GUÍA DE EJECUCIÓN Y FUNCIONAMIENTO

Computación en Internet II

GABRIEL RESTREPO
JUAN GONZÁLEZ
SEBASTIÁN ESCOBAR

A continuación, se presentan los resultados y las pruebas de los endpoints para cada entidad del modelo. En esta guía de ejecución se mostrará lo necesario para, en ese orden:

Autor:

(1) Crear un nuevo autor.
(2) Listar todos los autores.
(3) Obtener detalles de un autor específico.
(4) Actualizar un autor existente.
(5) Eliminar un autor

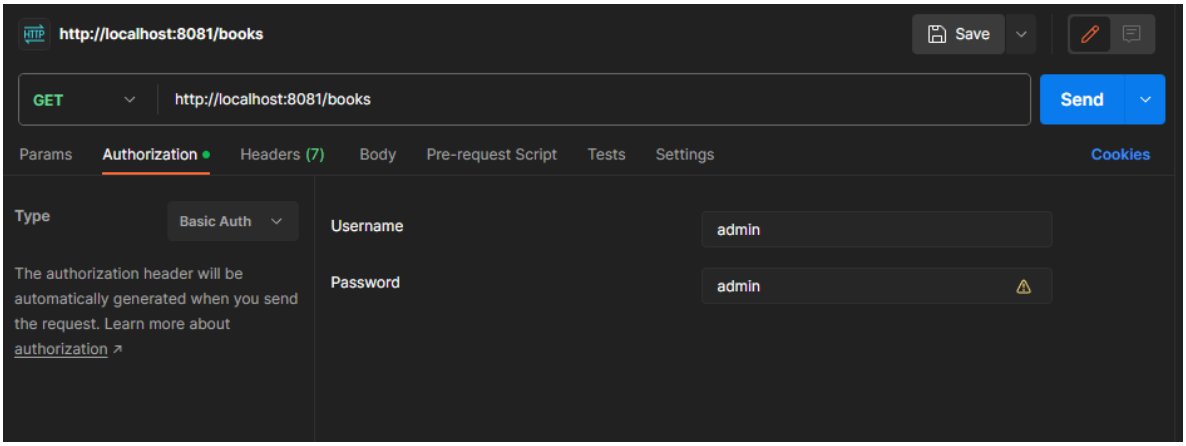
Libro:

(6) Crear un nuevo libro
(7) Listar todos los libros.
(8) Obtener detalles de un libro específico.
(9) Actualizar un libro existente.
(10) Eliminar un libro.

Autor:

(11) Listar los libros de un autor específico.

IMPORTANTE: Antes de realizar cualquier prueba con el Postman, es necesario configurarlo usando las credenciales mostradas a continuación, de lo contrario, no se podrá efectuar ninguna operación.



Nótese que debe seleccionarse la opción de autenticación “Basic Auth”.

AUTOR

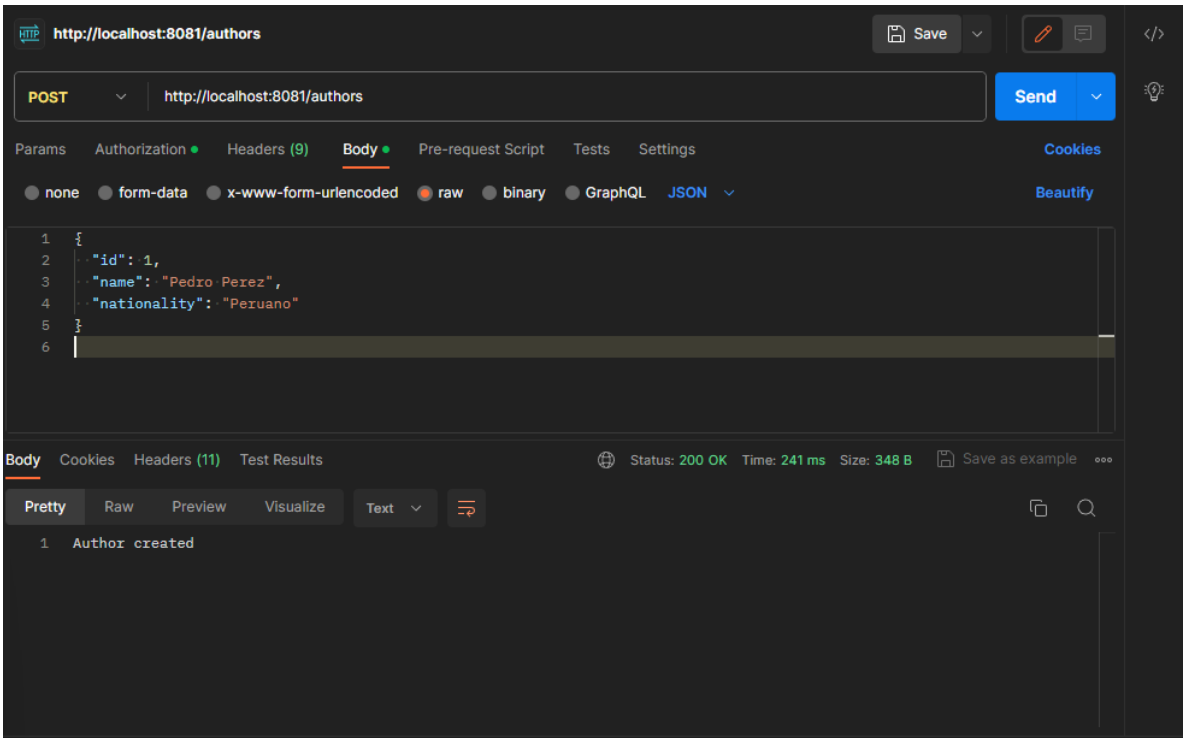
(1) POST /autores: Crear un nuevo autor.

