

Documentación API Biblioteca

Daniel Hernández Gómez

Enlace a github: <https://github.com/Danonexe/Proyecto-API-REST-Segura-Biblioteca>

Introducción

Este sistema implementa un conjunto de funcionalidades para la gestión de usuarios, libros, editoriales y préstamos de libros. Utiliza Spring Boot y Kotlin, organizando las entidades, servicios y controladores para ofrecer una solución robusta y escalable.

Modelo de Datos

El sistema está construido alrededor de cuatro entidades principales: **Usuario**, **Editorial**, **Libro**, y **PrestamoLibro**. Estas entidades están vinculadas a través de relaciones definidas por JPA.

Entidades y Tablas

Usuario

Representa a los usuarios registrados en el sistema.

- **Tabla: usuarios**
- **Campos:**
 - **id** (Primary Key): Identificador único del usuario.
 - **username** (Unique, Not Null): Nombre de usuario.
 - **password** (Not Null): Contraseña cifrada.
 - **nombre** (Not Null): Nombre del usuario.
 - **apellidos** (Not Null): Apellidos del usuario.
 - **roles** (Not Null): Roles asignados al usuario.

Editorial

Representa las editoriales responsables de la publicación de libros.

- **Tabla: editoriales**
- **Campos:**
 - **id_editorial** (Primary Key): Identificador único de la editorial.

- **nombre** (Not Null): Nombre de la editorial.

Libro

Representa a los libros disponibles en el sistema.

- **Tabla: libros**
- **Campos:**
 - **id_libro** (Primary Key): Identificador único del libro.
 - **titulo** (Not Null): Título del libro.
 - **autor** (Not Null): Autor del libro.
 - **editorial** (Foreign Key): Relación con la tabla **editoriales**.
 - **fecha_de_publicacion** (Not Null): Fecha de publicación del libro.

PrestamoLibro

Representa el préstamo de un libro a un usuario.

- **Tabla: prestamo_libro**
- **Campos:**
 - **id** (Primary Key): Identificador único del préstamo.
 - **libro** (Foreign Key): Relación con la tabla **libros**.
 - **usuario** (Foreign Key): Relación con la tabla **usuarios**.
 - **devuelto**: Indica si el libro ha sido devuelto.
 - **fecha_prestamo**: Fecha en la que se realizó el préstamo.
 - **limite_prestamo**: Fecha límite para devolver el libro.

Servicios y Lógica de Negocio

UsuarioService

Gestiona las operaciones relacionadas con los usuarios.

- **Funciones principales:**
 - **registerUsuario**: Registra un nuevo usuario con contraseña cifrada.
 - **editUsuario**: Actualiza los datos de un usuario existente.
 - **getUsuarioPorId**: Obtiene un usuario por su ID.
 - **getAllUsuarios**: Lista todos los usuarios.
 - **deleteUsuarioPorId**: Elimina un usuario por su ID.

EditorialService

Gestiona las operaciones relacionadas con las editoriales.

- **Funciones principales:**

- **createEditorial:** Crea una nueva editorial verificando que no exista otra con el mismo nombre.
- **getAllEditoriales:** Lista todas las editoriales.
- **getEditorialByNombre:** Recupera una editorial por su nombre.
- **updateEditorial:** Actualiza los datos de una editorial existente.
- **deleteEditorial:** Elimina una editorial.
- **getLibrosByEditorial:** Lista los libros asociados a una editorial específica.

LibroService

Gestiona las operaciones relacionadas con los libros.

- **Funciones principales:**

- **createLibro:** Crea un libro verificando la unicidad relativa al título y la editorial.
- **getAllLibros:** Lista todos los libros.
- **getLibroById:** Obtiene un libro por su ID.
- **updateLibro:** Actualiza los datos de un libro existente.
- **deleteLibro:** Elimina un libro específico.

PrestamoLibroService

Gestiona los préstamos de libros.

- **Funciones principales:**

- **setPrestamo:** Crea un préstamo de libro verificando que no existan préstamos activos para el usuario y el libro.
- **marcarComoDevuelto:** Marca un préstamo como devuelto.
- **getTodosLosPrestamos:** Lista todos los préstamos registrados.
- **getPrestamoPorId:** Obtiene un préstamo por su ID.

TokenService

Maneja la generación de tokens JWT para autenticación.

