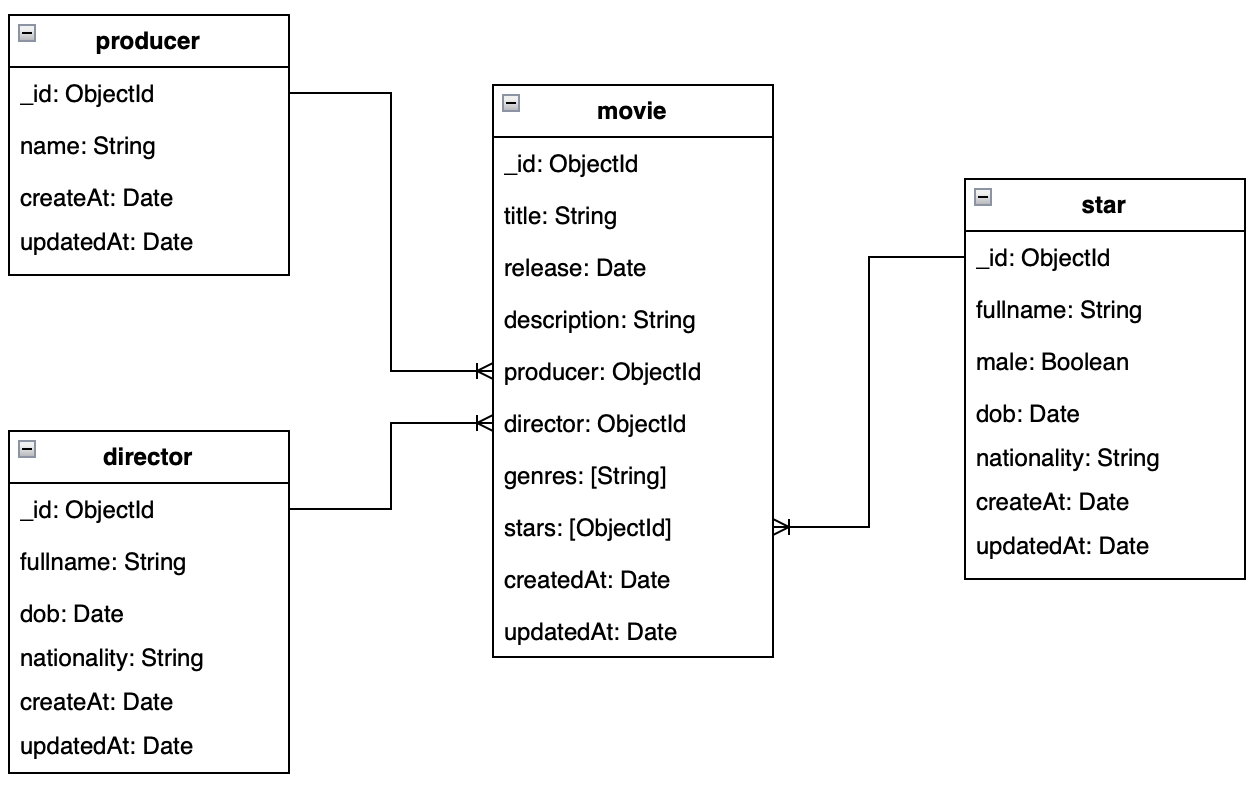
**PROGREST TEST 3**

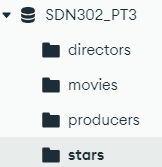
**Use the following database schema for doing this exam:**



**­­­­**

**Requirements:**

Using **MongoDB Compass**, create a database named “**SDN302\_PT3**”. Create collections corresponding to the data in the **Given data** directory. Then, import data into the collections created in the database.