POST: http://localhost:8081/authors

Body:

```
{
  "id": 1,
  "name": "Pedro Perez",
  "nationality": "Peruano"
}
```

Resultado:

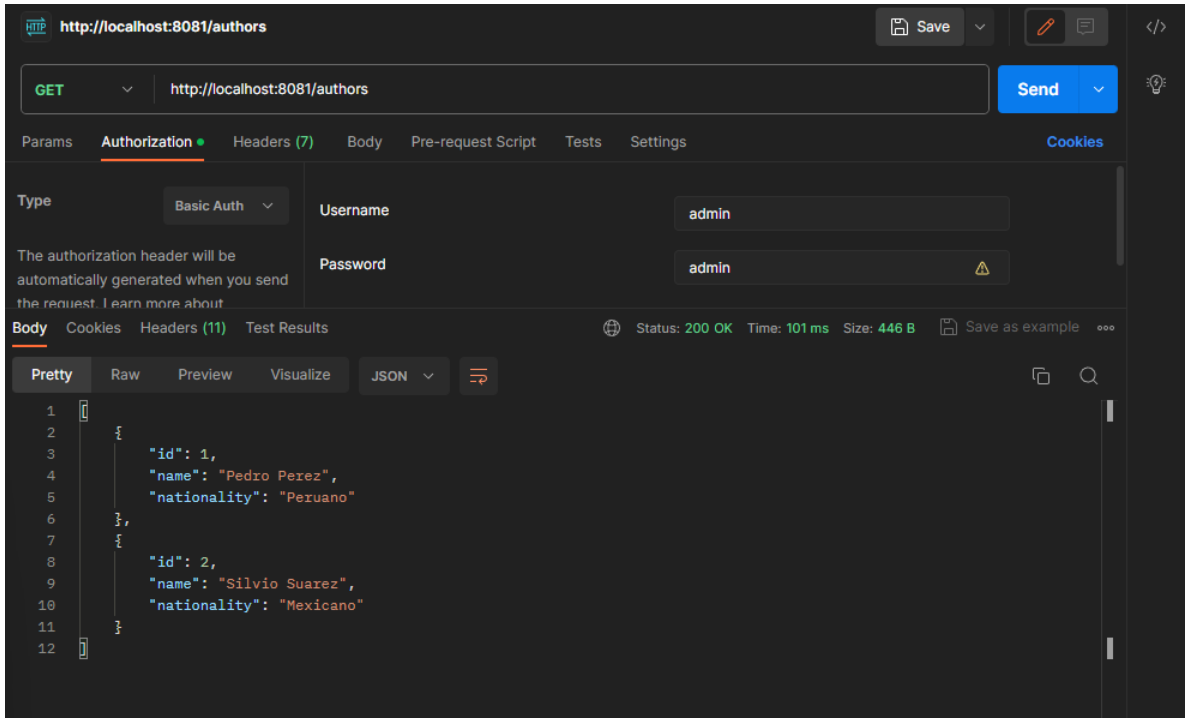


La inserción de este dato se verificará a continuación, haciendo uso del endpoint GET. Para mostrar que la aplicación retorna todos los autores, se creará otro autor.

(2) GET /autores: Listar todos los autores.

GET: http://localhost:8081/authors

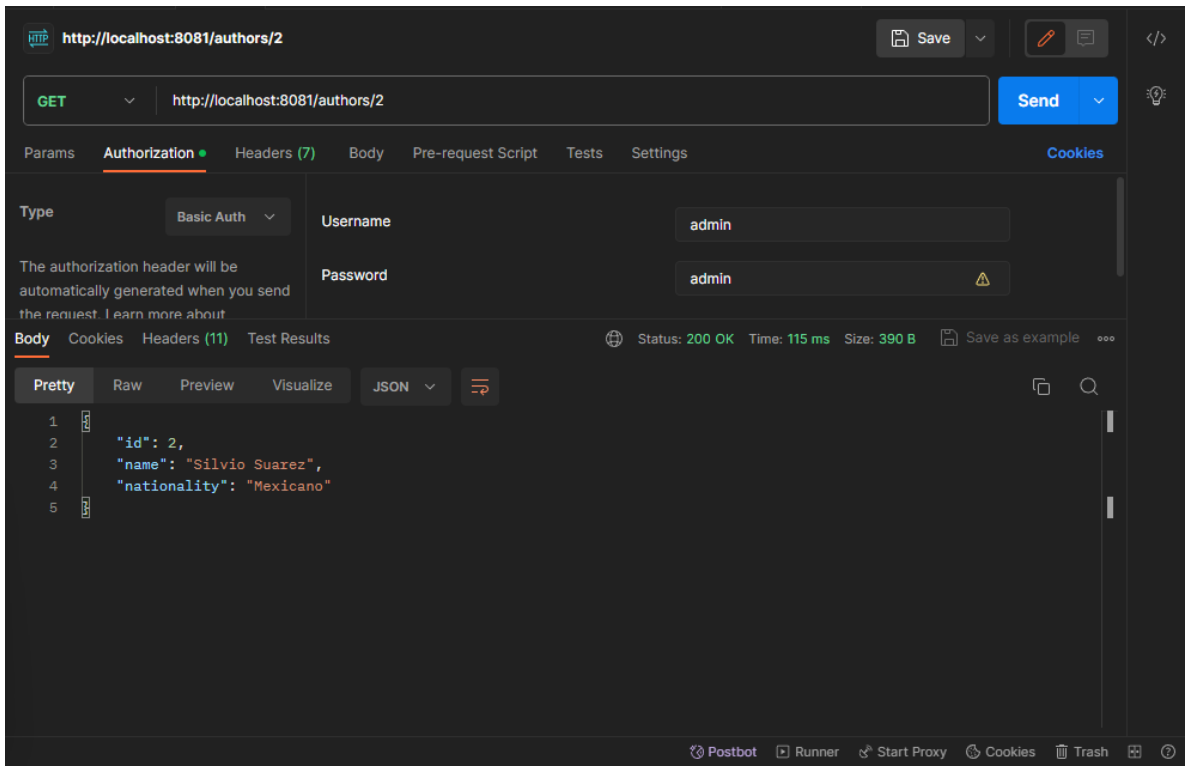
Resultado:



(3) GET /autores/{id}: Obtener detalles de un autor específico.