- **Funciones principales:**

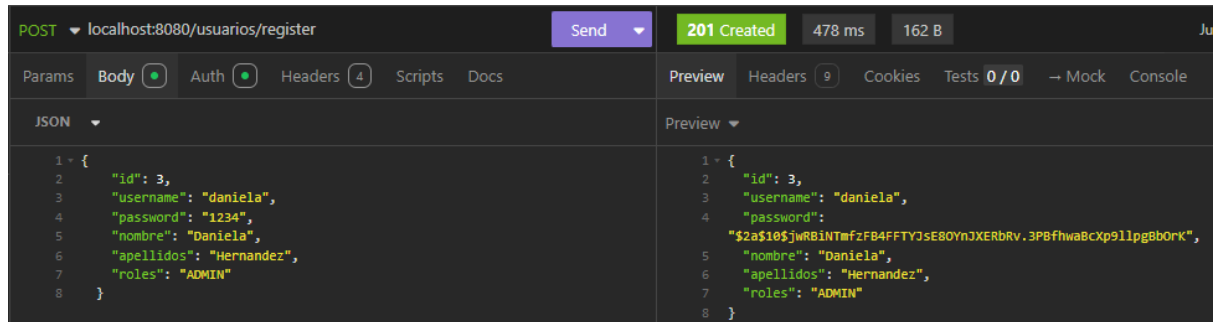
- **generarToken:** Genera un token JWT basado en los roles y datos del usuario autenticado.

Controladores

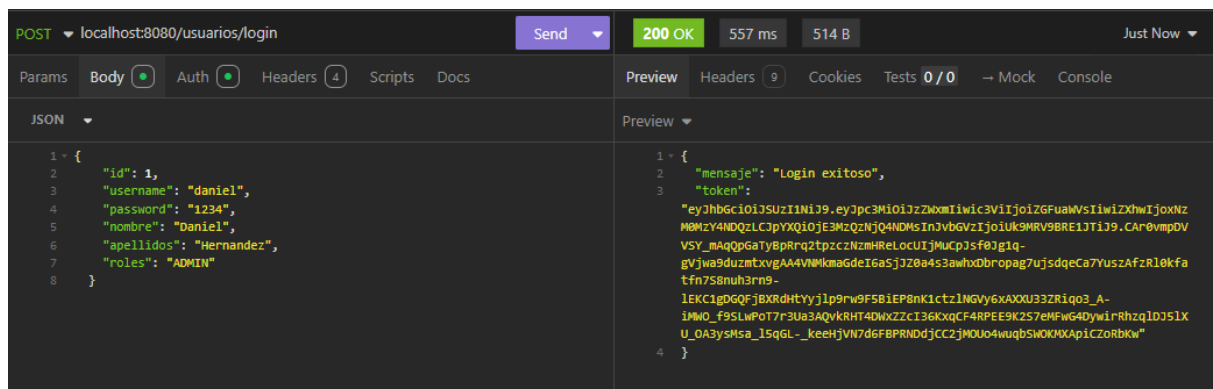
Los controladores exponen las funcionalidades a través de endpoints REST.

UserController

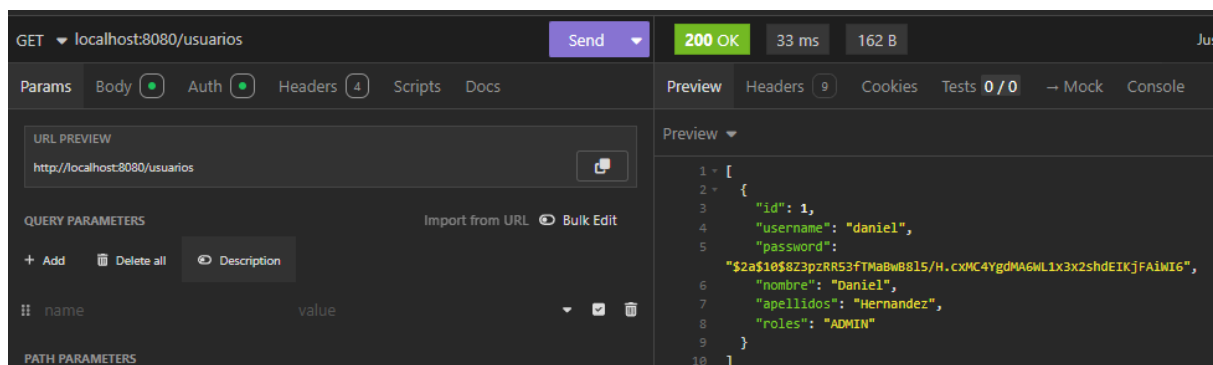
- **Endpoints:**
 - **POST /usuarios/register:** Registra un nuevo usuario.



