

DATE : 09/09/2024

API Mini Project

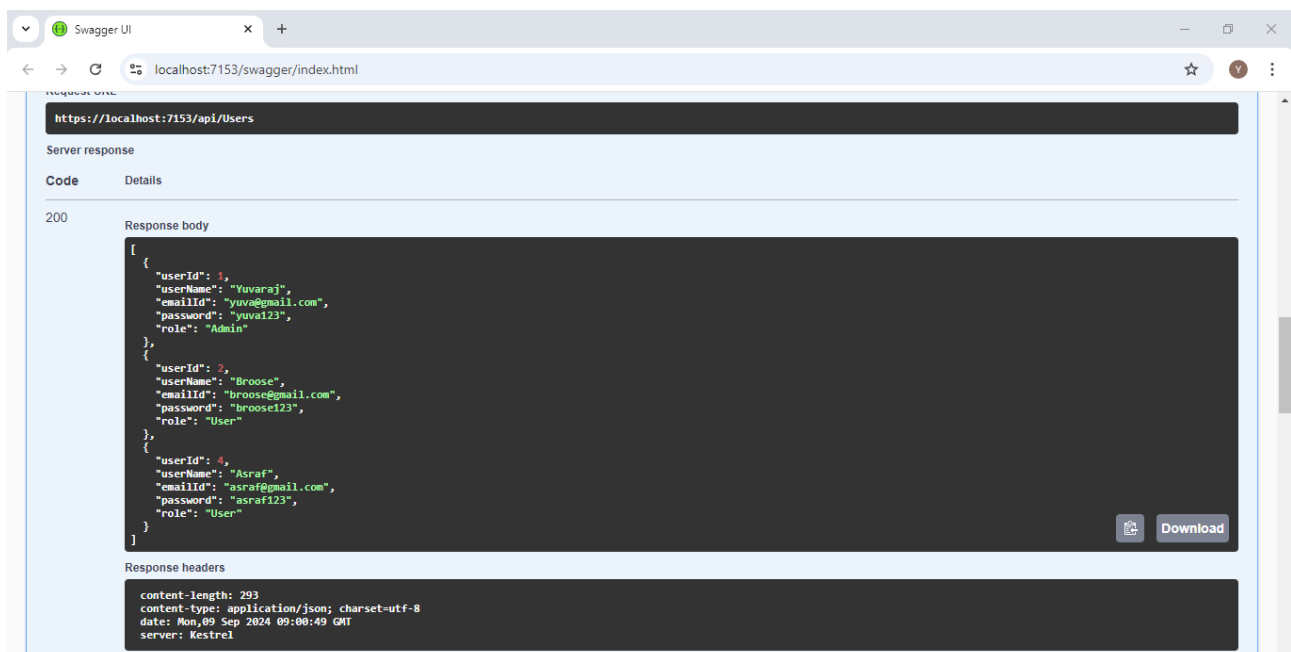
A fitness app has hired your company to build a Web API using Entity Framework's database-first approach to manage workout plans and their associated exercises by performing CRUD, and you need to implement filtering and counting operations for exercise difficulty and workout plan popularity.

Output:

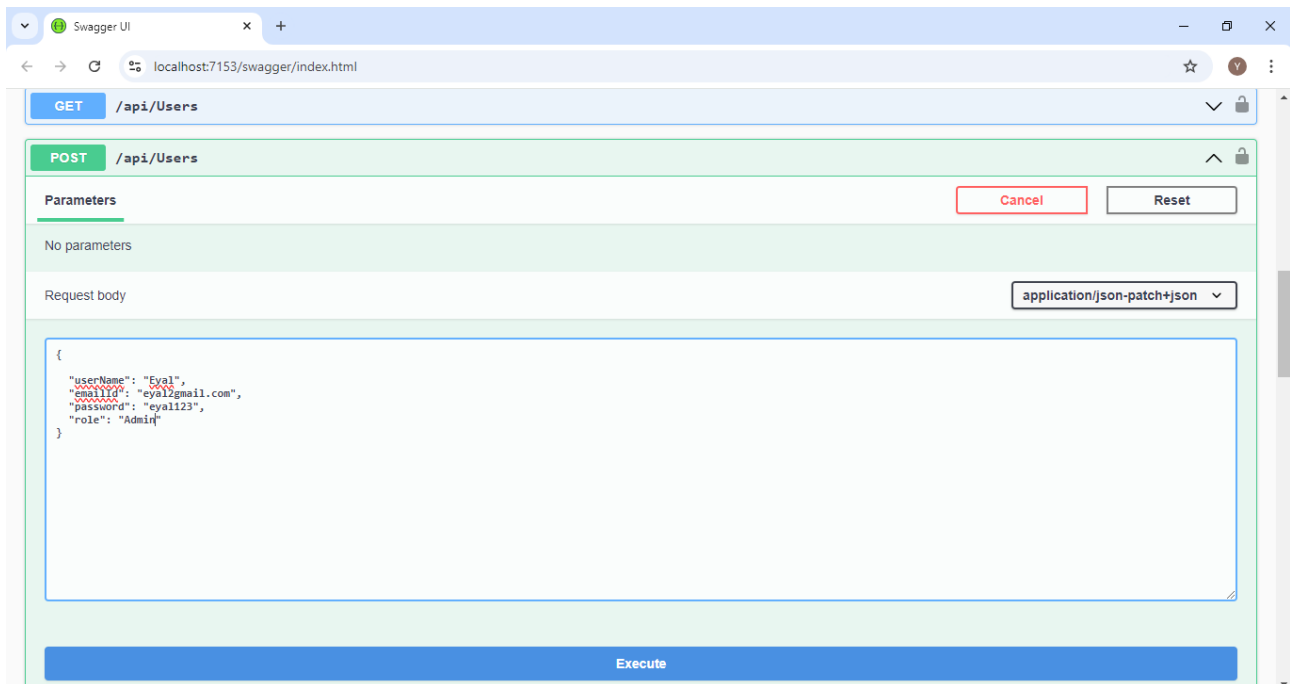


User Controller:

Get:



Post:

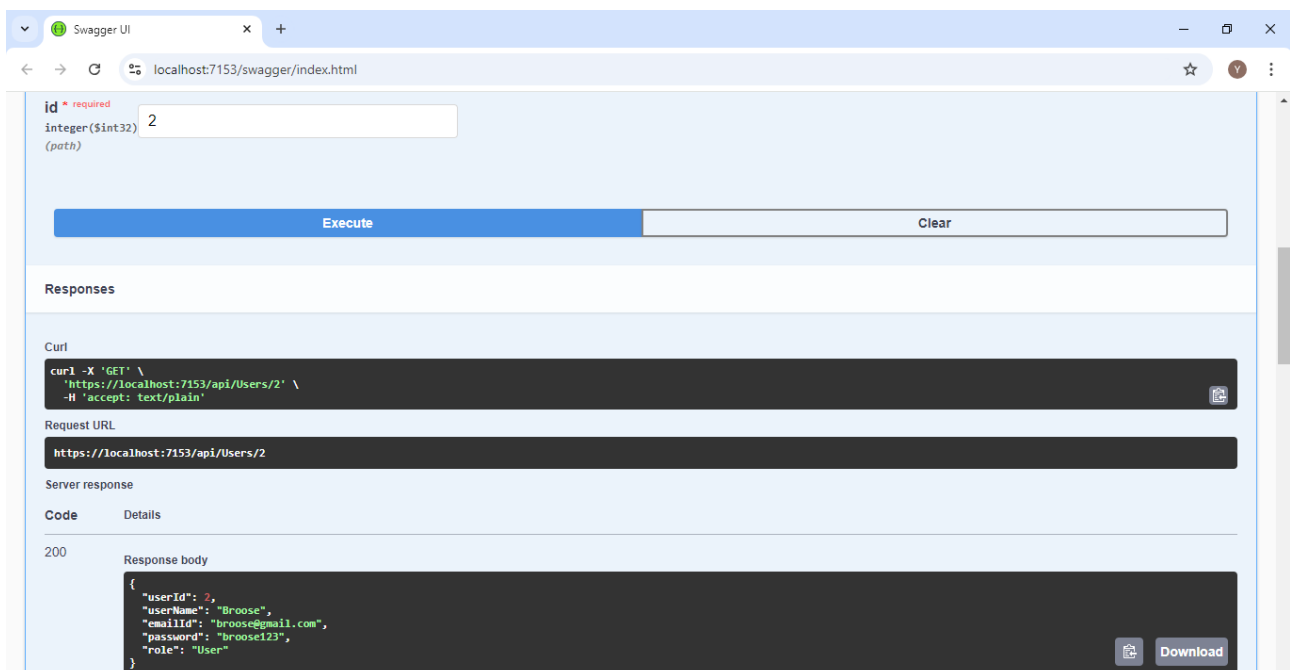


The image shows the Swagger UI interface for a POST endpoint at `/api/Users`. The interface is in a browser window with the address `localhost:7153/swagger/index.html`. The endpoint is selected, and the 'Parameters' section shows 'No parameters'. The 'Request body' section is set to `application/json-patch+json` and contains a JSON object:

```
{  "userName": "Eval",  "emailId": "eval2@gmail.com",  "password": "eval123",  "role": "Admin"}
```

. At the bottom, there is a large blue 'Execute' button.

