



# Laboratorio 4a

## Explore REST APIs with API Simulator and Postman

Sergio Sebastian Pezo Jimenez - 20224087G

**Parte 1: Inicializamos la VM de DEVASC.**

**Parte 2: Nos conectamos a <http://library.demo.local/>**

Welcome to the School Library

Are you a developer? We have APIs! Click here for API docs.

Navigation

Our Books

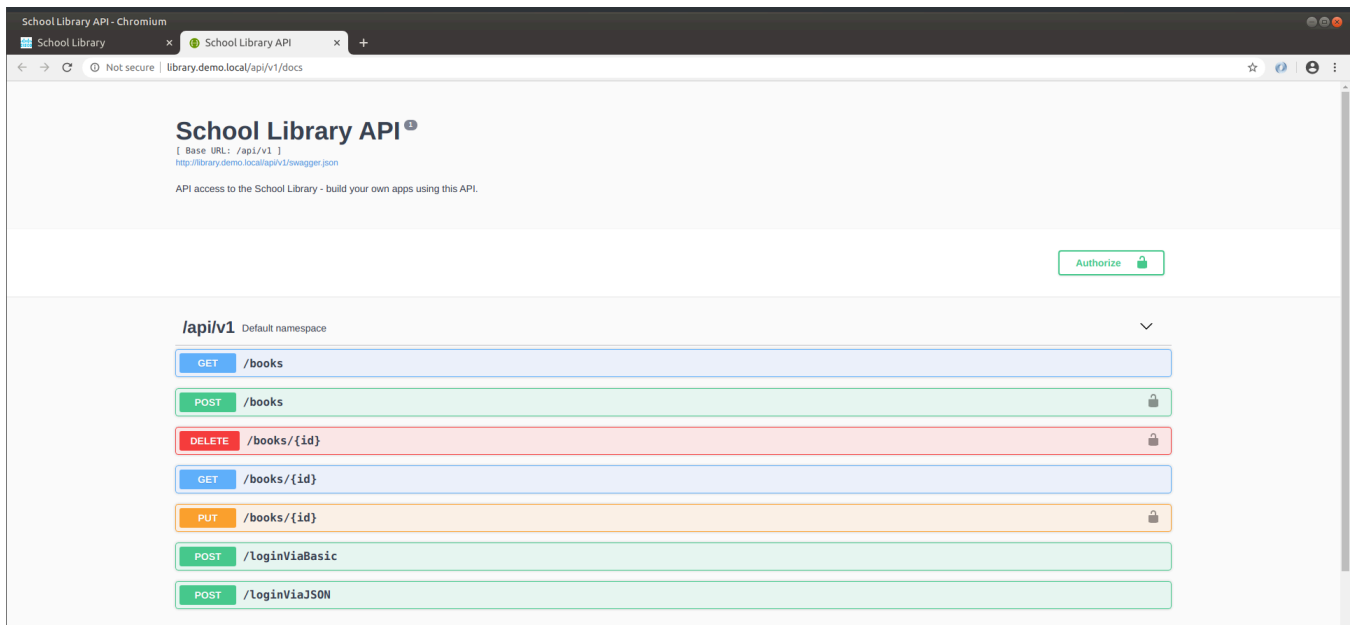
Open Hours

Contact

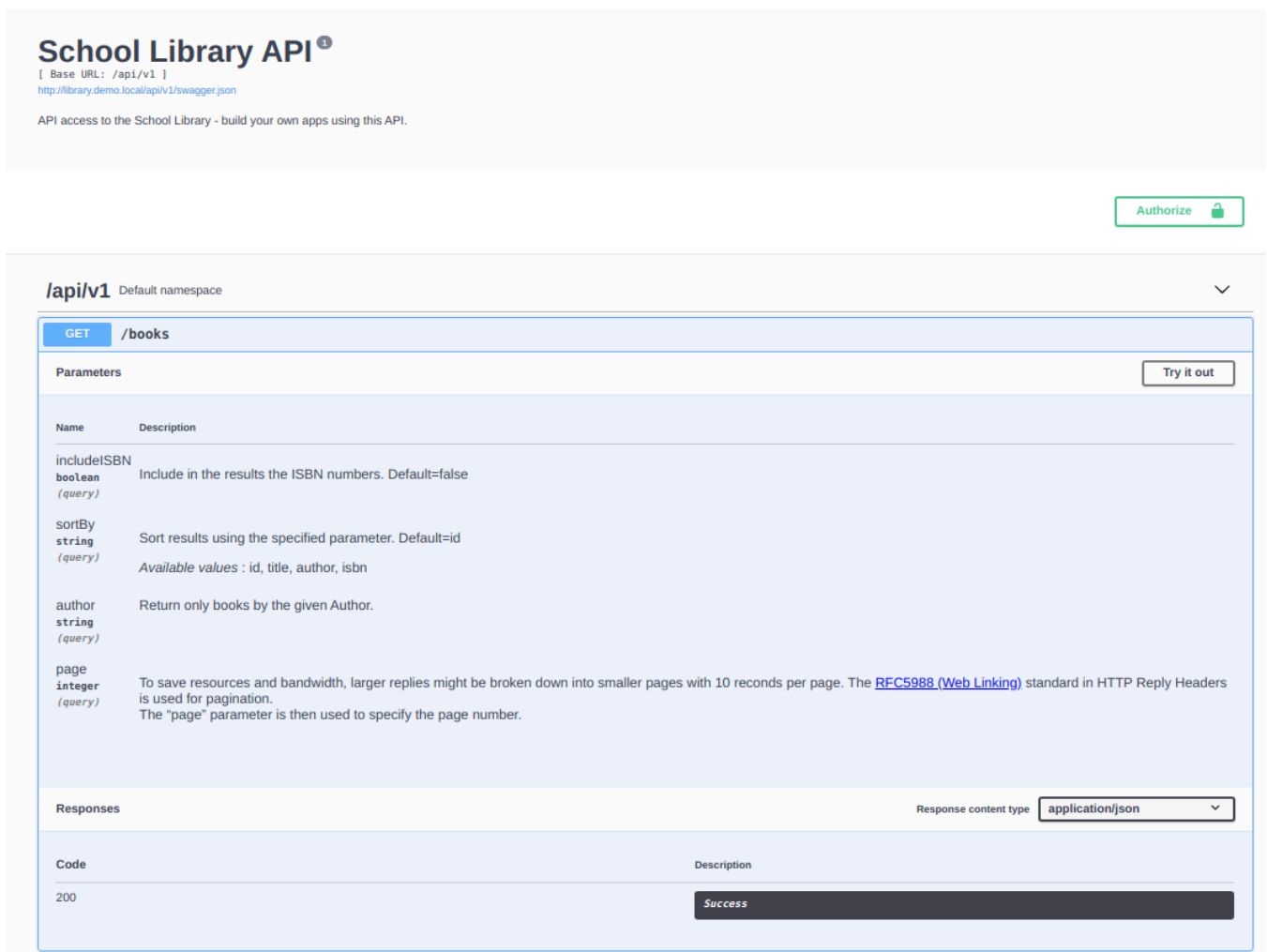
Here is a list of our books currently in the library.

Book ID	Title	Author
0	IP Routing Fundamentals	Mark A. Sportack
1	Python for Dummies	Stef Maruch Aahz Maruch
2	Linux for Networkers	Cisco Systems Inc.
3	NetAcad: 20 Years Of Online-Learning	Cisco Systems Inc.

Vamos al swagger de la API



Clickamos en GET /books API



Damos click en Try it out, para probar el endpoint previamente mencionado, y notamos qu el nos devuelve una lista de libros.