GET: http://localhost:8081/authors/2

Resultado:



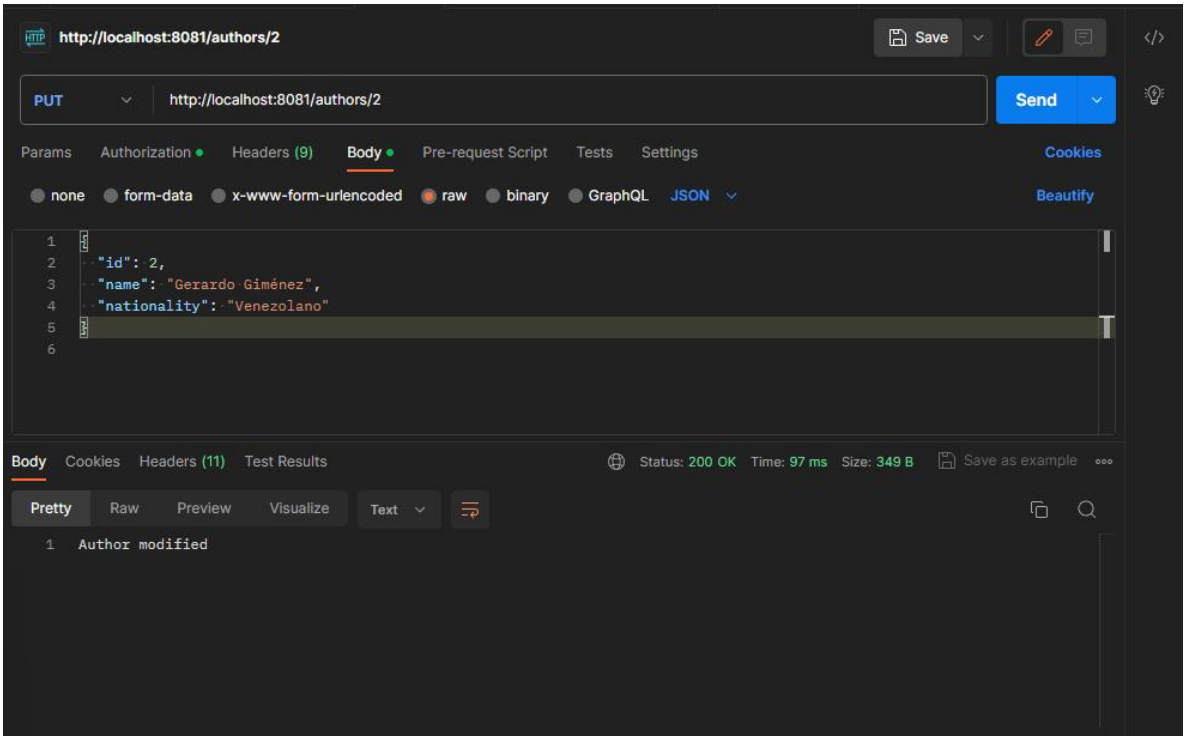
(4) PUT /autores/{id}: Actualizar un autor existente.

PUT: http://localhost:8081/authors/2

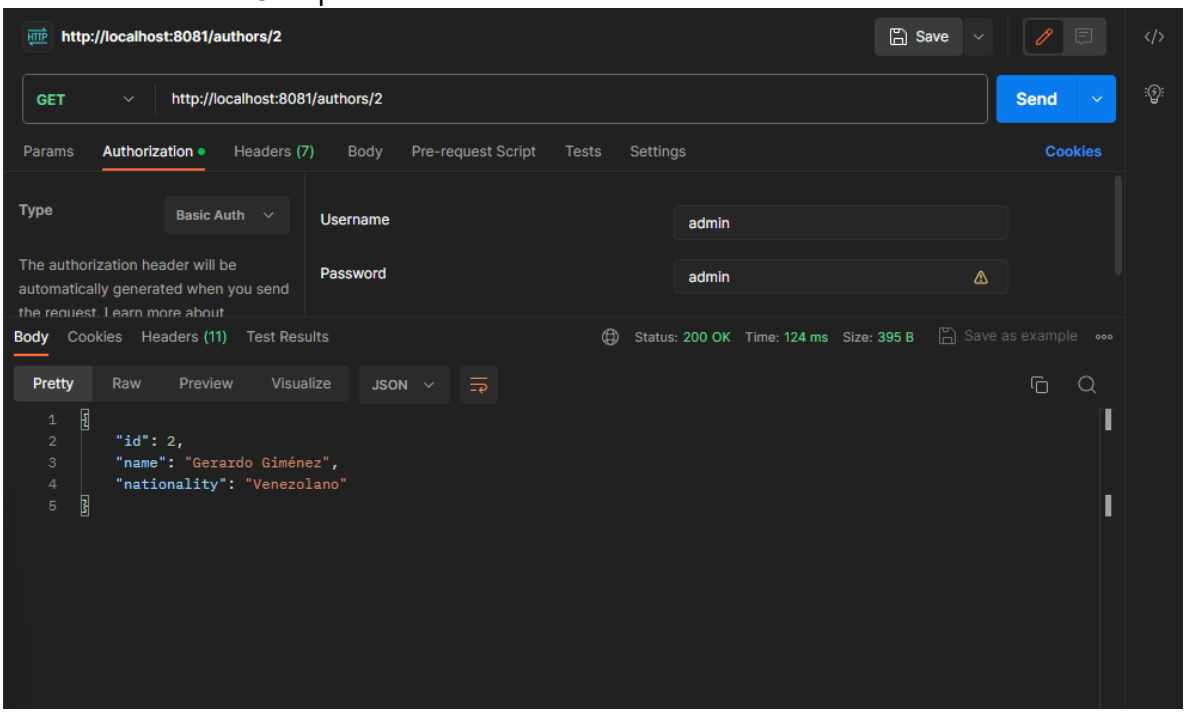
Body:

```
{
  "id": 2,
  "name": "Gerardo Giménez"
  "nationality": "Venezolano"
}
```

