

27 de Noviembre del 2021

DOCUMENTACION CHALLENGE BACKEND JAVA SPRING BOOT (API)



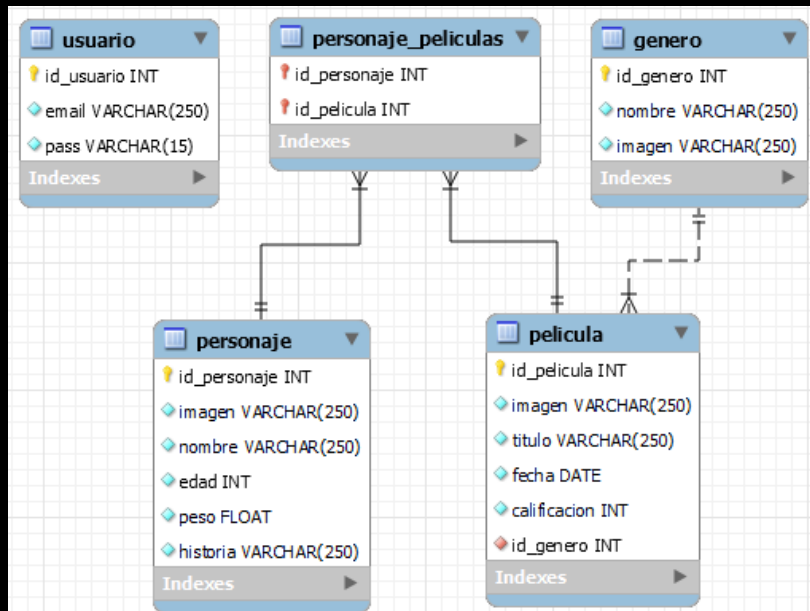
ELABORADO POR CORTES NAVA MARCO ANTONIO

MODELO DE BASE DE DATOS

Base de datos: MySQL

Para poder usar la API es necesario ejecutar el archivo "Disney.sql" dado que este tiene el modelo de base de datos que requiere la API para poder almacenar los datos.

Al ejecutar el archivo se creara la base de datos llamada "dinsey" que tendrá las siguientes tablas:



API

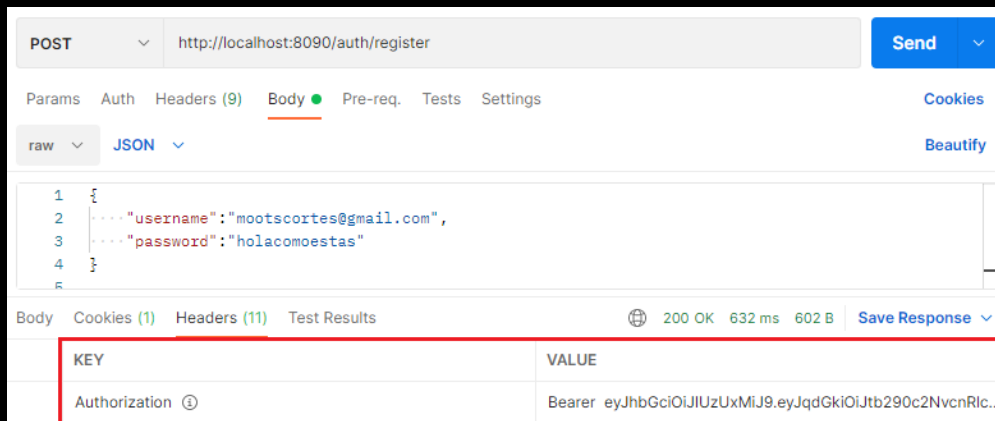
Configuración previa

Deberá abrir el archivo "application.properties" del proyecto y agregará su usuario de MySQL y su contraseña, esto para poder realizar la conexión con la base de datos.

```
1 logging.pattern.dateformat=hh:mm
2 server.port=8090
3 #MySQLConnection
4 spring.datasource.url=jdbc:mysql://localhost:3306/disney
5 spring.datasource.username=
6 spring.datasource.password=
```

Registro de Usuarios

Para realizar el registro de un usuario usará el endpoint /auth/register, con un metodo post, pasando un objeto de tipo usuario el cual tiene de atributos username y password, al hacer la petición la API devolverá un token para la sesión además de que llegará un email al correo ingresado.