Using the given-app, build a RESTFul API application that performs the requirements of the problem:

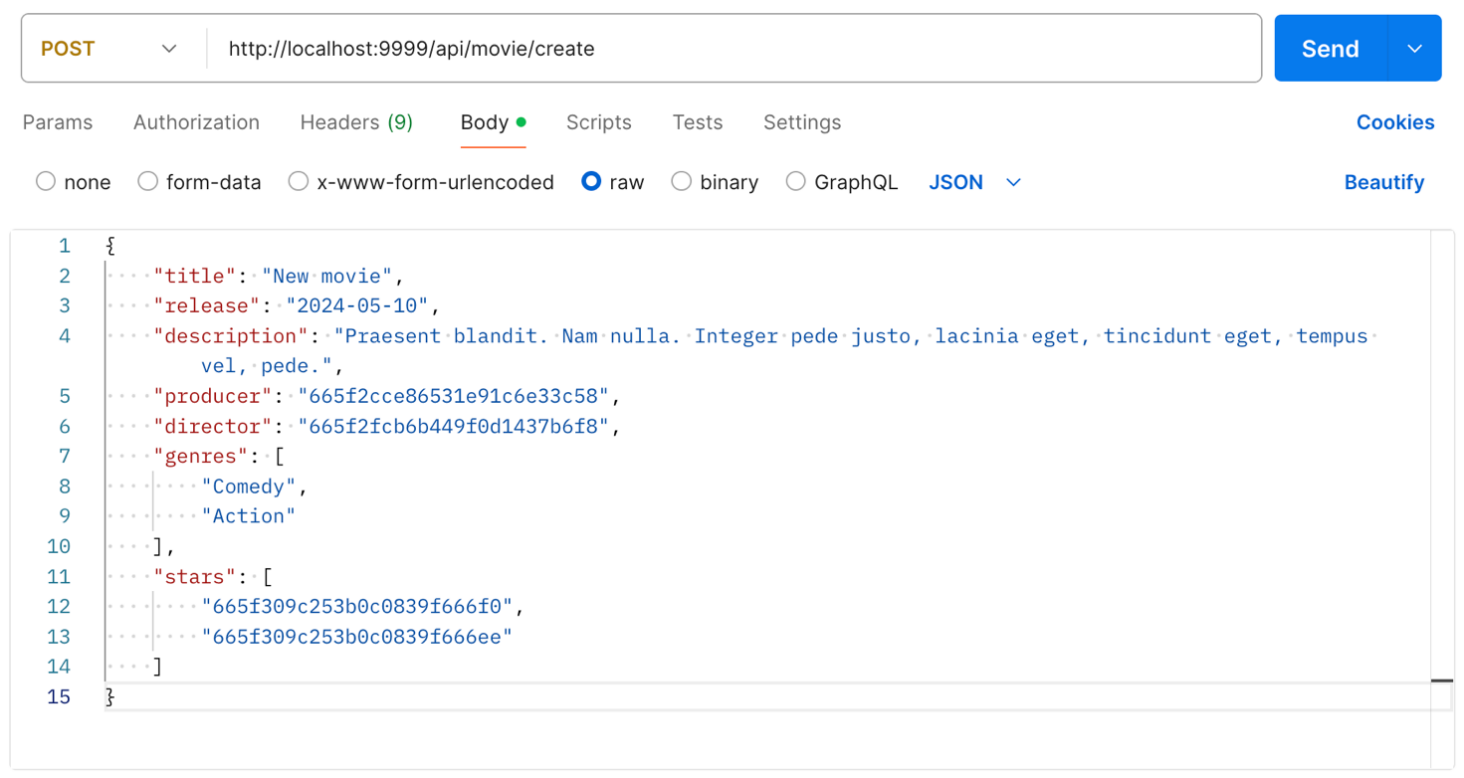
**Note: 0 will be given to the work that**

* Not using connection string to MongoDB in the **.env** file
* Not configuring the root path of RESTFul API Web App at: **http://localhost: 8888**

**Question 1.**

Create a new movie as shown in *Figure 1*, with the following API end-point parameters:

* URL: <http://localhost:9999/api/movie/create>
* Method: POST
* Input: a new movie object
* Example:



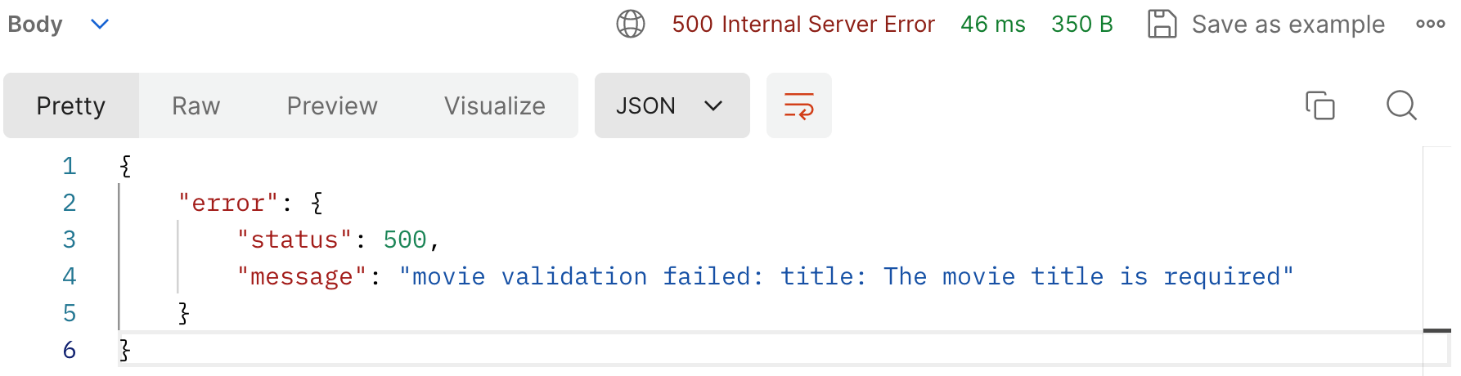
*Figure 1 - Send a request to create a new movie*

* The creation process was successful, the movie was saved to the database and returned data as shown in *Figure 2*.



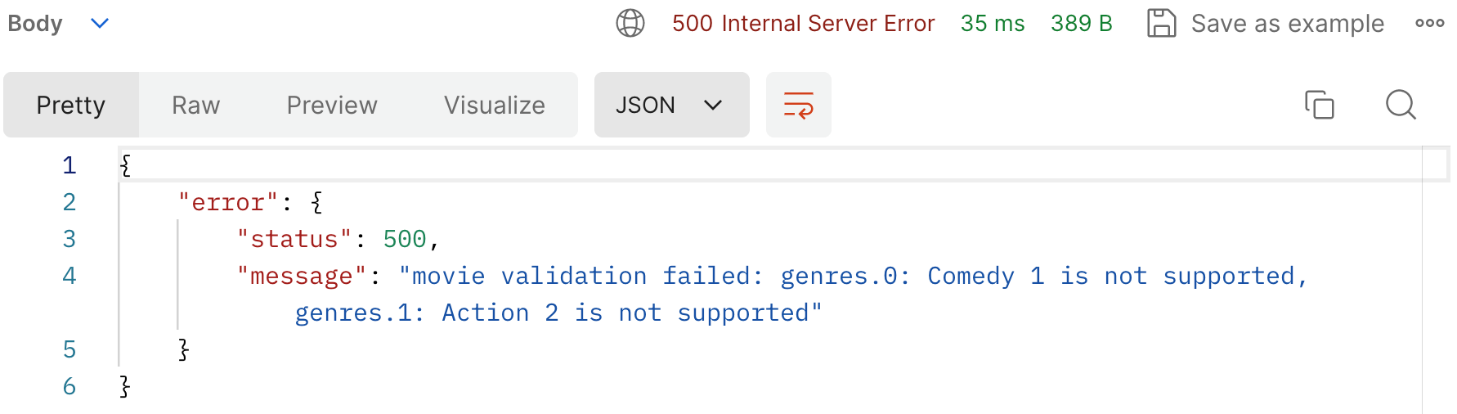
*Figure 2 - Receive results response from the API after creating success*

* When creating a new movie without a **title** field, the result is shown in *Figure 3*.