Resultado:



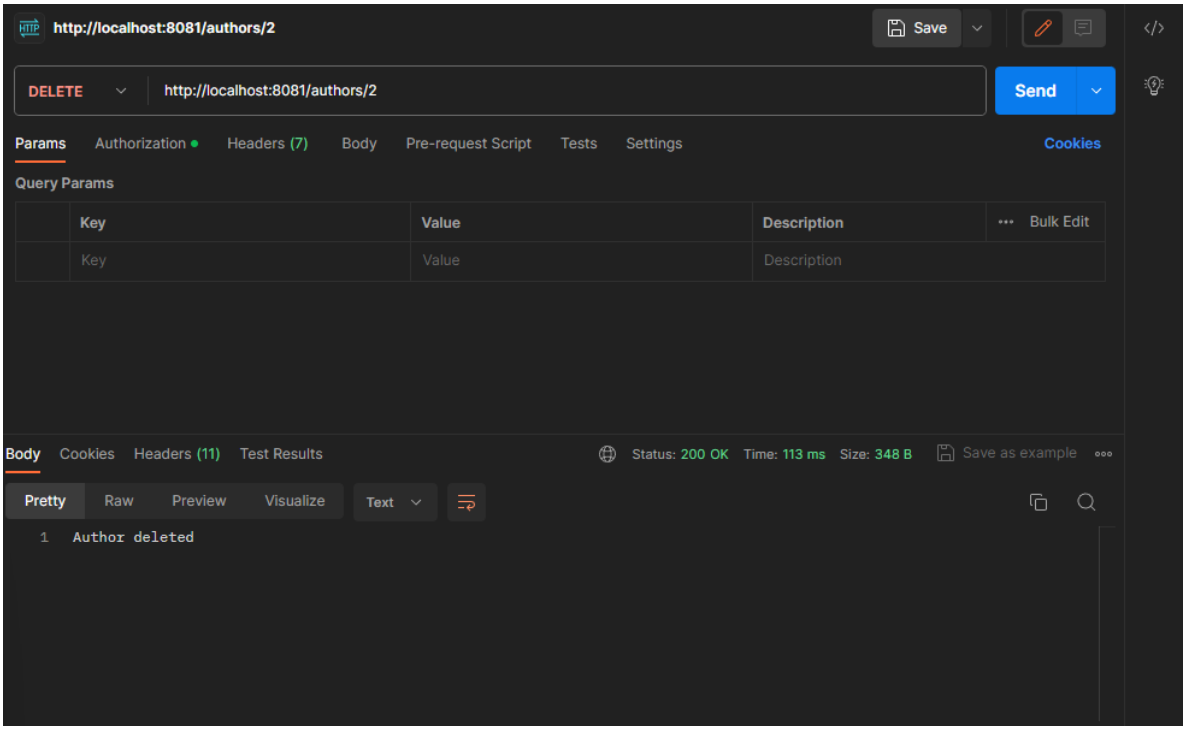
Verificamos con el GET por id:



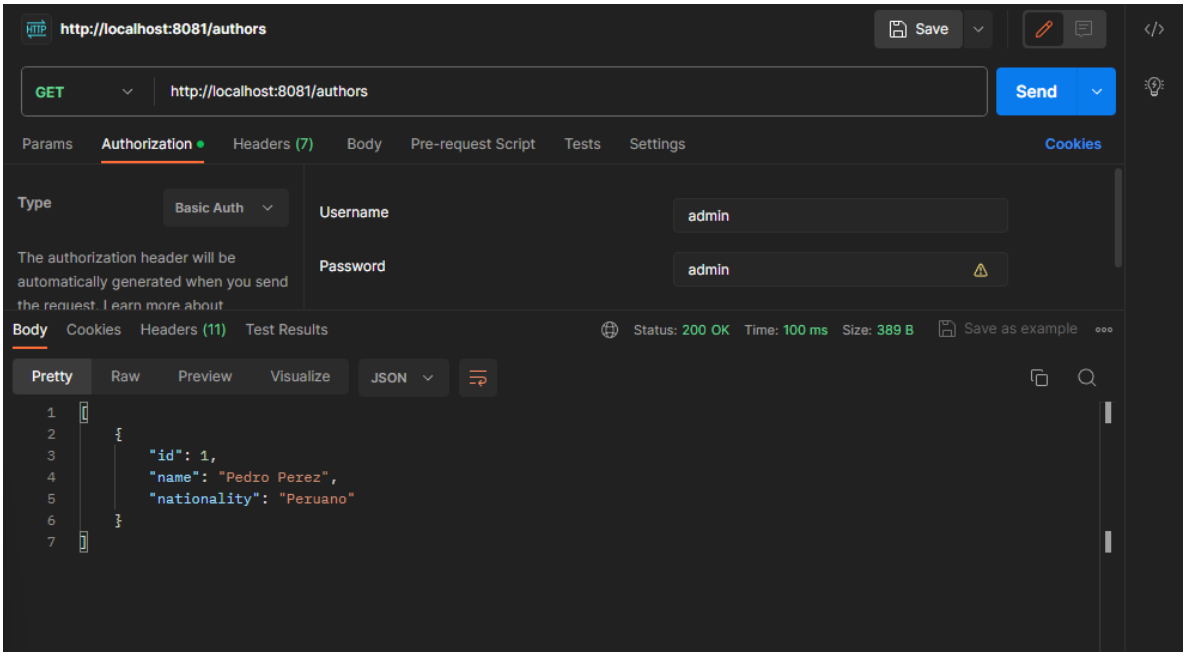
(5) DELETE /autores/{id}: Eliminar un autor.