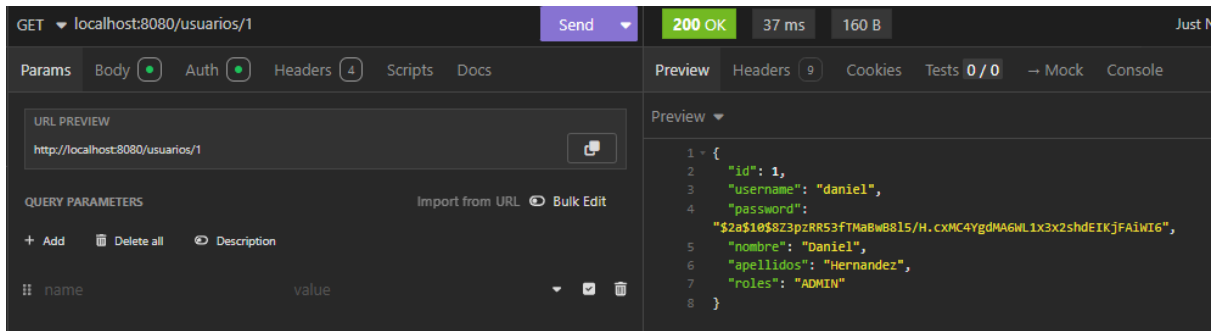
- **POST /usuarios/login:** Autentica un usuario y genera un token JWT.



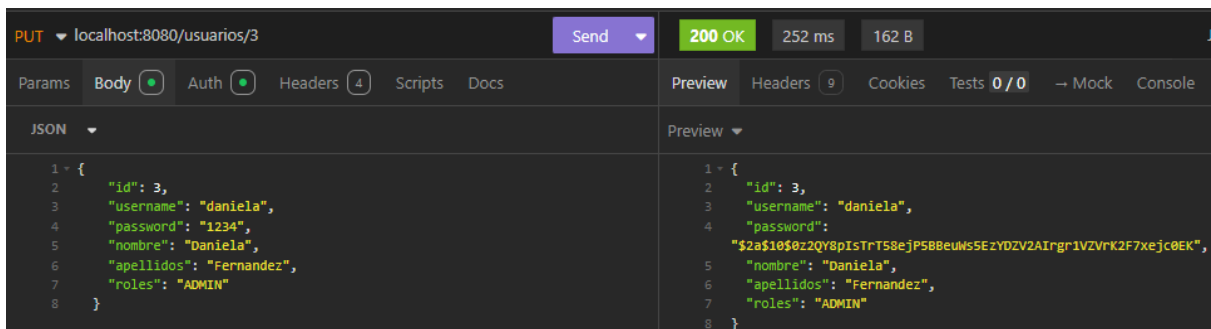
- **GET /usuarios:** Lista todos los usuarios.



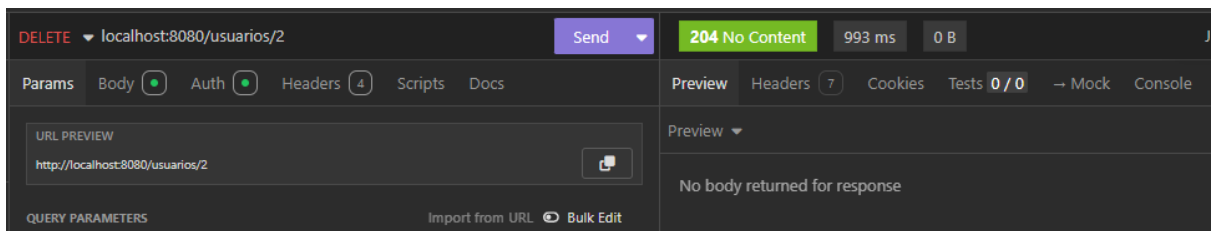
- **GET /usuarios/{id}**: Obtiene un usuario por su ID.



- **PUT /usuarios/{id}**: Actualiza un usuario.