Get By ID:



The image shows the Swagger UI interface for a GET endpoint at `/api/Users/2`. The 'id' parameter is required and has a value of '2' entered in the input field. Below the input field are 'Execute' and 'Clear' buttons. The 'Responses' section is expanded, showing a 'Curl' command:

```
curl -X 'GET' \  'https://localhost:7153/api/Users/2' \  -H 'accept: text/plain'
```

 The 'Request URL' is `https://localhost:7153/api/Users/2`. The 'Server response' section shows a '200' status code and a 'Response body' containing a JSON object:

```
{  "userId": 2,  "userName": "Broose",  "emailId": "broose@gmail.com",  "password": "broose123",  "role": "User"}
```

 A 'Download' button is visible next to the response body.

Put:

Swagger UI interface showing the PUT /api/Users/{id} endpoint. The 'id' parameter is set to 2. The request body is a JSON object with fields: userId, userName, emailId, password, and role.

Parameters

Name	Description
id * required	integer(\$int32) (path)

Request body

```
{  "userId": 0,  "userName": "string",  "emailId": "string",  "password": "string",  "role": "string"}
```

Delete:

Swagger UI interface showing the DELETE /api/Users/{id} endpoint. The 'id' parameter is set to 3. The 'Execute' button is visible.

Parameters

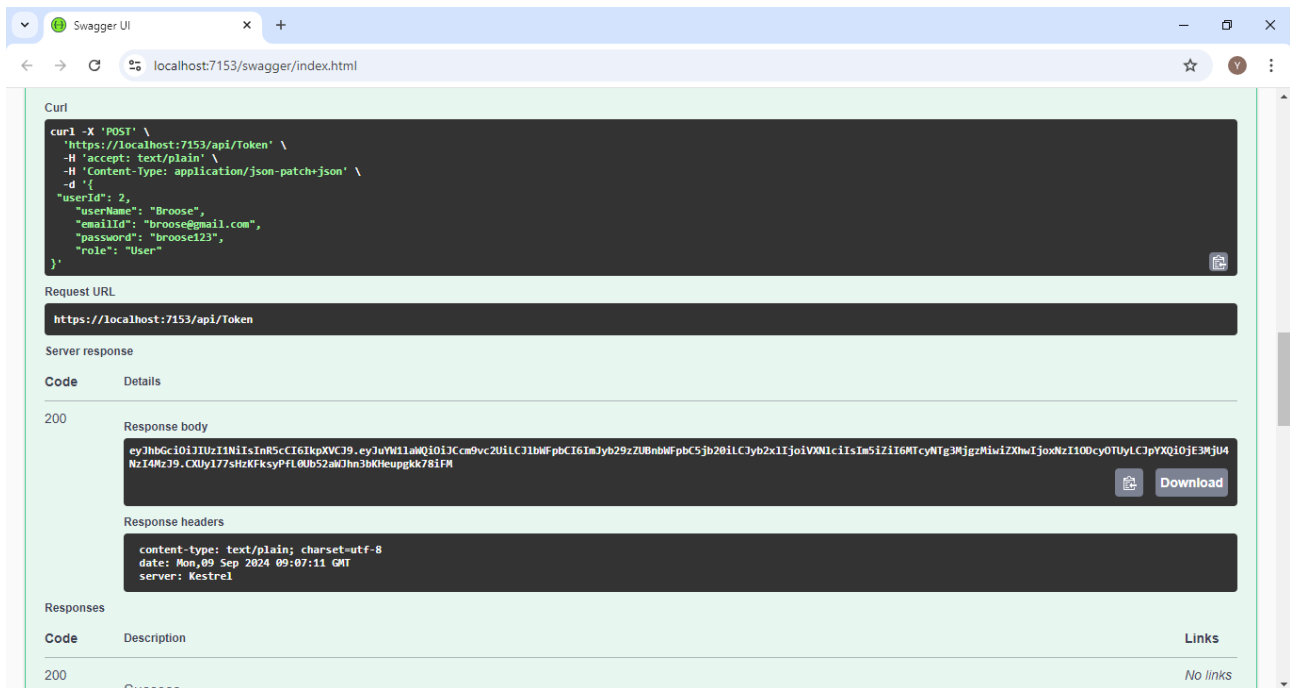
Name	Description
id * required	integer(\$int32) (path)