DELETE: http://localhost:8081/authors/2

Resultado:



Verificamos con el GET:



NOTA: Dado que es necesario crear libros para mostrar los libros de un autor, “GET /autores/{id}/libros: Listar los libros de un autor específico.” será el último endpoint a comprobar en este informe/guía de uso.

Recordatorio: Es necesario usar la autenticación para las siguientes operaciones.

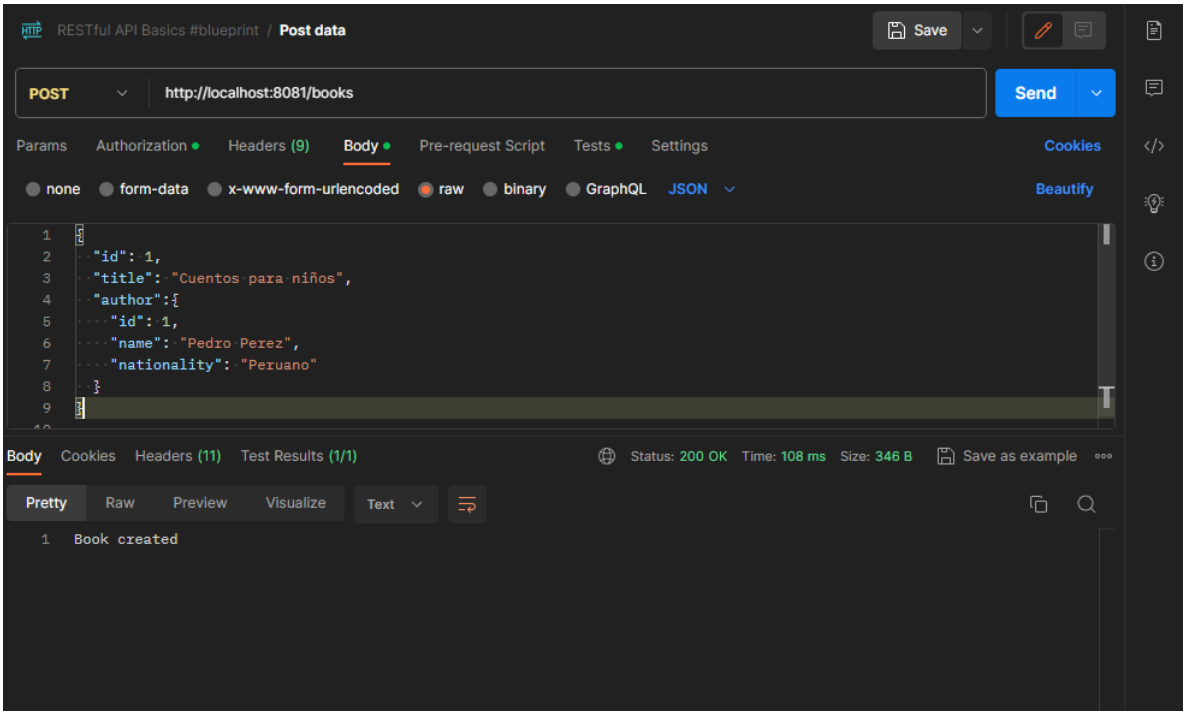
(6) POST /libros: Crear un nuevo libro.

POST: http://localhost:8081/books

Body:

```
{
  "id": 1,
  "title": "Cuentos para niños",
  "author":{
    "id": 1,
    "name": "Pedro Perez",
    "nationality": "Peruano"
  }
}
```

Resultado:

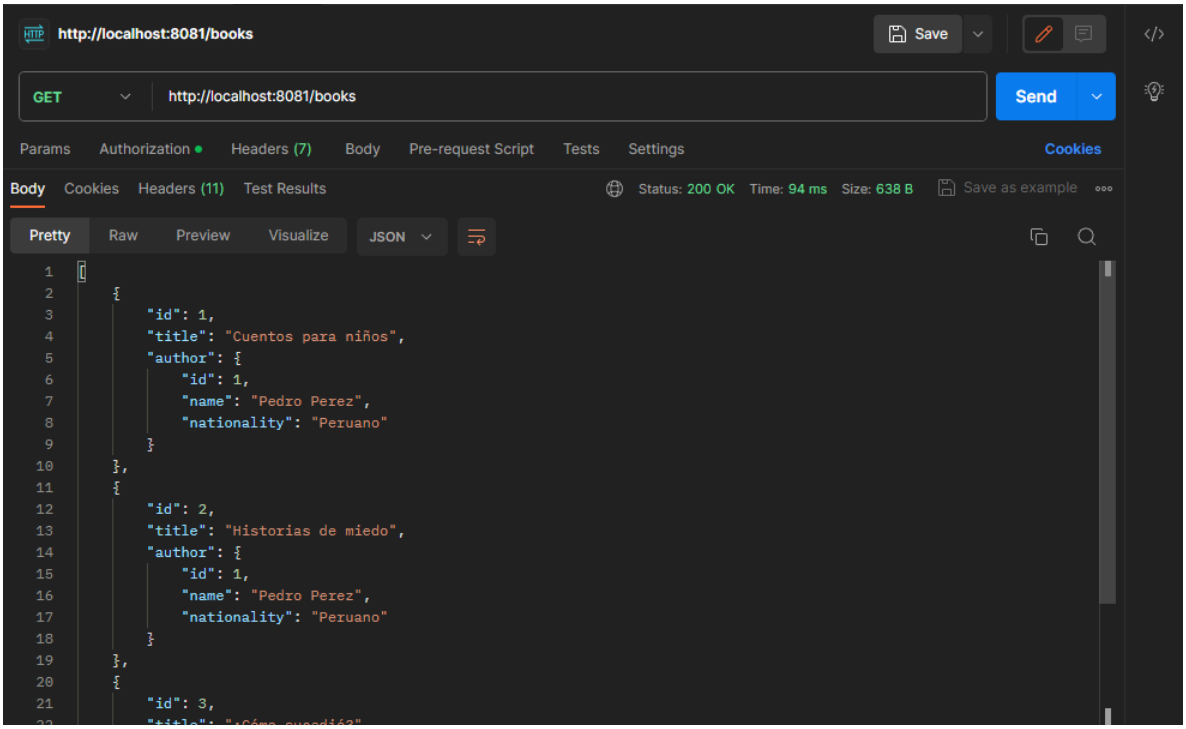


La inserción de este dato se verificará a continuación, haciendo uso del endpoint GET. Para mostrar que la aplicación retorna todos los libros, se crearán otros 2 libros con el autor que quedó después de hacer las operaciones del modelo Autor.