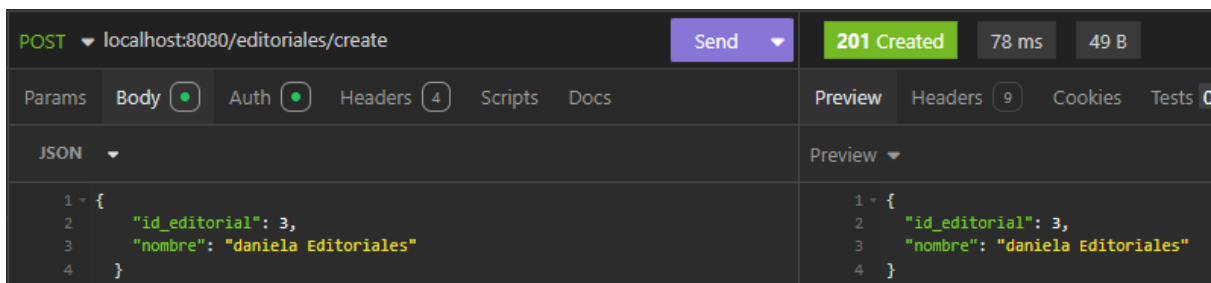


- **DELETE /usuarios/{id}**: Elimina un usuario.



EditorialController

- **Endpoints:**
 - **POST /editoriales/create**: Crea una nueva editorial.



- **GET /editoriales**: Lista todas las editoriales

GET localhost:8080/editoriales Send 200 OK 539 ms 412 B

Params Body Auth Headers Scripts Docs Preview Headers Cookies Tests

URL PREVIEW
http://localhost:8080/editoriales

QUERY PARAMETERS
+ Add - Delete all Description

name value

PATH PARAMETERS
Path parameters are url path segments that start with a colon ':' e.g. 'id'

Preview

```

1 [
2   {
3     "id_editorial": 1,
4     "nombre": "Penguin Random House"
5   },
6   {
7     "id_editorial": 2,
8     "nombre": "Planeta"
9   },
10  {
11    "id_editorial": 3,
12    "nombre": "Santillana"
13  },
14  {
15    "id_editorial": 4,
16    "nombre": "Anagrama"
17  },
18  {
19    "id_editorial": 5,
20    "nombre": "Alfaguara"
21  },
22 ]

```

- GET /editoriales/{nombre}: Obtiene una editorial por su nombre.

GET localhost:8080/editoriales/planeta Send 200 OK 123 ms 37 B

Params Body Auth Headers Scripts Docs Preview Headers Cookies

URL PREVIEW
http://localhost:8080/editoriales/planeta

QUERY PARAMETERS
Import from URL Bulk Edit

Preview

```

1 {
2   "id_editorial": 2,
3   "nombre": "Planeta"
4 }

```

- PUT /editoriales/{nombre}: Actualiza una editorial.

PUT localhost:8080/editoriales/daniela Editoriales Send 200 OK 57 ms 58 B

Params Body Auth Headers Scripts Docs Preview Headers Cookies Tests

JSON

```

1 {
2   "id_editorial": 3,
3   "nombre": "daniela Editoriales Editadas"
4 }