The screenshot displays a REST client interface with the following sections:

- Execute**: A blue button to execute the request, highlighted with a red box.
- Clear**: A button to clear the request.
- Responses**: A section showing the response content type as `application/json`.
- Curl**: A text area containing the command: `curl -X GET "http://library.demo.local/api/v1/books" -H "accept: application/json"`.
- Request URL**: A text area containing the URL: `http://library.demo.local/api/v1/books`.
- Server response**: A section showing the response code `200`.
- Details**: A section showing the response body, which is a JSON array of book objects, highlighted with a red box. The JSON data is as follows:

```
{
  "id": 0,
  "title": "IP Routing Fundamentals",
  "author": "Mark A. Sportack"
},
{
  "id": 1,
  "title": "Python for Dummies",
  "author": "Stef Maruch Aahz Maruch"
},
{
  "id": 2,
  "title": "Linux for Networkers",
  "author": "Cisco Systems Inc."
},
{
  "id": 3,
  "title": "NetAcad: 20 Years Of Online-Learning",
  "author": "Cisco Systems Inc."
}
}
```
- Response headers**: A section showing the response headers:

```
access-control-allow-origin: *
content-length: 463
content-type: application/json
date: Sat, 14 Sep 2024 17:12:57 GMT
server: Werkzeug/0.14.1 Python/3.8.2
```
- Responses**: A section showing the response code `200` and a **Success** status.

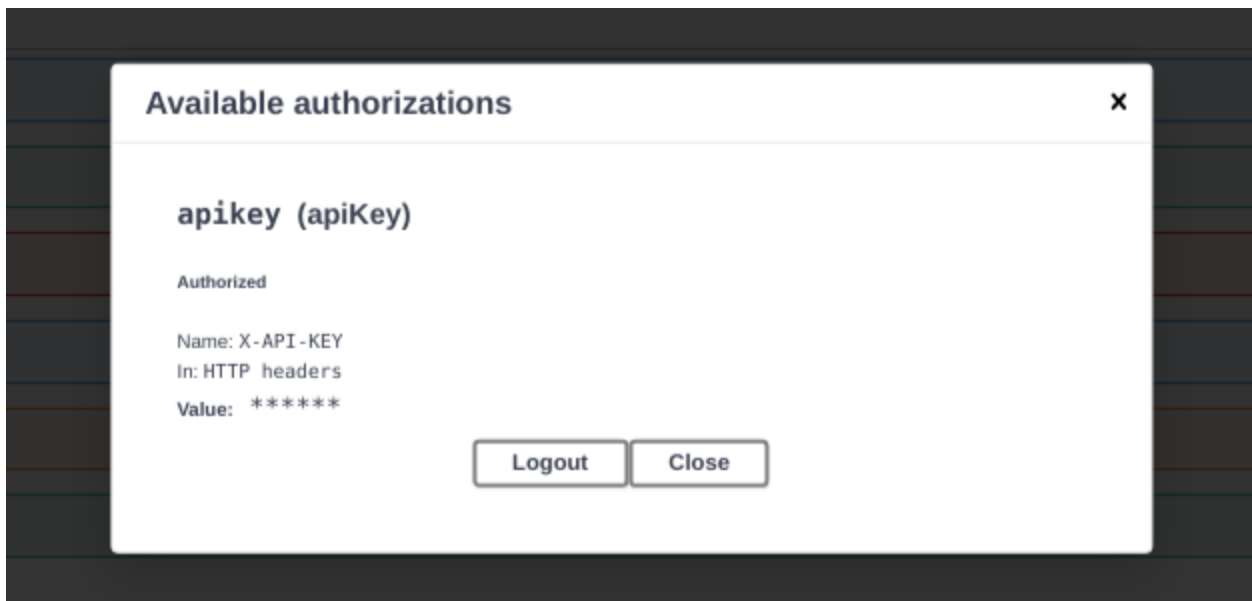
Ahora probamos el endpoint con el comando curl en la terminal.

```
devasc@labvm: ~  
File Edit View Search Terminal Help  
devasc@labvm:~$ curl -X GET "http://library.demo.local/api/v1/books" -H "accept: application/json"  
[{"id": 0,  
  "title": "IP Routing Fundamentals",  
  "author": "Mark A. Sportack"},  
{"id": 1,  
  "title": "Python for Dummies",  
  "author": "Stef Maruch Aahz Maruch"},  
{"id": 2,  
  "title": "Linux for Networkers",  
  "author": "Cisco Systems Inc."},  
{"id": 3,  
  "title": "NetAcad: 20 Years Of Online-Learning",  
  "author": "Cisco Systems Inc."}]
```