(7) GET /libros: Listar todos los libros.

GET: http://localhost:8081/books

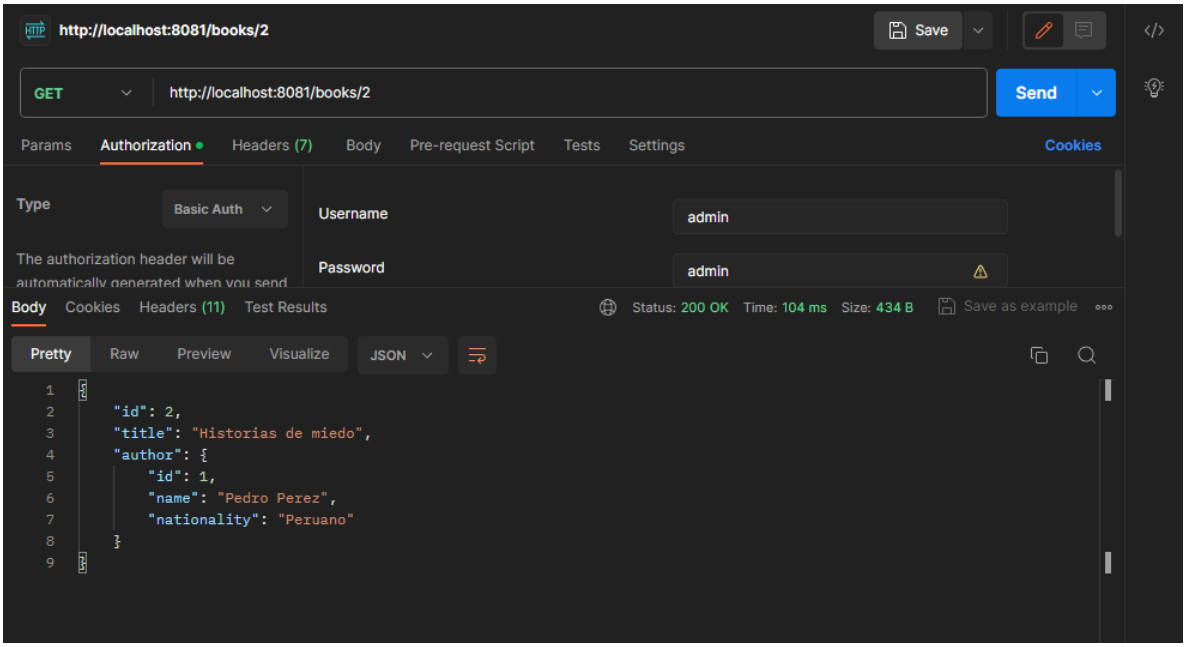
Resultado:



(8) GET /libros/{id}: Obtener detalles de un libro específico.

GET: http://localhost:8081/books/2

Resultado:



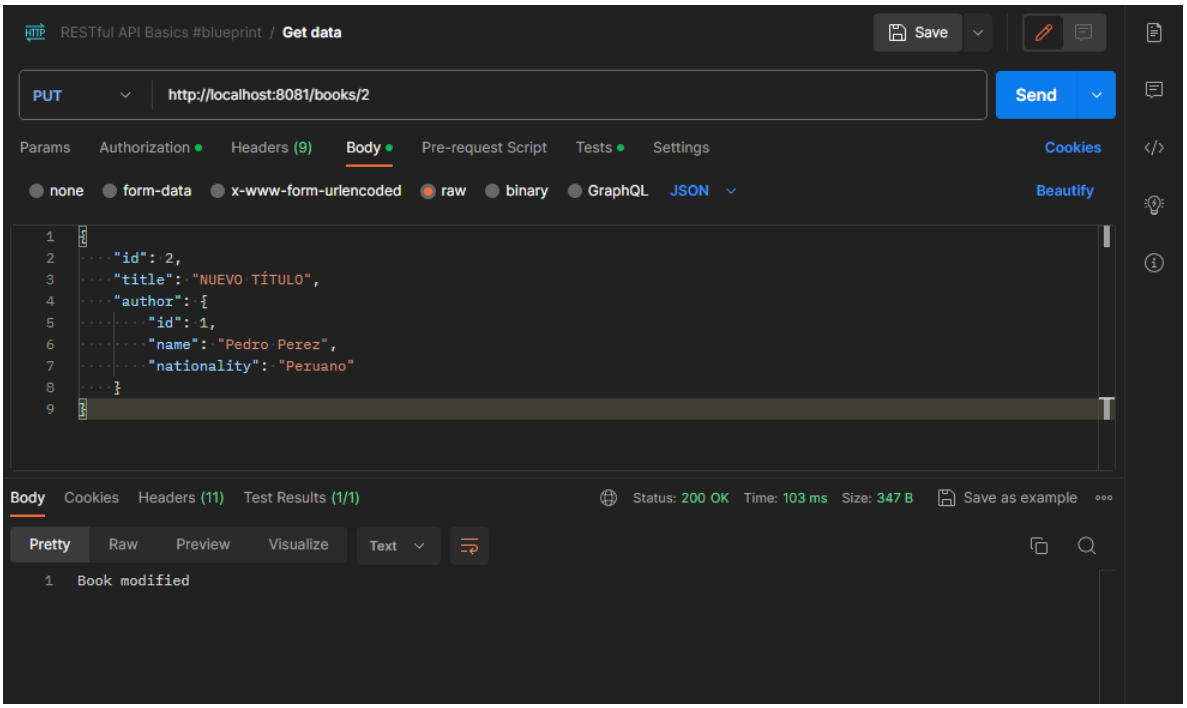
(9) PUT /libros/{id}: Actualizar un libro existente.

