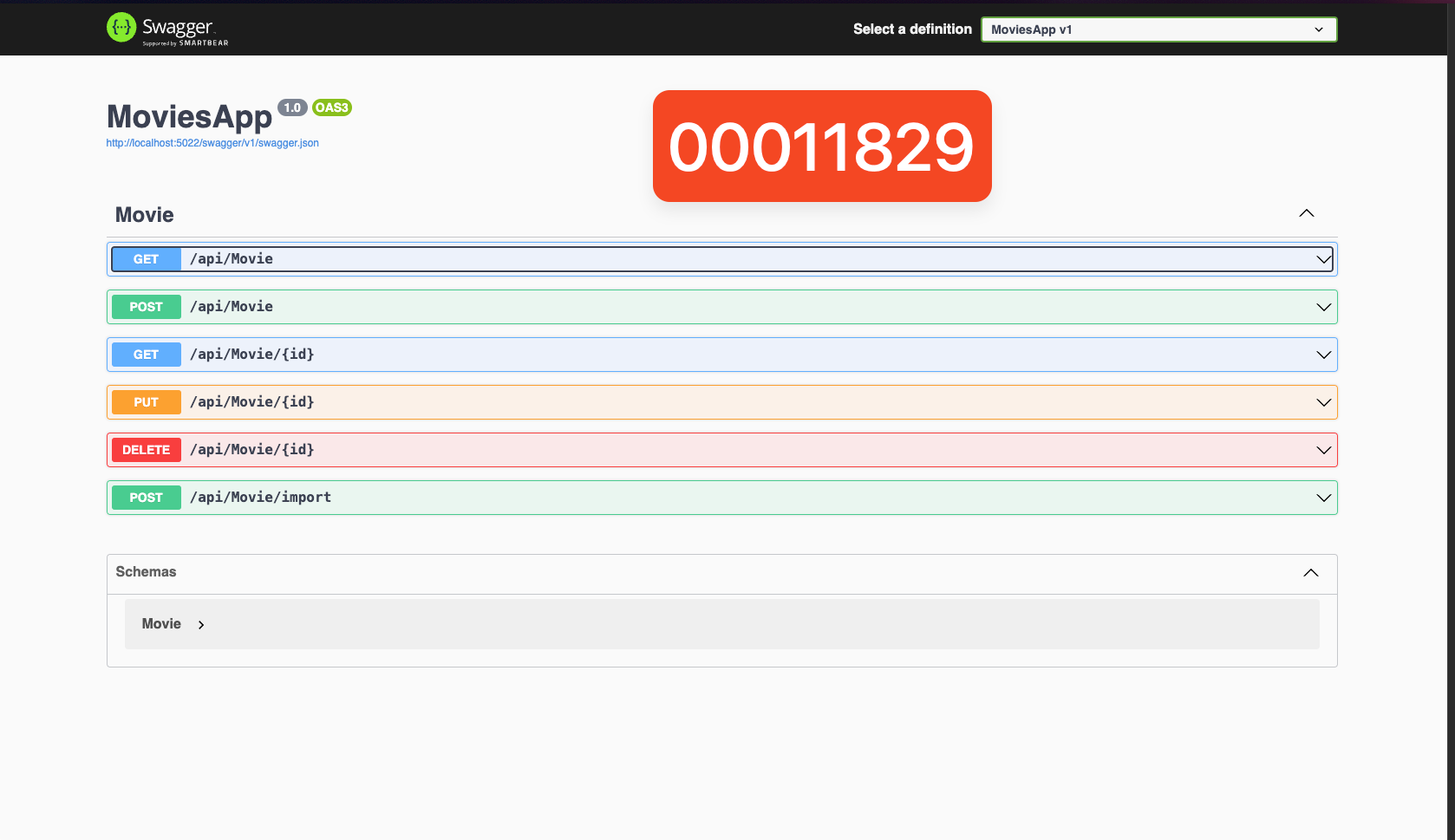
GitHub repository link: https://github.com/BexTuychiev/wad\_coursework\_11829

Development plan:  
  