Execute

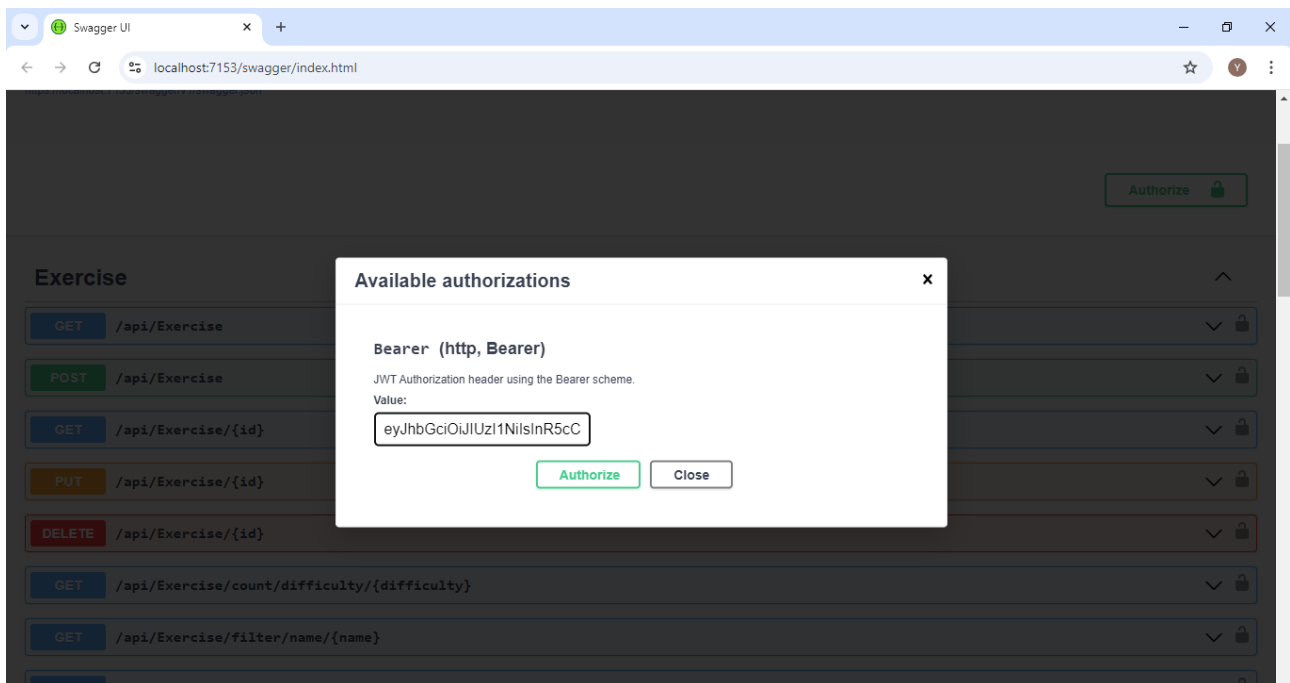
Responses

Code	Description	Links
200	Success	No links

Generate Token for User:



Authorize:



WorkoutPlan Controller:

In this controller Get,GetById and filter operations only allow to accessed by User.

Get:

The screenshot shows the Swagger UI interface for the endpoint `https://localhost:7153/api/WorkoutPlans`. The response status is 200. The response body is a JSON array of workout plans:

```
[
  {
    "workoutPlanId": 2,
    "name": "Beginner Plan",
    "description": "A plan for beginners",
    "popularity": 5,
    "exercises": [
      {
        "exerciseId": 1,
        "name": "Push Up",
        "difficulty": 2,
        "description": "A basic push-up exercise",
        "workoutPlanId": 2
      },
      {
        "exerciseId": 3,
        "name": "Pull Up",
        "difficulty": 4,
        "description": "A basic pull-up exercise",
        "workoutPlanId": 2
      }
    ]
  },
  {
    "workoutPlanId": 3,
    "name": "Intermediate Plan",
    "description": "A plan for intermediates",
    "popularity": 10
  }
]
```