PUT: http://localhost:8081/books/2

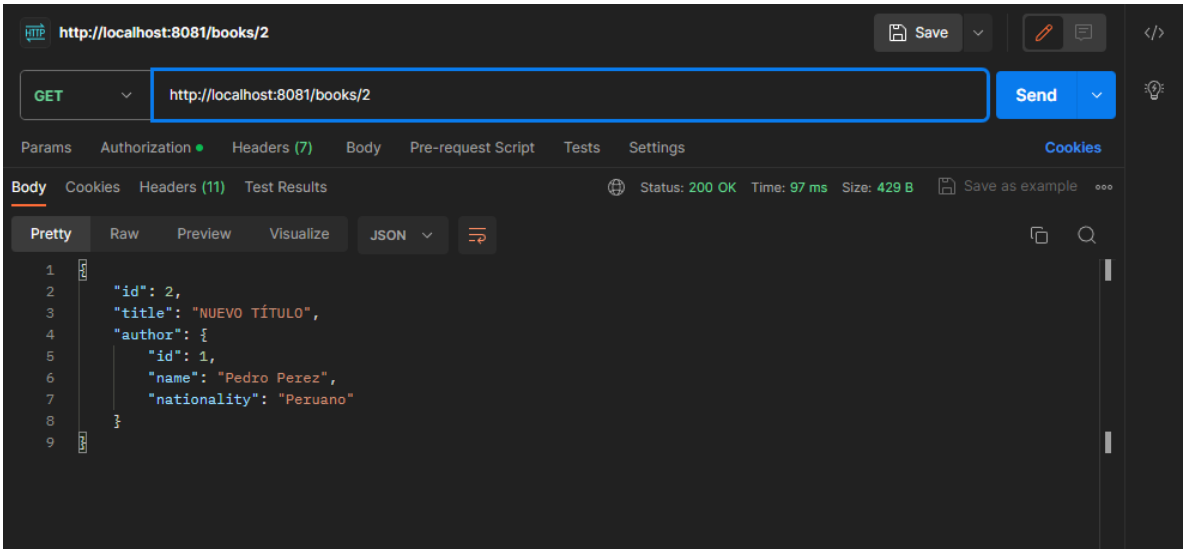
```
{
    "id": 2,
    "title": "NUEVO TITULO",
    "author": {
```

```
        "id": 1,
        "name": "Pedro Perez",
        "nationality": "Peruano"
    }
}
```

Resultado:



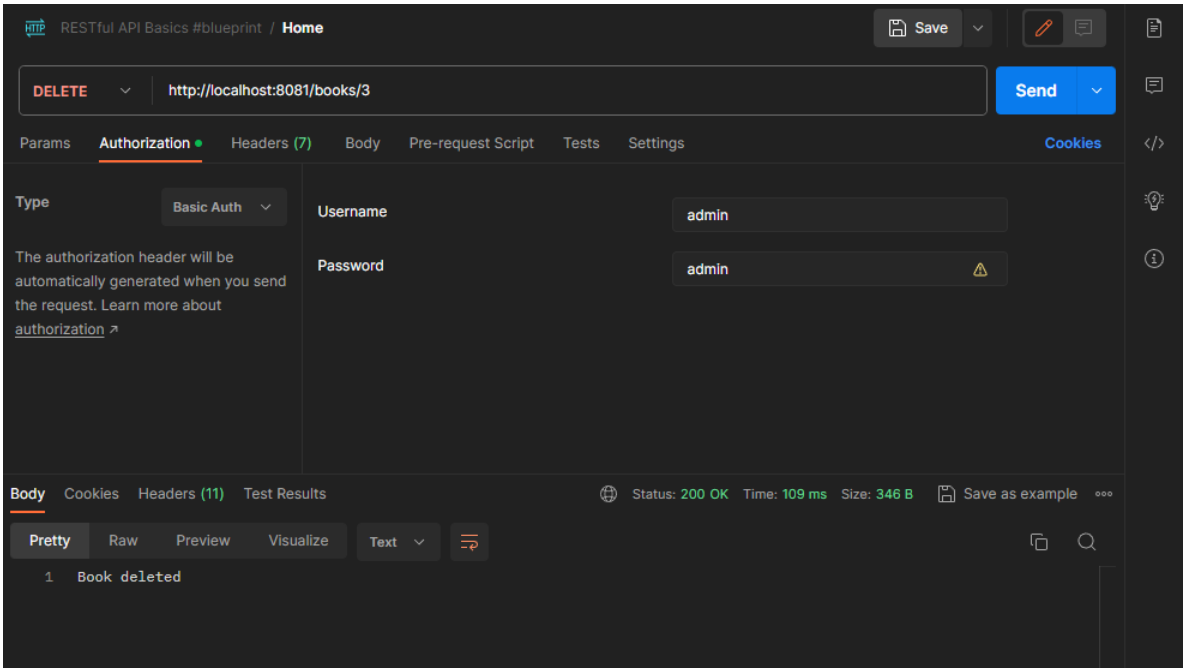
Para comprobarlo, hacemos la consulta mediante el GET por id:



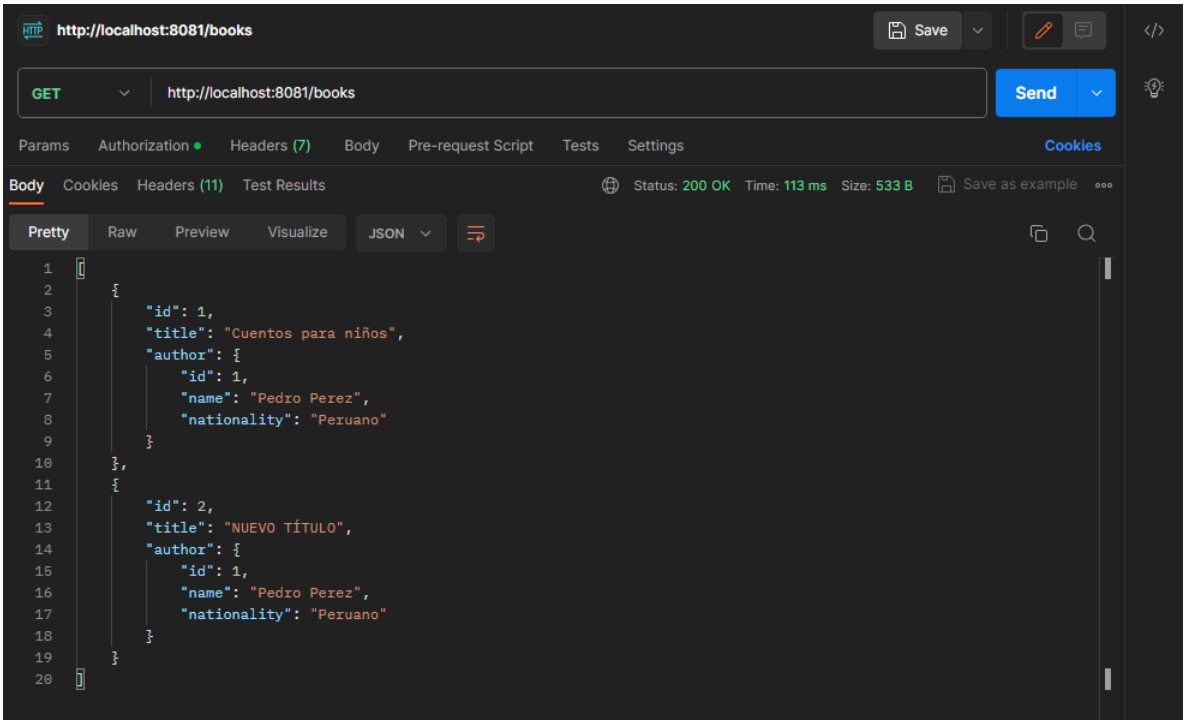
(10) DELETE /libros/{id}: Eliminar un libro.

DELETE: http://localhost:8081/books/3

Resultado:



Comprobamos el resultado mediante el GET:



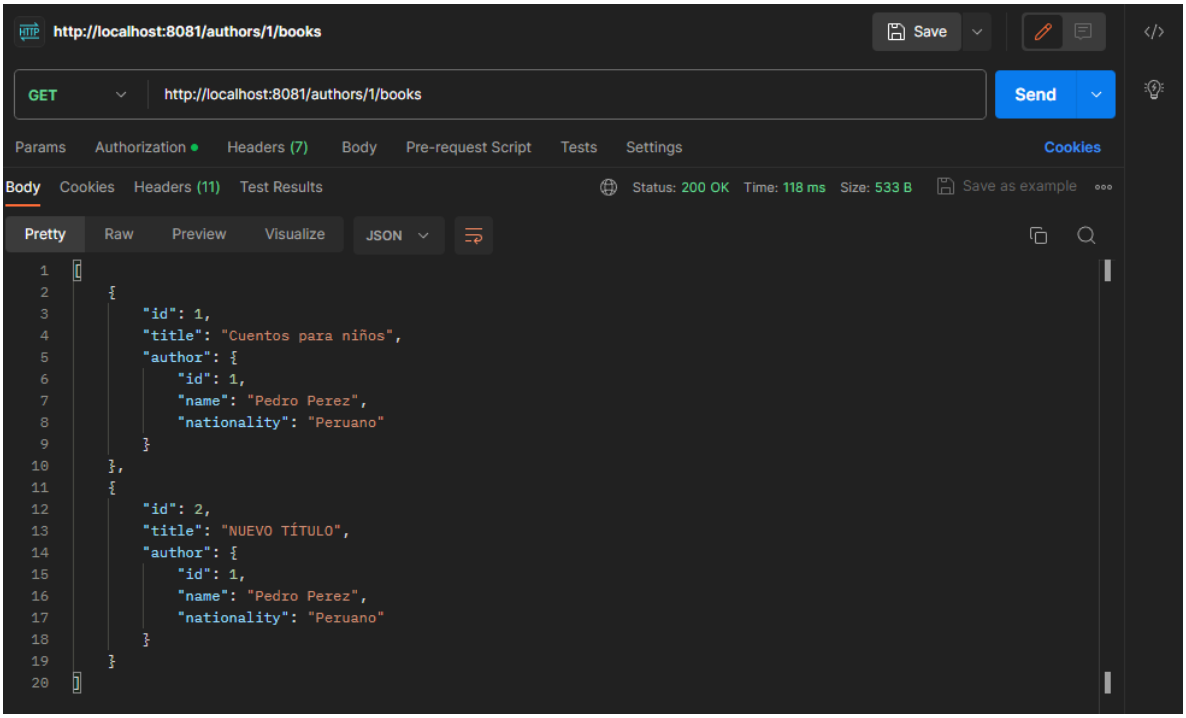
AUTOR

Como se estableció al final de las operaciones de autor, el último endpoint a comprobar será el siguiente:

(11) GET /autores/{id}/libros: Listar los libros de un autor específico.

GET: http://localhost:8081/authors/1/books

Resultado:


http://localhost:8081/authors/1/books
Save
GET
http://localhost:8081/authors/1/books
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Body
Cookies
Headers (11)
Test Results
Status: 200 OK
Time: 118 ms
Size: 533 B
Save as example
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"title": "Cuentos para niños",
"author": {
"id": 1,
"name": "Pedro Perez",
"nationality": "Peruano"
}
},
{
"id": 2,
"title": "NUEVO TÍTULO",
"author": {
"id": 1,
"name": "Pedro Perez",
"nationality": "Peruano"
}
}
]