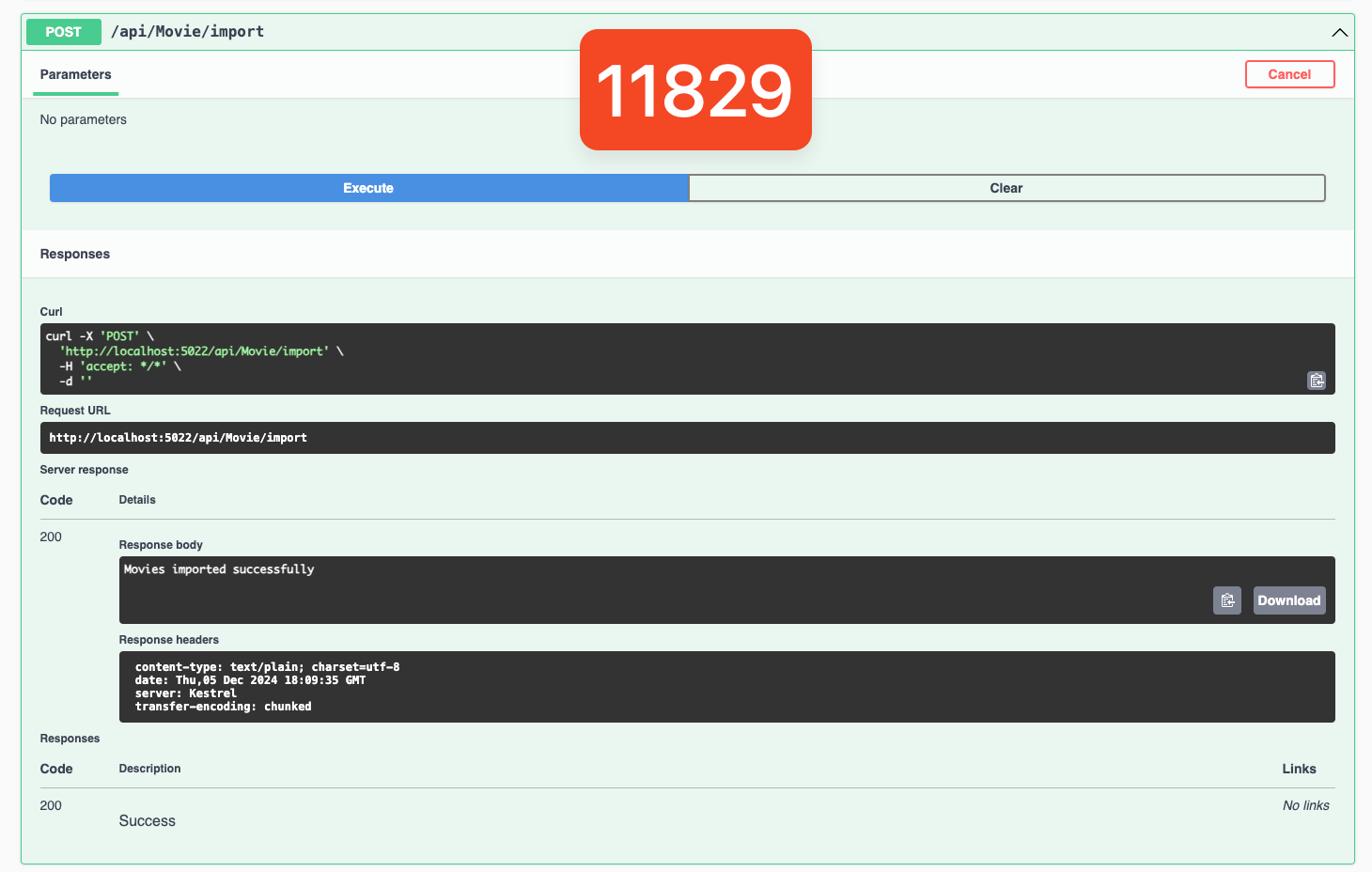
(Of course, I couldn’t follow the timeline, but I managed to create this chart at the start of November, and can be seen in the coursework’s GitHub repository).

# API (Backend) screenshots

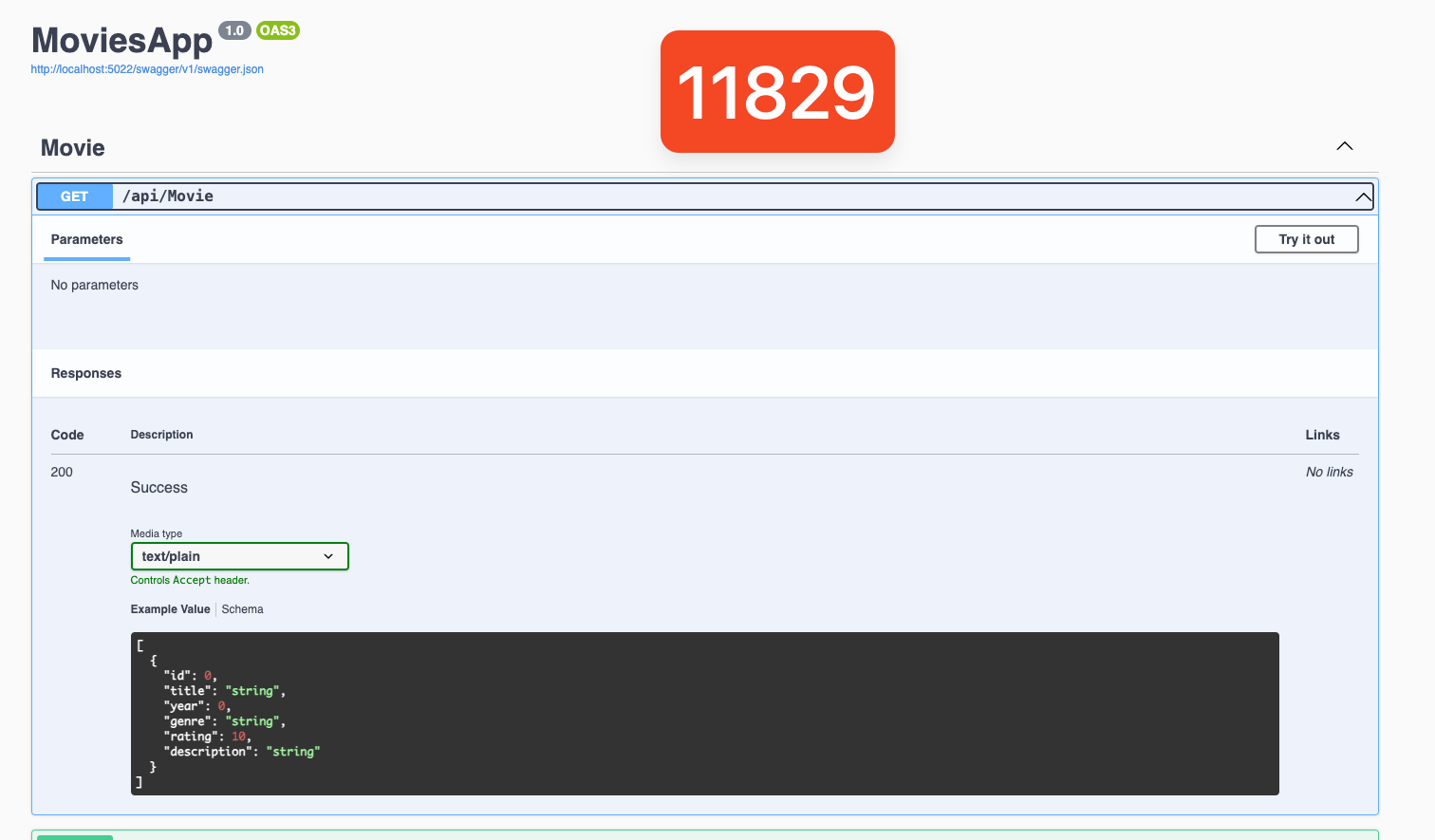
Here is the screenshot of my Swagger UI after everything was set up successfully and no endpoints were run yet:



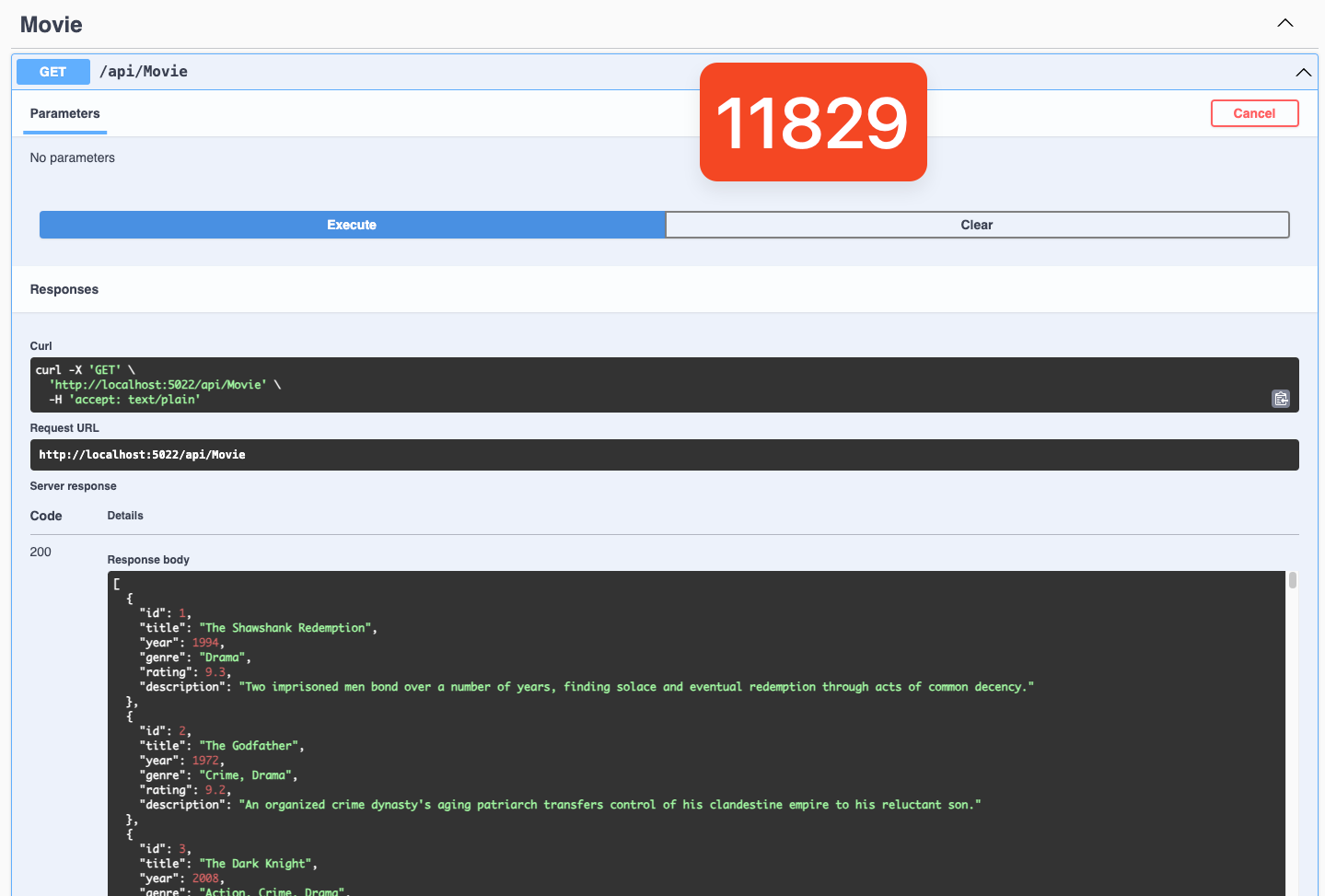
Since my app is a movies app, I decided to use [the IMDB top 1000 movies dataset from Kaggle](https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows). It was in a CSV format, so I wrote a quick python script to convert it into a desirable JSON format (suitable for my application, see scripts/csv\_to\_json.py file). Then, I created the necessary components to enable the “Import” endpoint in the API. The below screenshot shows the swagger Ui state after I click on “Try it out” and then “Execute”:



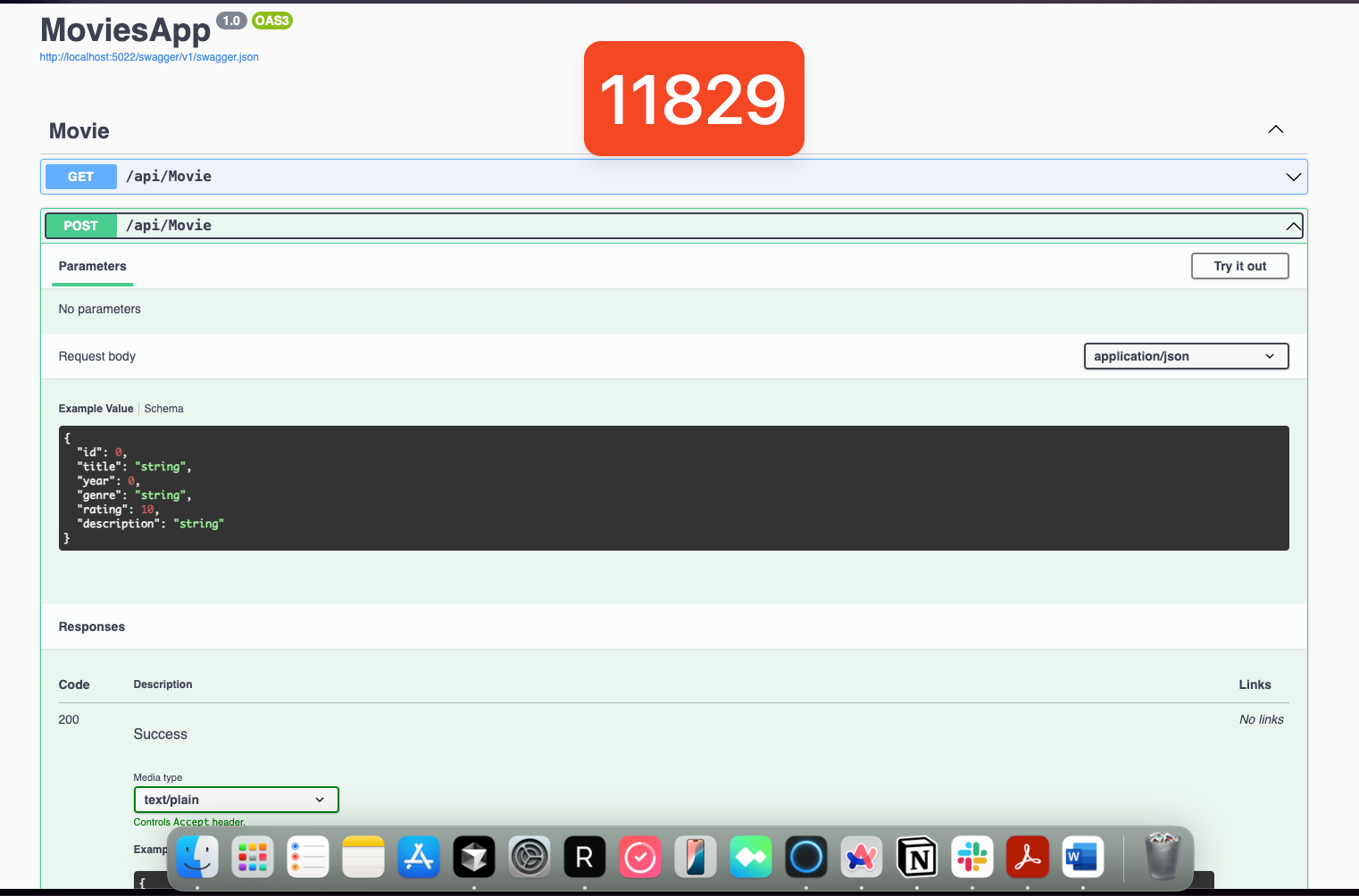
As you can see, I got a success message telling me the movies were imported. So, let’s check out one of the movies (this is the GET endpoint before running it):