POST http://localhost:8090/auth/register

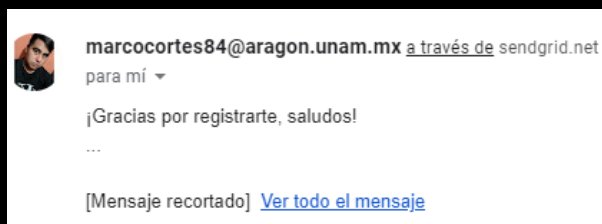
Params Auth Headers (9) Body Pre-req. Tests Settings

raw JSON

```
1 {
2   "username": "mootscortes@gmail.com",
3   "password": "holacomostas"
4 }
```

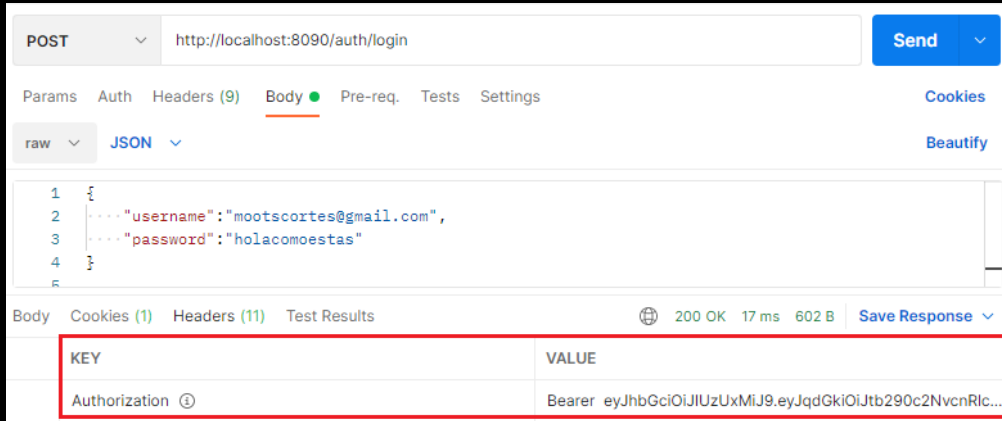
Body Cookies (1) Headers (11) Test Results 200 OK 632 ms 602 B Save Response

KEY	VALUE
Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJqdGkiOiJtb290c2NvcnRlc...

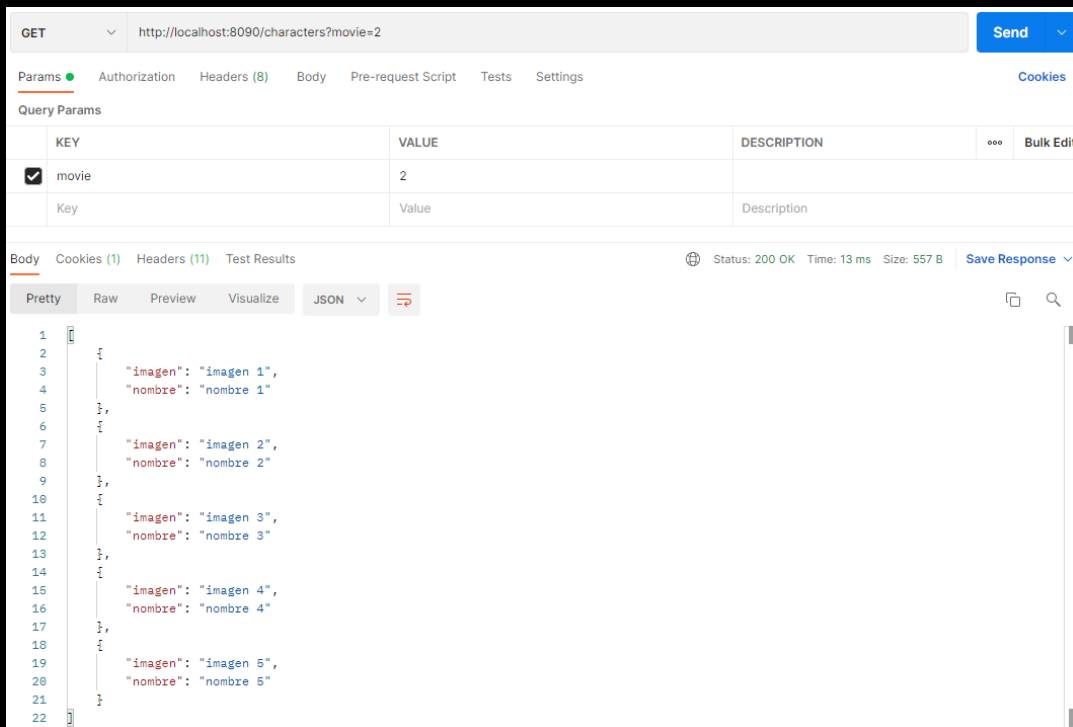


Login de Usuarios

Para el login de usuarios se accederá al endpoint /auth/login y se hará lo mismo que en la petición anterior, se enviará un objeto de tipo usuario y al recibir la petición la API generará el token.