*Figure 3 - Returns an error object when the title field is missing*

* The movie genres only accept values ​​including: "**Action**, **Drama**, **Comedy**, **Cartoon**". If the data is incorrect, an error will be displayed as shown in *Figure 4*.

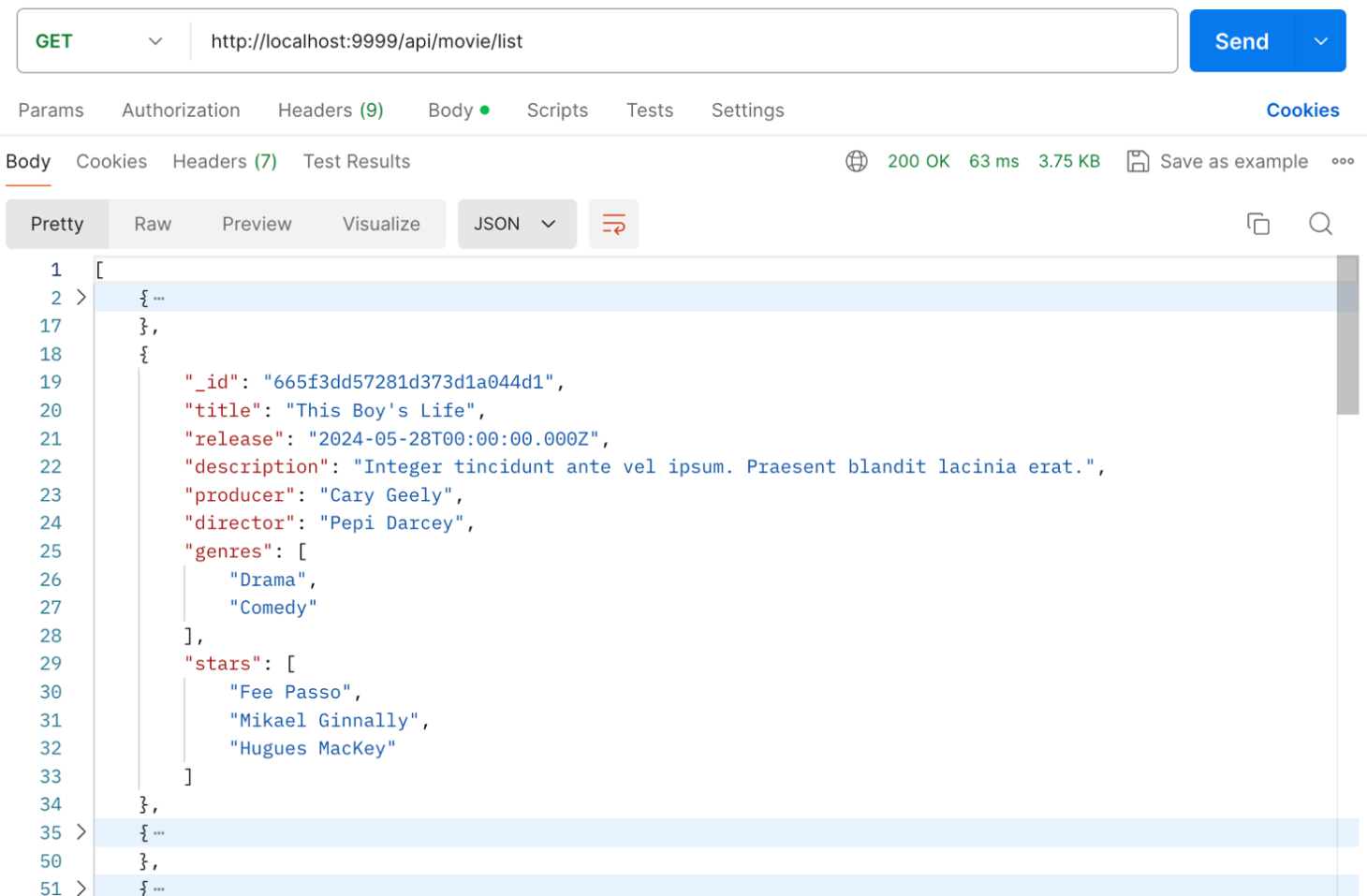


*Figure 4 - Returns an error object when the genres field has an incorrect value*