```

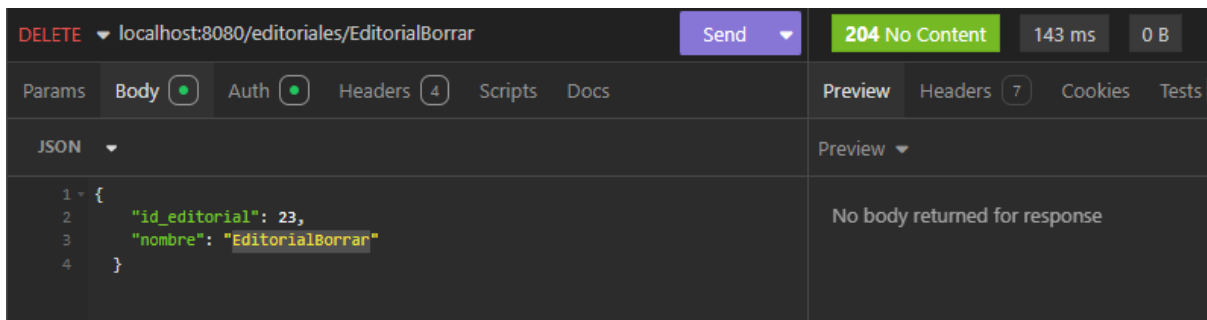
Preview

```

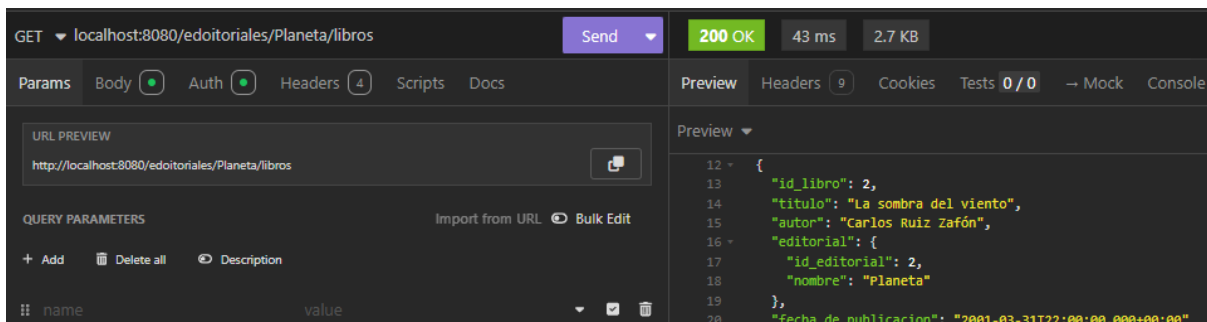
1 {
2   "id_editorial": 3,
3   "nombre": "daniela Editoriales Editadas"
4 }

```

- DELETE /editoriales/{nombre}: Elimina una editorial.

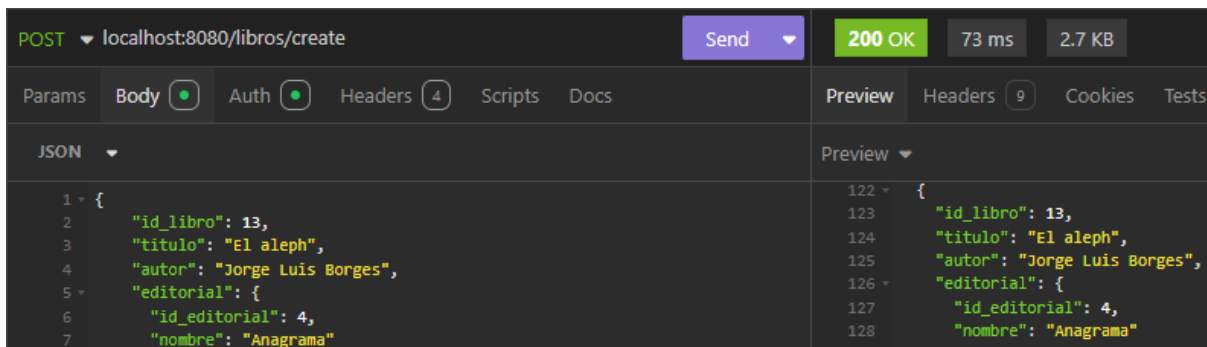


- **GET /editoriales/{nombreEditorial}/libros:** Lista los libros de una editorial.