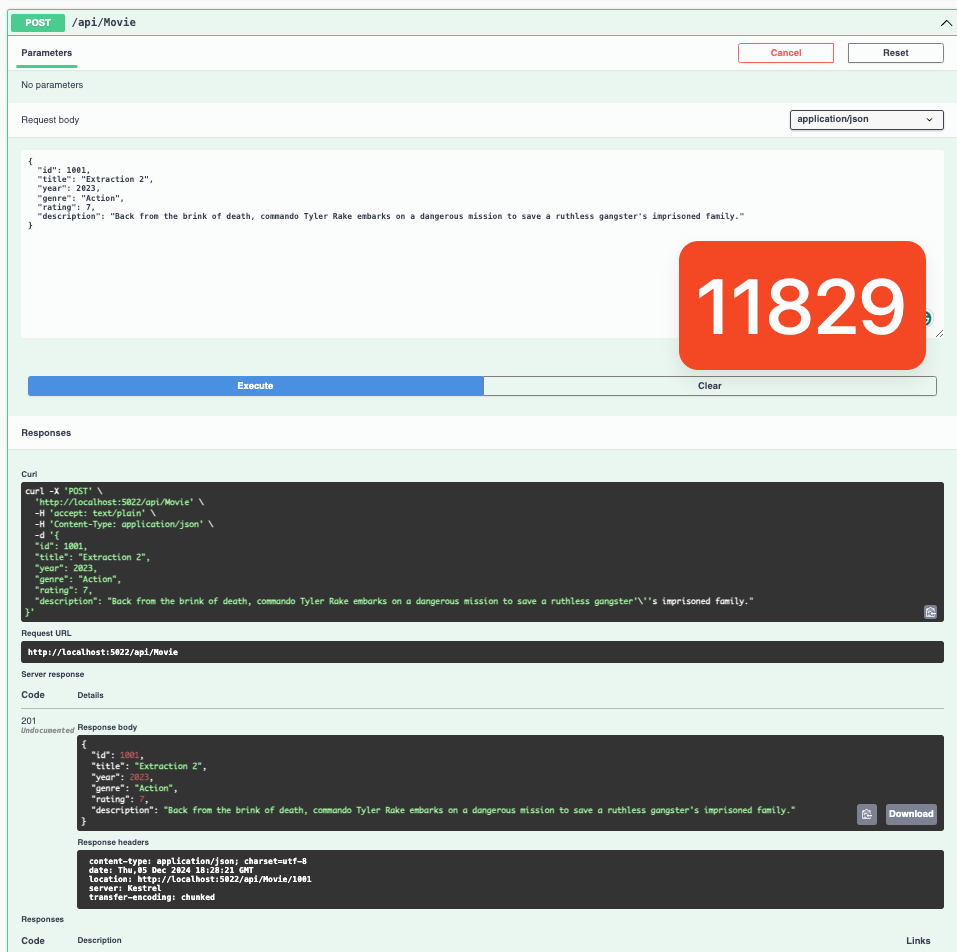


Here is the same endpoint after running it:

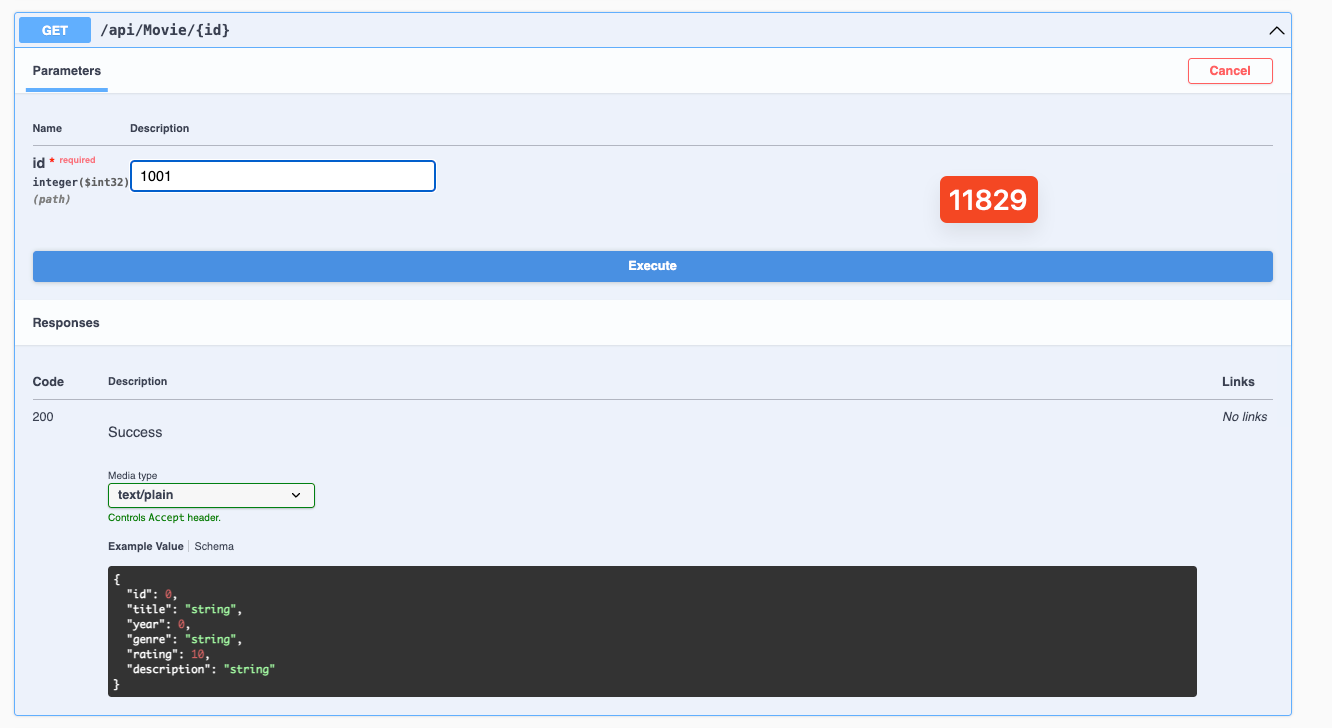


As you can see, the imported movies are present. Now, I will try to add a new movie with a POST request:

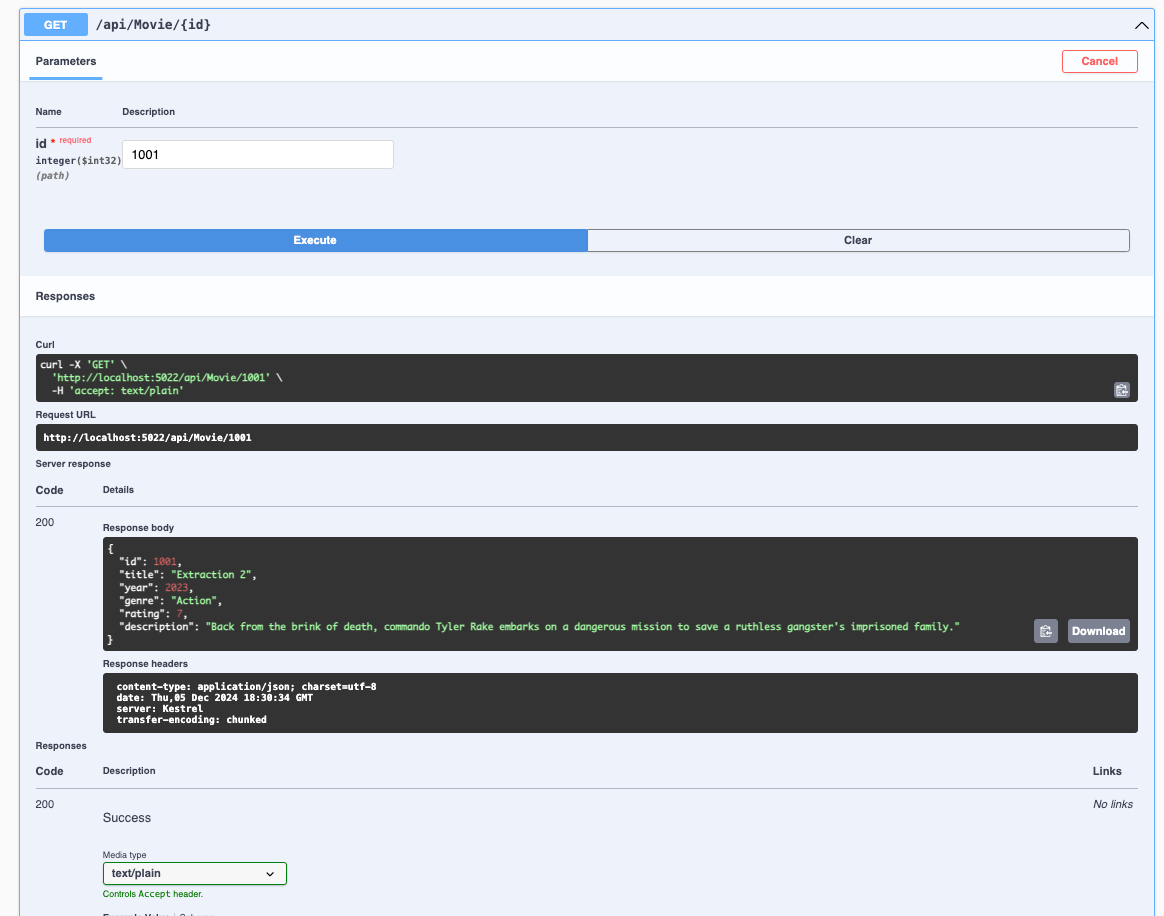
BEFORE:  
  


After:  


Now, I will try to get a movie using its ID. Let me try the one I just added with id 1001:

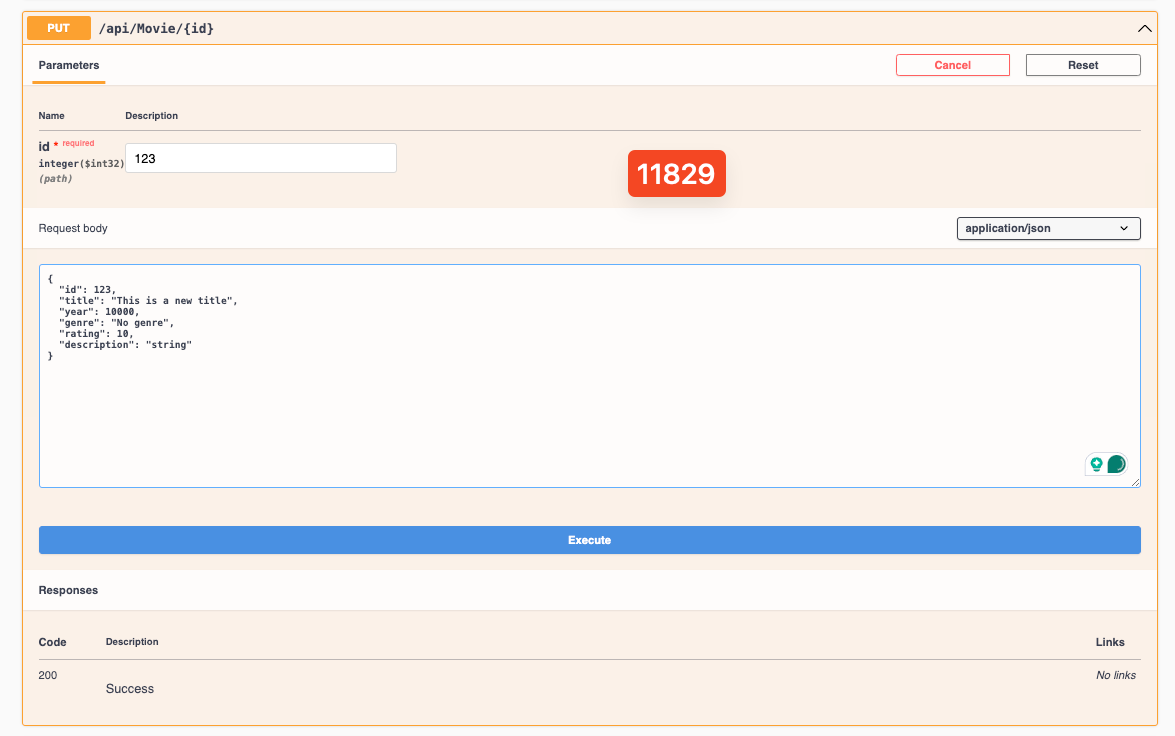
Before:  
  


After:

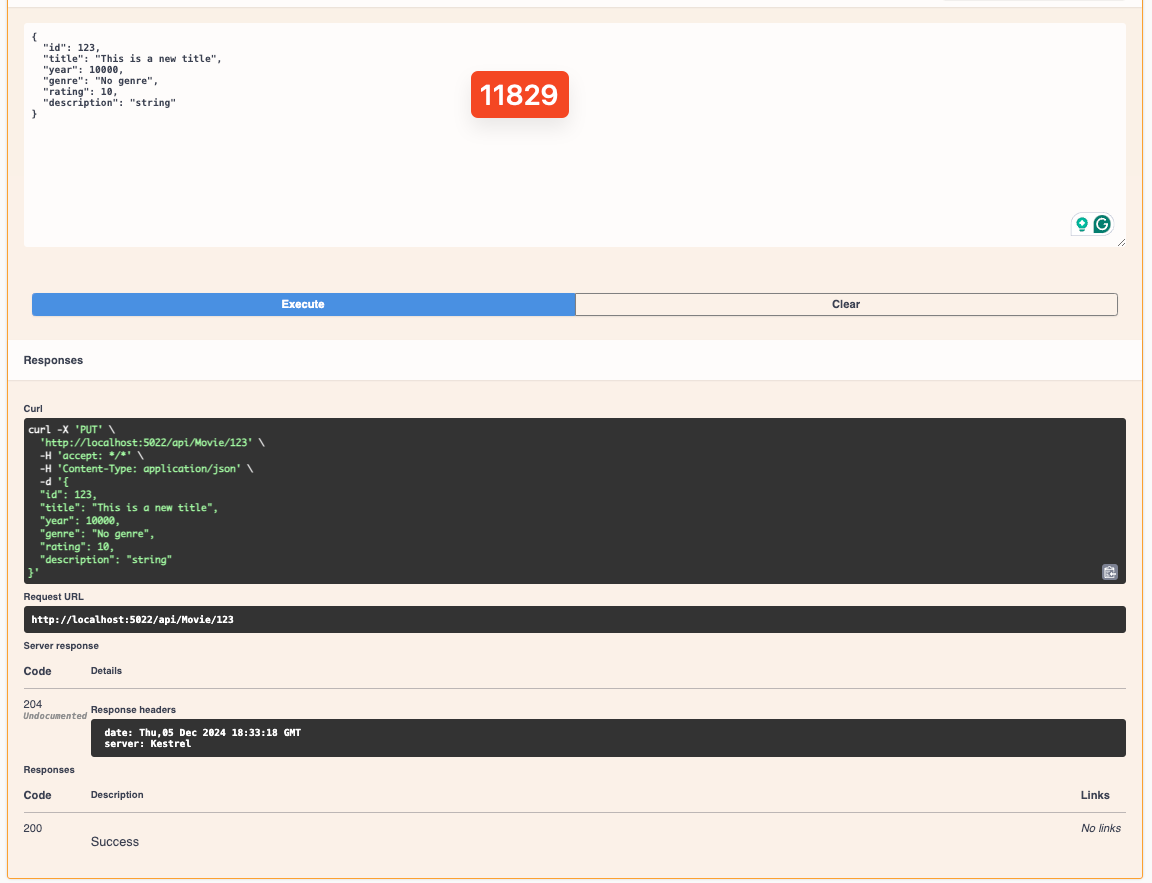


Now, I will update the details of one of the films using a PUT request:

Before:

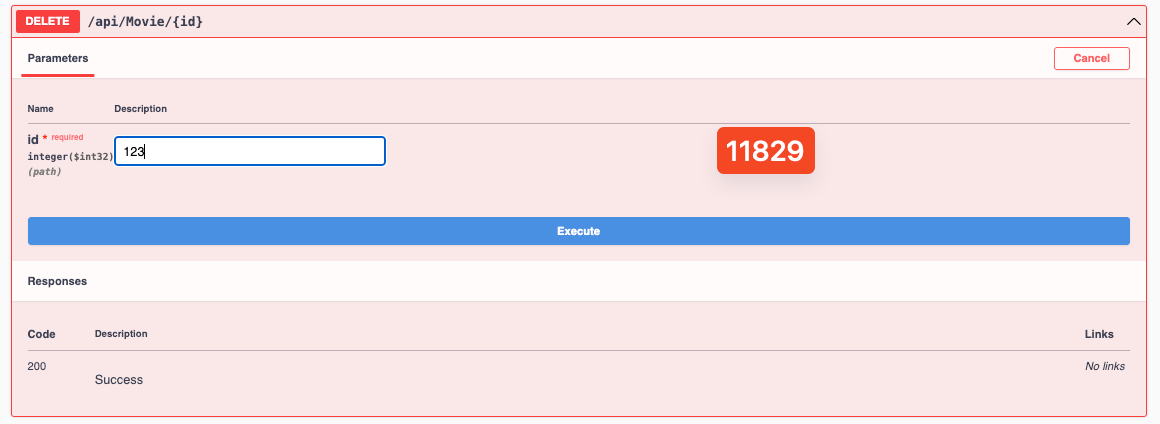


As you can see, I am updating a movie with 123 ID with some gibberish content. Here is what the endpoint looks like after execution:

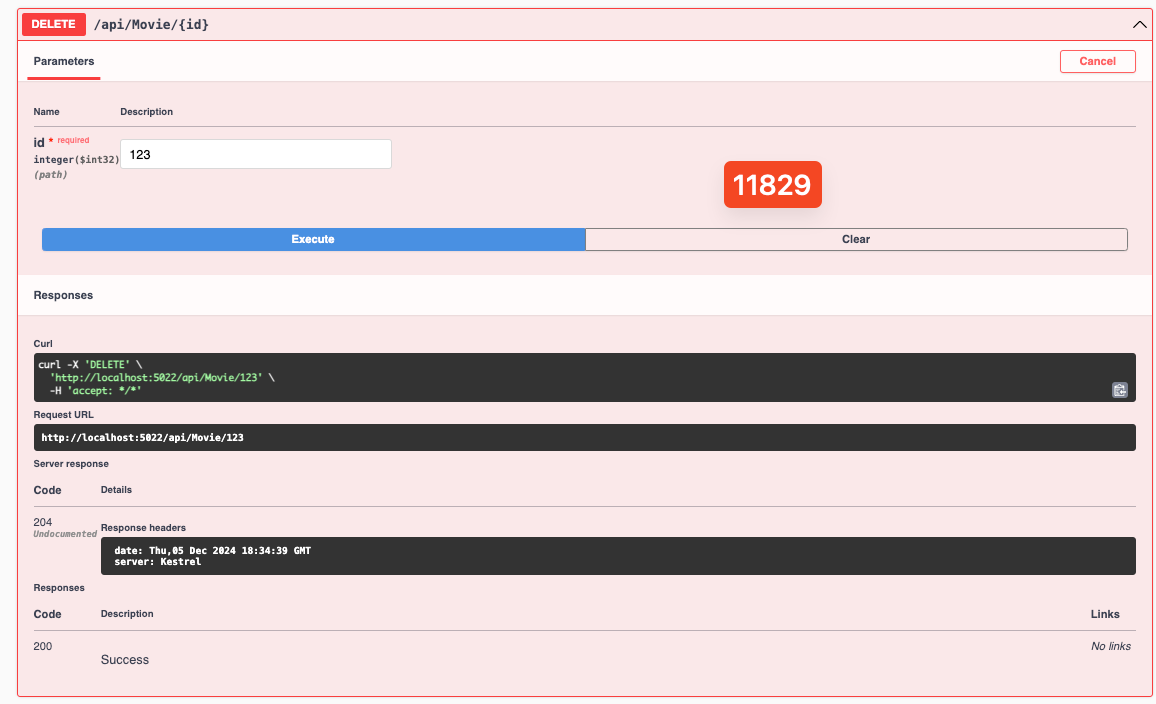


Now, I will delete this movie since it makes no sense to keep it in my database:

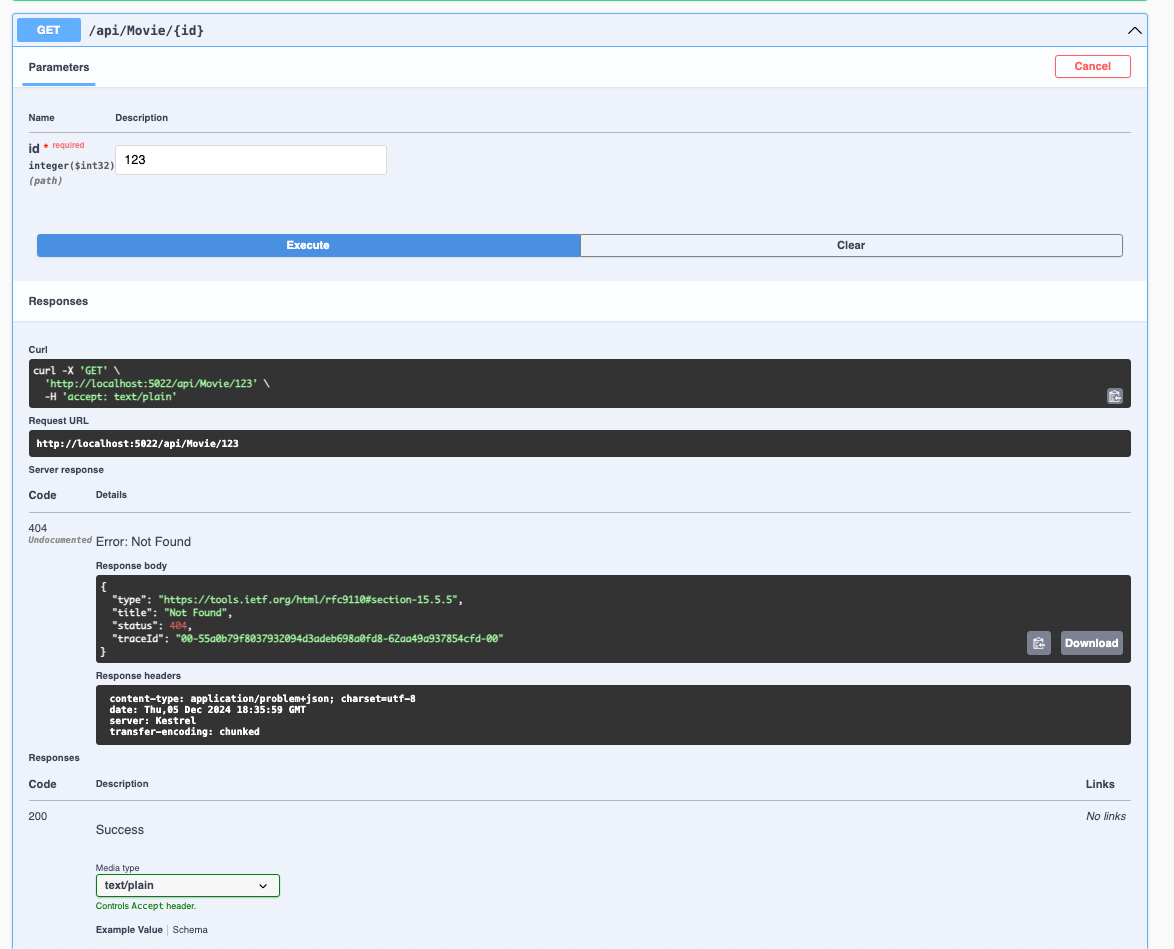
Before:



After:



To double-check, let’s do a GET request for that deleted movie:



As you can see, I got a not found error.