**Question 2**.

List all movies as shown in *Figure 5*, with the following API endpoint parameters:

* URL: <http://localhost:9999/api/movie/list>
* Method: GET
* Example:



*Figure 5 - List of movies returned successfully*

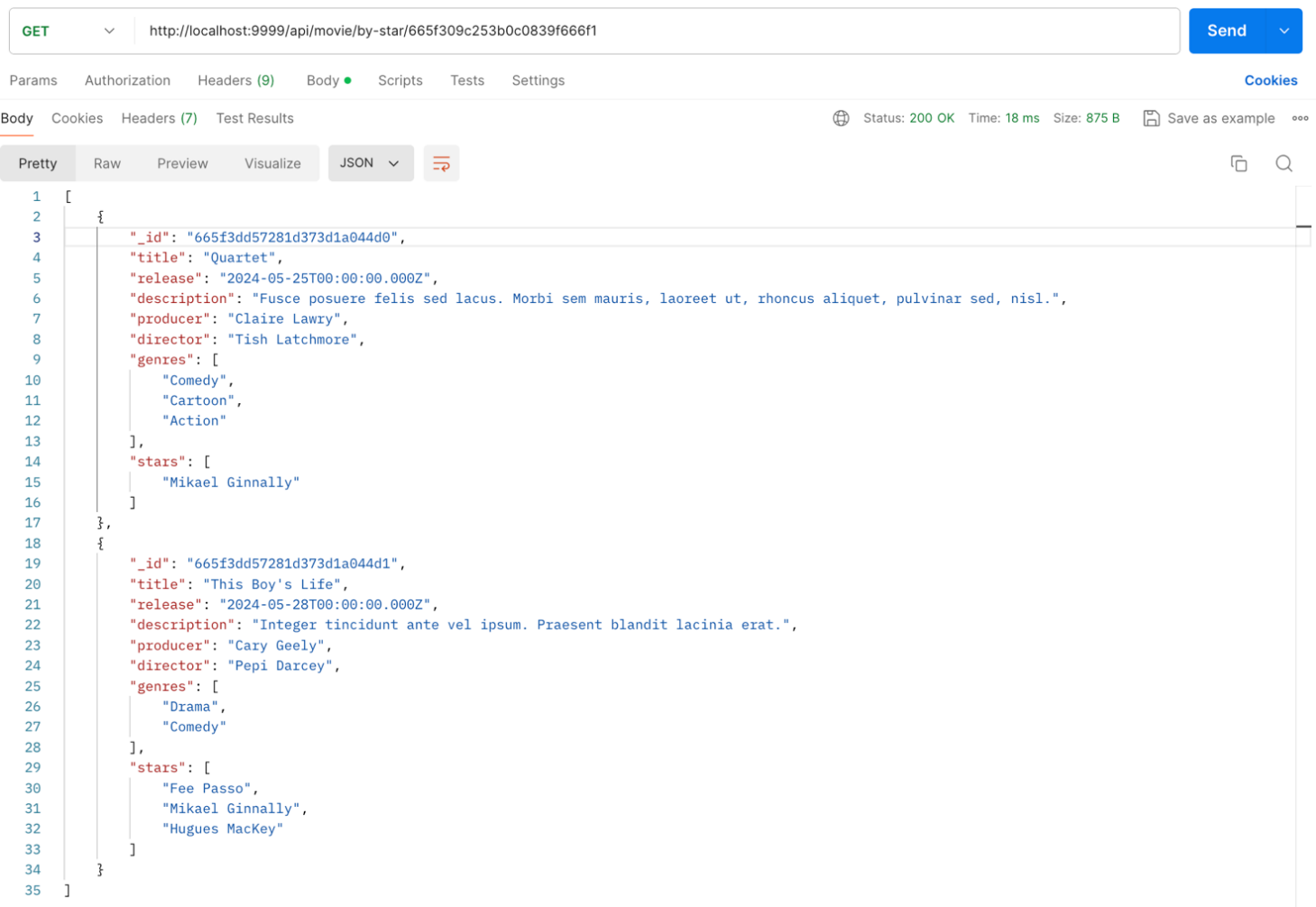
* Returns an array containing all the movies from the database.
* Display correct information of fields: **director** and **producer**.
* Display correct information of the **stars** field.