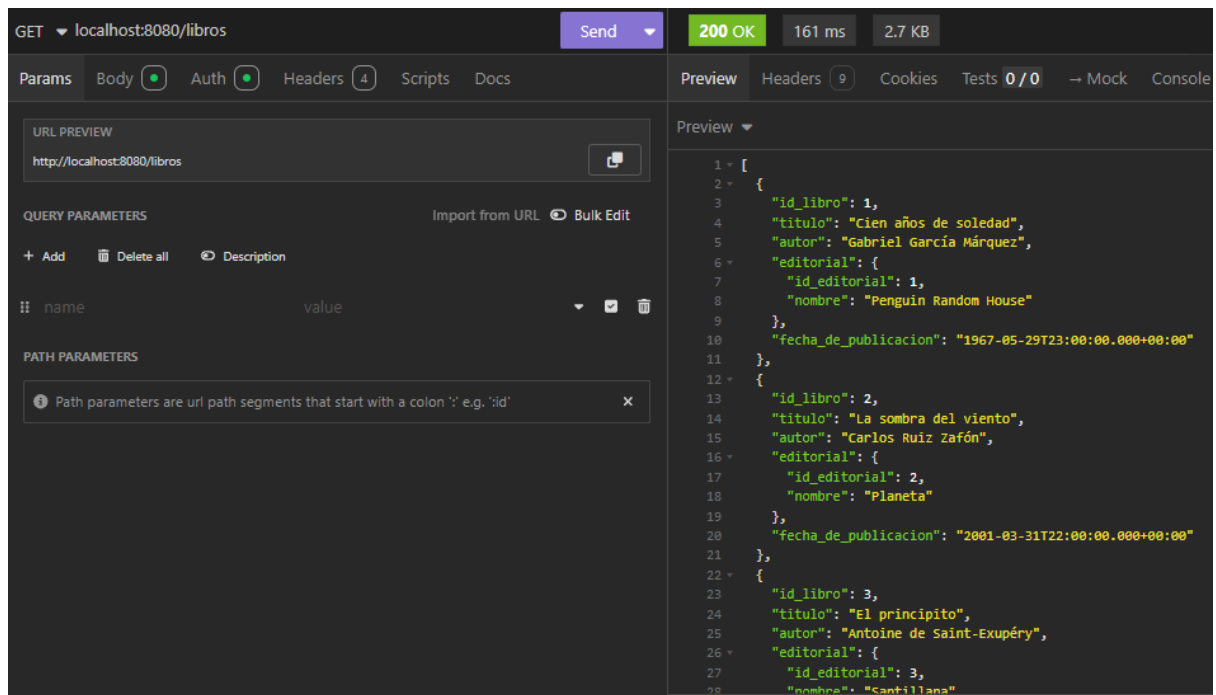


LibroController

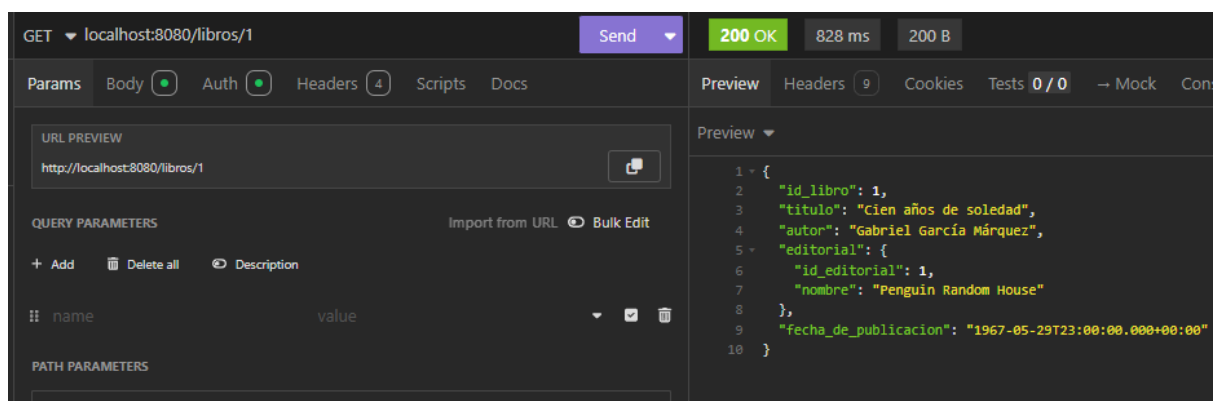
- **Endpoints:**
 - **POST /libros/create:** Crea un nuevo libro.



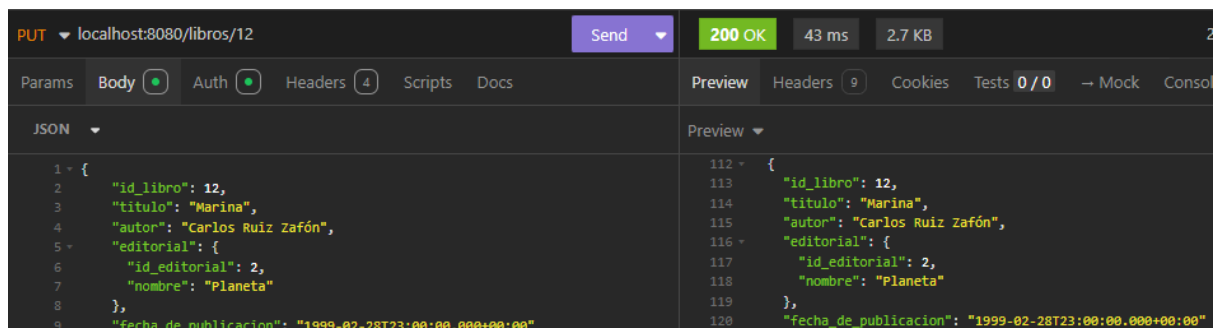
- **GET /libros:** Lista todos los libros.