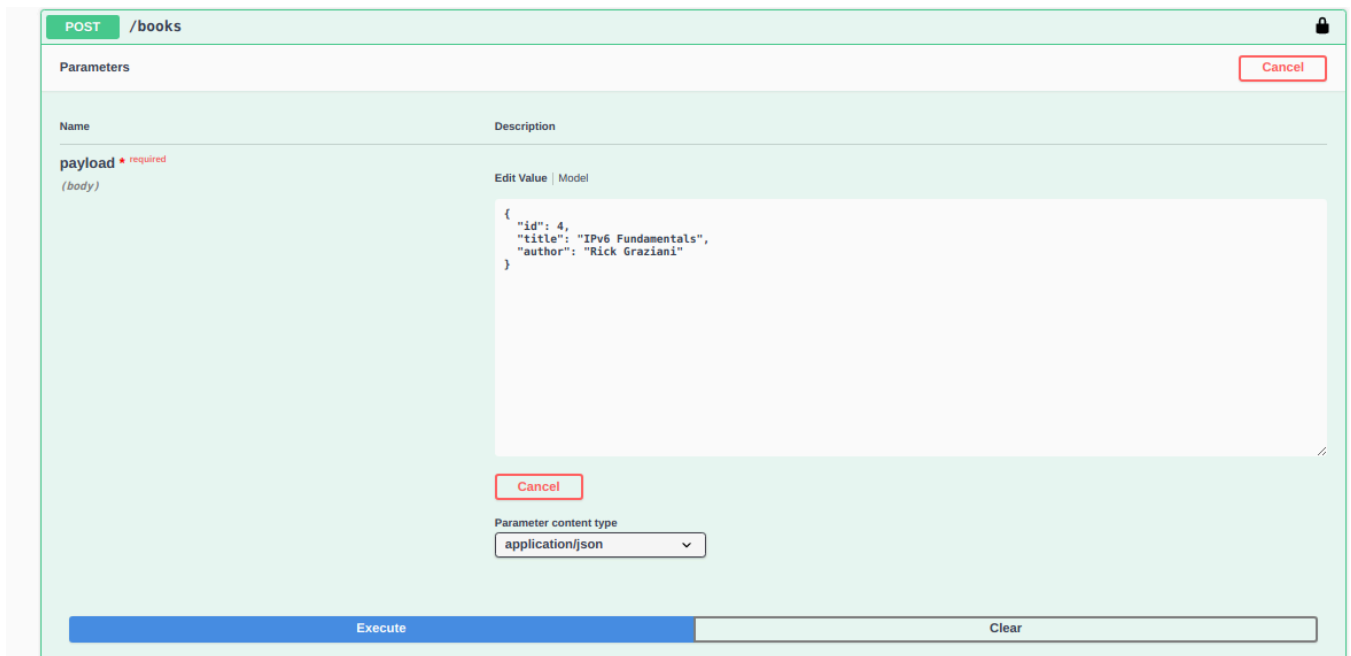
Incluimos el parámetro includeISBN a la petición

```
devasc@labvm:~$ curl -X GET "http://library.demo.local/api/v1/book?includeISBN=true" -H "accept: application/json"  
[{"id": 0,  
  "title": "IP Routing Fundamentals",  
  "author": "Mark A. Sportack",  
  "isbn": "978-1578700714"},  
{"id": 1,  
  "title": "Python for Dummies",  
  "author": "Stef Maruch Aahz Maruch",  
  "isbn": "978-0471778646"},  
{"id": 2,  
  "title": "Linux for Networkers",  
  "author": "Cisco Systems Inc.",  
  "isbn": "000-0000000123"},  
{"id": 3,  
  "title": "NetAcad: 20 Years Of Online-Learning",  
  "author": "Cisco Systems Inc.",  
  "isbn": "000-0000001123"}]
```

Ahora estamos en el endpoint POST /loginViaBasis, el cual no devolvió un APIkey de autorización



Ahora procederemos a añadir un libro con el endpoint POST /books,



EN caso coloquemos correctamente el token previamente mencionado, obtendremos una respuesta 200.