Para obtener acceso a los endpoints de los personajes se deberá pasar el token en la petición como un parámetro de los "headers" de la siguiente forma:

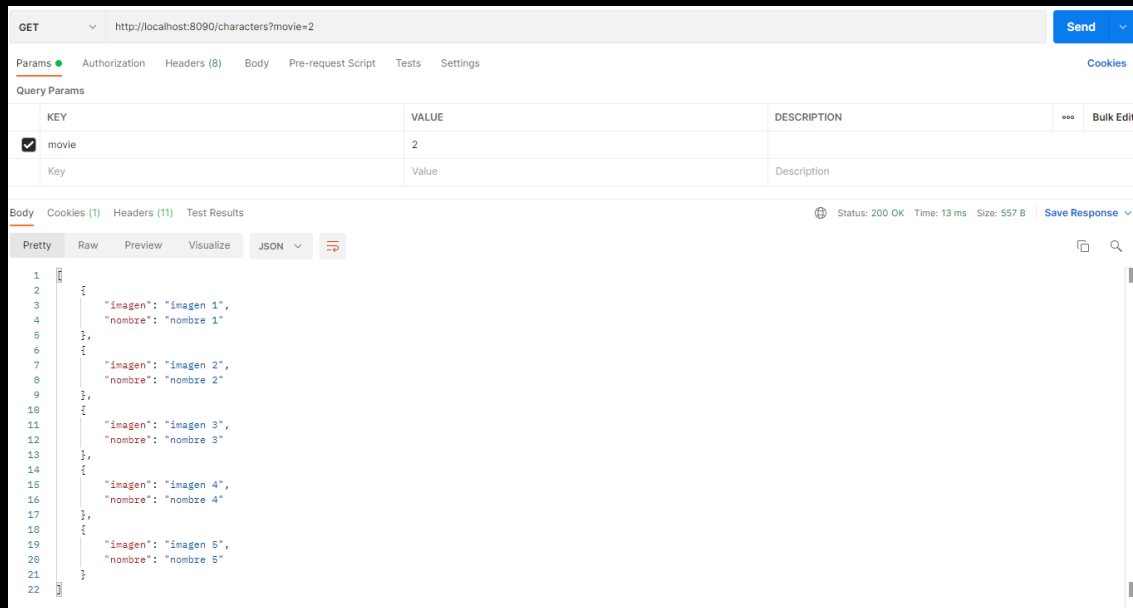


Personajes

Para obtener los datos de los personajes y realizar las operaciones del CRUD se accederá a los siguientes endpoints:

- Guardar personajes: /characters/save (POST)
- Lista de personajes: /characters (GET)
- Detalles de personaje: /characters/id (GET)
- Actualizar personaje: /characters/update/id (PUT)
- Eliminar personaje: /characters/delete/id (DELETE)

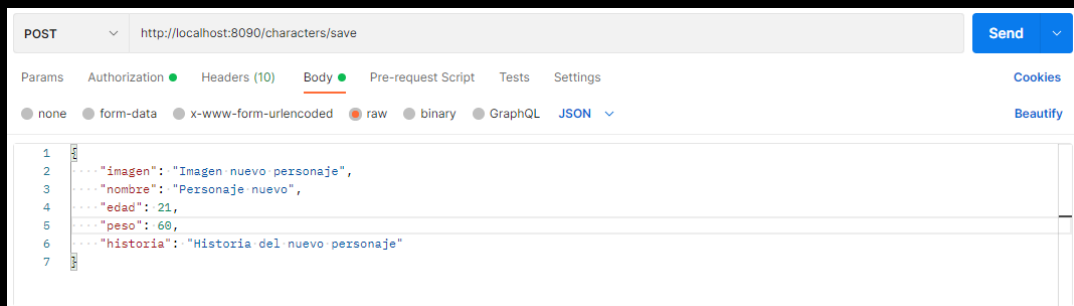
Por otra parte, los filtros de personajes quedarán en el endpoint de "/characters" simplemente pasando el parametro del filtro requerido y el valor del parametro. Por ejemplo:



Para personajes los únicos métodos que requieren un objeto JSON son los métodos de crear y actualizar, para estos el objeto que se les debe pasar debe tener los siguientes atributos:

- Imagen (String)
- Nombre (String)
- Edad (int)
- Peso (float)
- Historia (String)

Esto muestra en el siguiente ejemplo:



Y en la base de datos se muestra el nuevo dato ingresado:

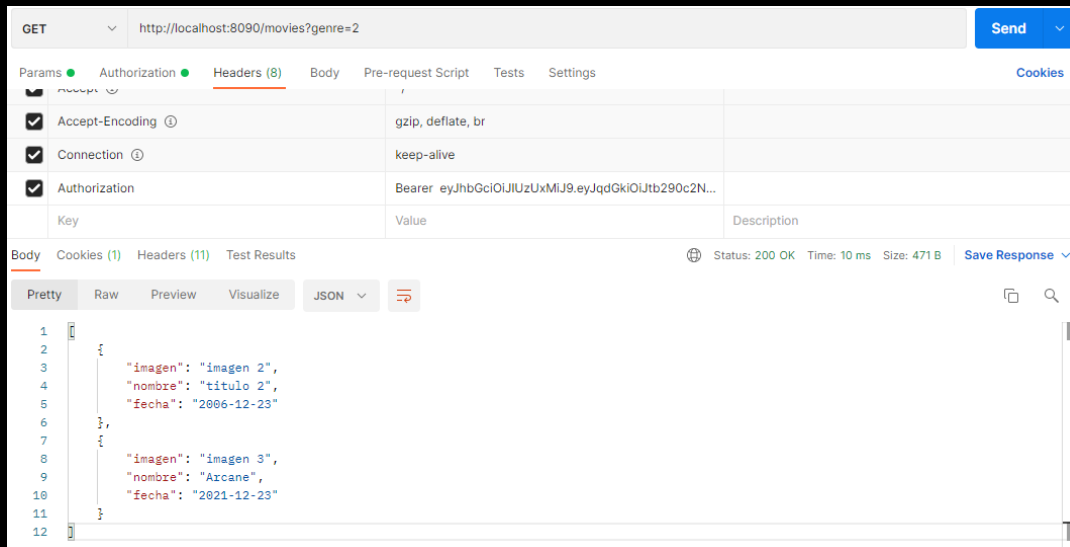
id_personaje	imagen	nombre	edad	peso	historia
1	imagen 1	nombre 1	25	43.54	historia personaje 1
2	imagen 2	nombre 2	53	54.2	historia personaje 2
3	imagen 3	nombre 3	32	88.2	historia personaje 3
4	imagen 4	nombre 4	19	69.72	historia personaje 4
5	imagen 5	nombre 5	44	76.4	historia personaje 5
6	Imagen nuevo personaje	Personaje nuevo	21	60	Historia del nuevo personaje

Películas

Para obtener los datos de las películas y realizar las operaciones del CRUD se accederá a los siguientes endpoints:

- Guardar películas: /movies/save (POST)
- Lista de películas: /movies (GET)
- Detalles de película: /movies/id (GET)
- Actualizar película: /movies/update/id (PUT)
- Eliminar películas: /movies/delete/id (DELETE)

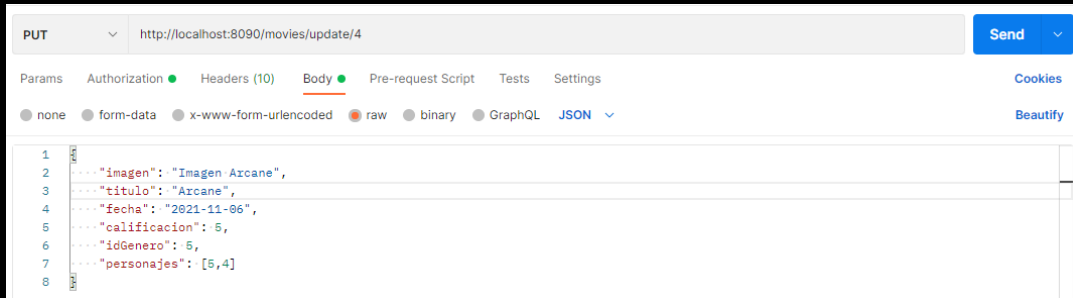
Finalmente, para los filtros de películas, estos serán realizados en el endpoint de "/películas" simplemente pasando el parametro del filtro requerido y el valor del parametro. Por ejemplo:



Para películas los únicos métodos que requieren un objeto JSON son los métodos de crear y actualizar, para estos el objeto que se les debe pasar debe tener los siguientes atributos:

- Imagen (String)
- Título (String)
- Fecha (String)
- Calificación (int)
- ID género (int)
- Personajes (Lista de enteros)

Esto muestra en el siguiente ejemplo:



Y en la base de datos se muestra el dato actualizado en la tabla película y personaje_peliculas:

id_pelicula	imagen	titulo	fecha	calificacion	id_genero
1	imagen 1	titulo 1	1989-12-23	4	1
2	imagen 2	titulo 2	2006-12-23	3	2
3	imagen 3	Arcane	2021-12-23	5	2
4	Imagen Arcane	Arcane	2021-11-06	5	5
5	imagen 5	titulo 5	1977-12-23	5	5

id_personaje	id_pelicula
1	1
2	1
1	2
2	2
3	2
3	3
4	3
5	3
4	4
5	4
1	5
2	5
5	5