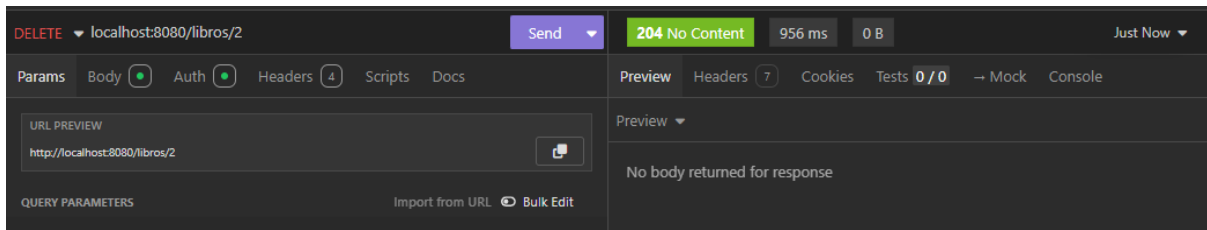
- GET /libros/{id}: Obtiene un libro por su ID.



- PUT /libros/{id}: Actualiza un libro.

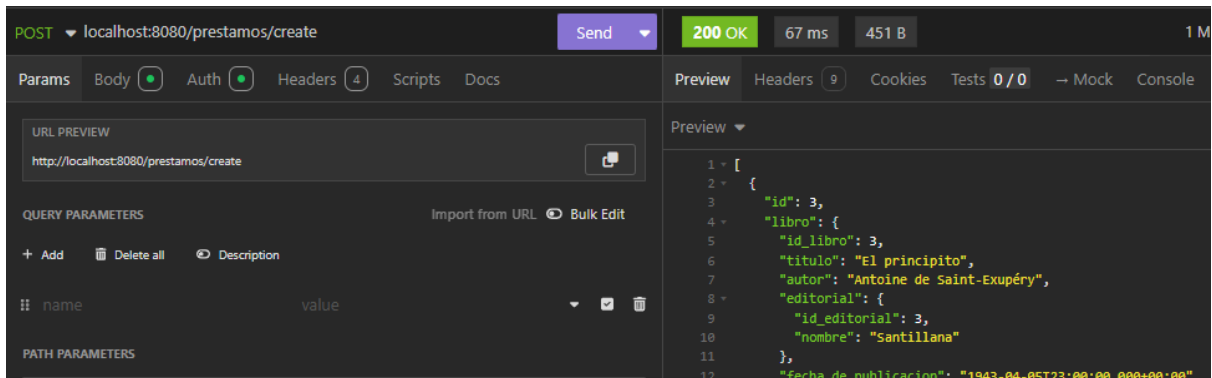


- DELETE /libros/{id}: Elimina un libro.