GetById:

The screenshot shows the Swagger UI interface for the endpoint `https://localhost:7153/api/WorkoutPlans/2`. The input field for the path parameter `id` (required, integer) is set to 2. The response status is 200. The response body is a JSON object representing a single workout plan:

```
{
  "workoutPlanId": 2,
  "name": "Beginner Plan",
  "description": "A plan for beginners",
  "popularity": 5,
  "exercises": null
}
```

GetWorkoutPlansCountByPopularity :

The screenshot shows a web browser window with the address bar displaying 'localhost:7153/swagger/index.html'. The Swagger UI interface is visible, showing a form for the 'popularity' endpoint. The form has a text input field with the value '5'. Below the form is an 'Execute' button. The 'Responses' section shows the request URL 'https://localhost:7153/api/WorkoutPlans/count/popularity/5' and the server response code '200'. The response body is partially visible, showing the number '1'.

GetWorkoutPlansByName:

The screenshot shows the Swagger UI interface for a REST API. The top bar indicates the URL is localhost:7153/swagger/index.html. The main area displays a GET request to /api/WorkoutPlans/filter/name/Beginner%20Plan. The 'Name' input field is filled with 'Beginner Plan'. Below the input field, there are 'Execute' and 'Clear' buttons. The 'Responses' section shows the curl command and the request URL. The server response is displayed in a table with columns for 'Code' and 'Details'.

Code	Details
200	<pre>[{"workoutPlanId": 2, "name": "Beginner Plan", "description": "plan for beginners"}]</pre>

If workoutplan name not found :

The screenshot displays the Swagger UI for a REST API. The top navigation bar shows the Swagger logo and the URL 'localhost:7153/swagger/index.html'. The main area is divided into two sections: 'Request' and 'Responses'.

In the 'Request' section, the 'Name' field is required and has the value 'Sample'. Below the input field are two buttons: 'Execute' (highlighted in blue) and 'Clear'.

The 'Responses' section shows the result of the request. It indicates a 404 status code with the message 'Error: response status is 404'. The response body is 'WorkoutPlan Name not found'.

GetWorkoutPlansByPopularity:

[illegible]

Get:



The screenshot shows the Swagger UI interface in a web browser. The top bar indicates the URL is `localhost:7153/swagger/index.html`. The main content area displays the details for the `POST https://localhost:7153/api/Exercise` endpoint. Under the "Server response" section, a `403` status code is shown with the message "Error: response status is 403". Below this, the "Response headers" are listed: `content-length: 0`, `date: Mon, 09 Sep 2024 09:26:23 GMT`, and `server: Kestrel`. The "Responses" table at the bottom is currently empty.

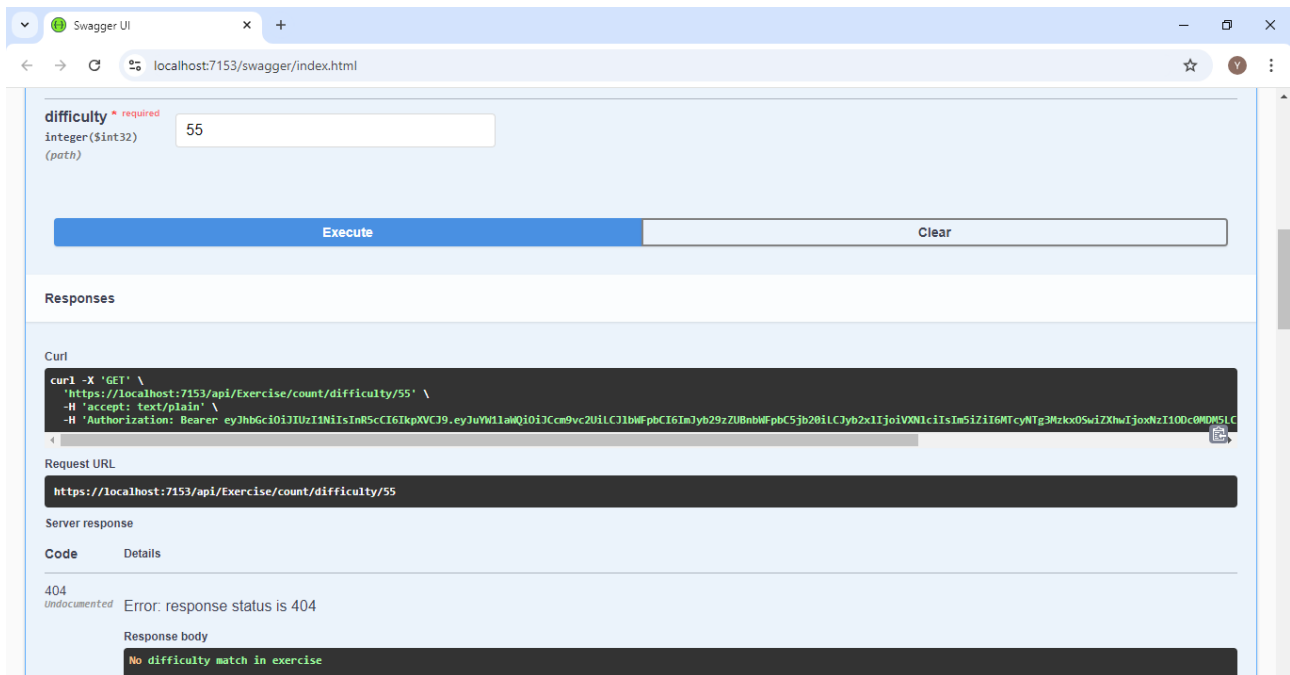
GetById:

[illegible]

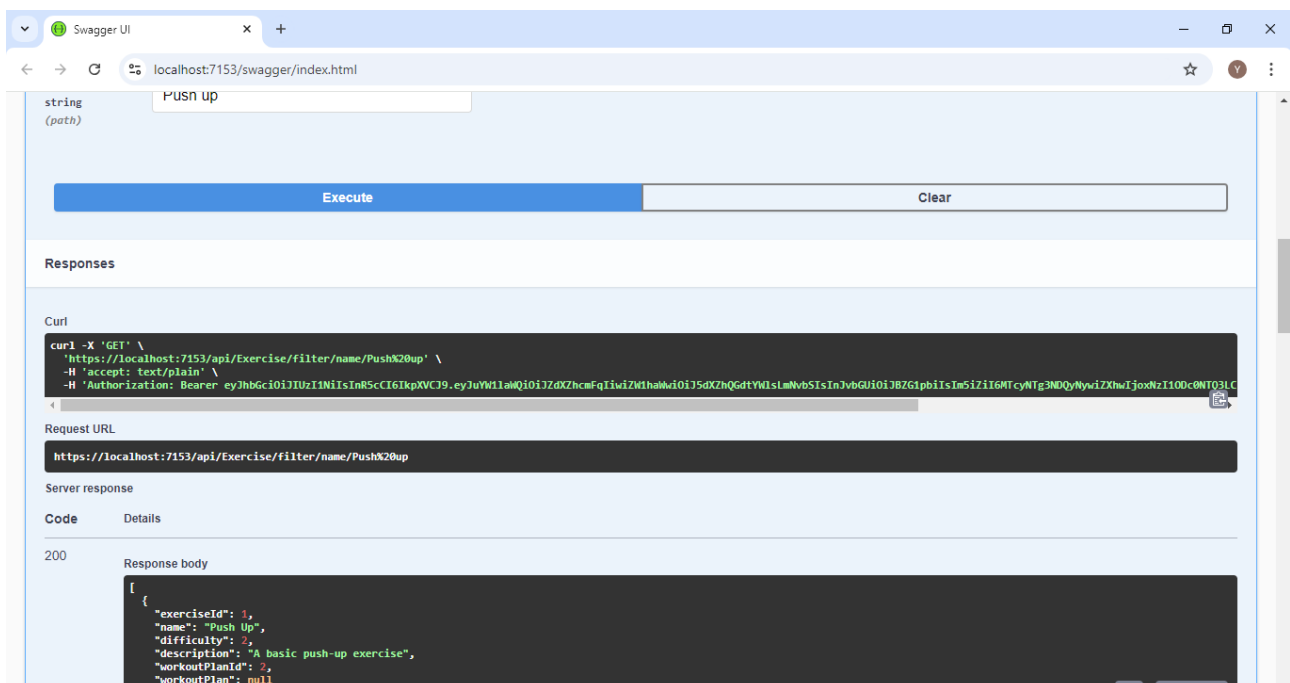
GetExercisesCountByDifficulty:

[illegible]

If Difficulty not match :



GetExercisesByName



ByWorkoutPlanId:

[illegible]