­­­­­

­­­­**Question 3**.

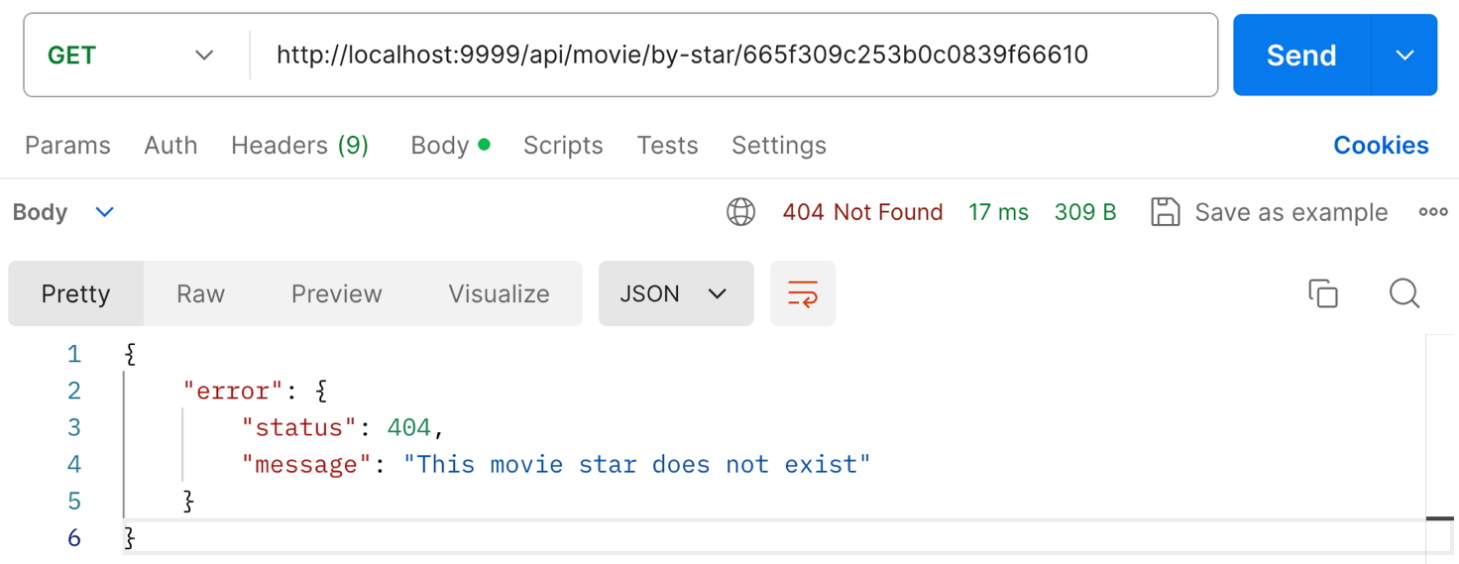
List all the movies that the star has played in as shown in *Figure 6*, with the following API endpoint parameters:

* URL: [http://localhost:9999/api/movie/by-star/**:starId**](http://localhost:9999/api/movie/by-star/:starId)
* Method: GET
* Example:



*Figure 6 - Movies that the star with \_id = “****665f309c253b0c0839f666f1****” has participated in*

* Return all movies as required by the question.
* Display correct information of fields: **director** and **producer**.
* Display correct information of the **stars** field.
* If the star’s **\_id** is incorrect, return an error as shown in *Figure 7*.



*Figure 7 - The star’s \_id is incorrect*

­­­­