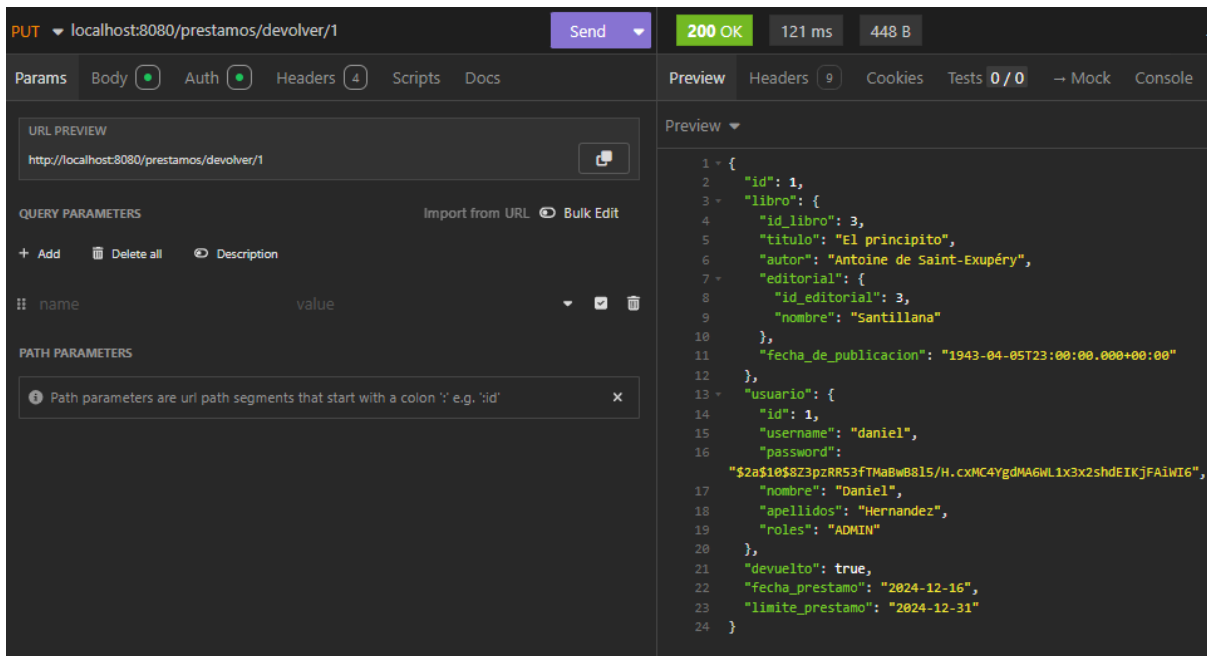


PrestamoLibroController

- Endpoints:
 - POST /prestamos/create: Crea un nuevo préstamo de libro.



- PUT /prestamos/devolver/{id}: Marca un préstamo como devuelto.



- GET /prestamos: Lista todos los préstamos.

GET localhost:8080/prestamos Send 200 OK 267 ms 451 B

Params Body Auth Headers 4 Scripts Docs

URL PREVIEW
http://localhost:8080/prestamos

QUERY PARAMETERS
+ Add Delete all Description

name	value
------	-------

PATH PARAMETERS
Path parameters are url path segments that start with a colon ':' e.g. 'id'

Preview

```
1 [
2 {
3   "id": 1,
4   "libro": {
5     "id_libro": 3,
6     "titulo": "El principito",
7     "autor": "Antoine de Saint-Exupéry",
8     "editorial": {
9       "id_editorial": 3,
10      "nombre": "Santillana"
11    },
12    "fecha_de_publicacion": "1943-04-05T23:00:00.000+00:00"
13  },
14  "usuario": {
15    "id": 1,
16    "username": "daniel",
17    "password":
18      "$2a$10$8Z3pzRR53fTmaBwB815/H.cxMC4YgdMA6WL1x3x2shdEIKjFAiWI6",
19    "nombre": "Daniel",
20    "apellidos": "Hernandez",
21    "roles": "ADMIN"
22  },
23  "devuelto": false,
24  "fecha_prestamo": "2024-12-16",
25  "limite_prestamo": "2024-12-31"
26 }
```

- GET /prestamos/{id}: Obtiene un préstamo por su ID.

GET localhost:8080/prestamos/1 Send 200 OK 34 ms 449 B

Params Body Auth Headers 4 Scripts Docs

URL PREVIEW
http://localhost:8080/prestamos/1

QUERY PARAMETERS
+ Add Delete all Description

name	value
------	-------

PATH PARAMETERS
Path parameters are url path segments that start with a colon ':' e.g. 'id'

Preview

```
1 {
2   "id": 1,
3   "libro": {
4     "id_libro": 3,
5     "titulo": "El principito",
6     "autor": "Antoine de Saint-Exupéry",
7     "editorial": {
8       "id_editorial": 3,
9       "nombre": "Santillana"
10    },
11    "fecha_de_publicacion": "1943-04-05T23:00:00.000+00:00"
12  },
13  "usuario": {
14    "id": 1,
15    "username": "daniel",
16    "password":
17      "$2a$10$8Z3pzRR53fTmaBwB815/H.cxMC4YgdMA6WL1x3x2shdEIKjFAiWI6",
18    "nombre": "Daniel",
19    "apellidos": "Hernandez",
20    "roles": "ADMIN"
21  },
22  "devuelto": false,
23  "fecha_prestamo": "2024-12-16",
24  "limite_prestamo": "2024-12-31"
25 }
```