Curl

```
curl -X POST "http://library.demo.local/api/v1/books" -H "accept: application/json" -H "X-API-KEY: cisco|Ms4dBNZEYKGS6JUuLTXXfpWIINDoTDZiZAqUFS3Ec" -H "Content-Type: application/json" -d '{"id": 4, "title": "IPv6 Fundamentals", "author": "Rick Graziani"}'
```

Request URL

http://library.demo.local/api/v1/books

Server response

Code	Details
200	<p>Response body</p> <pre>{   "id": 4,   "title": "IPv6 Fundamentals",   "author": "Rick Graziani" }</pre> <p>Response headers</p> <pre>access-control-allow-origin: http://library.demo.local content-length: 81 content-type: application/json date: Sat, 14 Sep 2024 17:33:06 GMT server: Werkzeug/0.14.1 Python/3.8.2 vary: Origin</pre>

Ahora al hacer un GET de los libros lograremos observar nuestros libros previamente aregados.

Curl

```
curl -X GET "http://library.demo.local/api/v1/books" -H "accept: application/json"
```

Request URL

http://library.demo.local/api/v1/books

Server response

Code	Details
200	<p>Response body</p> <pre>{   "author": "Mark A. Sportack",   "id": 1,   "title": "Python for Dummies",   "author": "Stef Maruch Aahz Maruch",   "id": 2,   "title": "Linux for Networkers",   "author": "Cisco Systems Inc.",   "id": 3,   "title": "NetAcad: 20 Years Of Online-Learning",   "author": "Cisco Systems Inc.",   "id": 4,   "title": "IPv6 Fundamentals",   "author": "Rick Graziani",   "id": 5,   "title": "31 Days Before Your CCNA Exam",   "author": "Allan Johnson" }</pre> <p>Response headers</p> <pre>access-control-allow-origin: * content-length: 679 content-type: application/json date: Sat, 14 Sep 2024 17:35:19 GMT server: Werkzeug/0.14.1 Python/3.8.2</pre>

Buscamos un libro por su id, con GET /books/{id}

GET /books/{id}

Parameters Cancel

Name	Description
id <span>required</span> integer (path)	<input type="text" value="4"/>

Execute Clear

Responses Response content type: application/json

Curl

```
curl -X GET "http://library.demo.local/api/v1/books/4" -H "accept: application/json"
```

Request URL

```
http://library.demo.local/api/v1/books/4
```

Server response

Code	Details
200	<div><div>Response body</div><pre>{   "id": 4,   "title": "IPv6 Fundamentals",   "author": "Rick Graziani" }</pre><span>Download</span></div>

Eliminamos un libro con el endpoint DELETE /books/{id}

DELETE /books/{id} 🔒

Parameters Cancel

Name	Description
id <span>required</span> integer (path)	<input type="text" value="4"/>

Execute Clear

Responses Response content type: application/json

Curl

```
curl -X DELETE "http://library.demo.local/api/v1/books/4" -H "accept: application/json" -H "X-API-KEY: cisco|Ms4dBNZEYKGNs6JUuLTXXfpwIINDoTDZiZAqUFS3Ec"
```

Request URL

```
http://library.demo.local/api/v1/books/4
```

Server response

Code	Details
200	<div><div>Response body</div><pre>{   "id": 4,   "title": "IPv6 Fundamentals",   "author": "Rick Graziani" }</pre><span>Download</span></div> <div><div>Response headers</div><pre>access-control-allow-origin: http://library.demo.local content-length: 81 content-type: application/json date: Sat, 14 Sep 2024 17:40:00 GMT server: Werkzeug/0.14.1 Python/3.8.2 vary: Origin</pre></div>

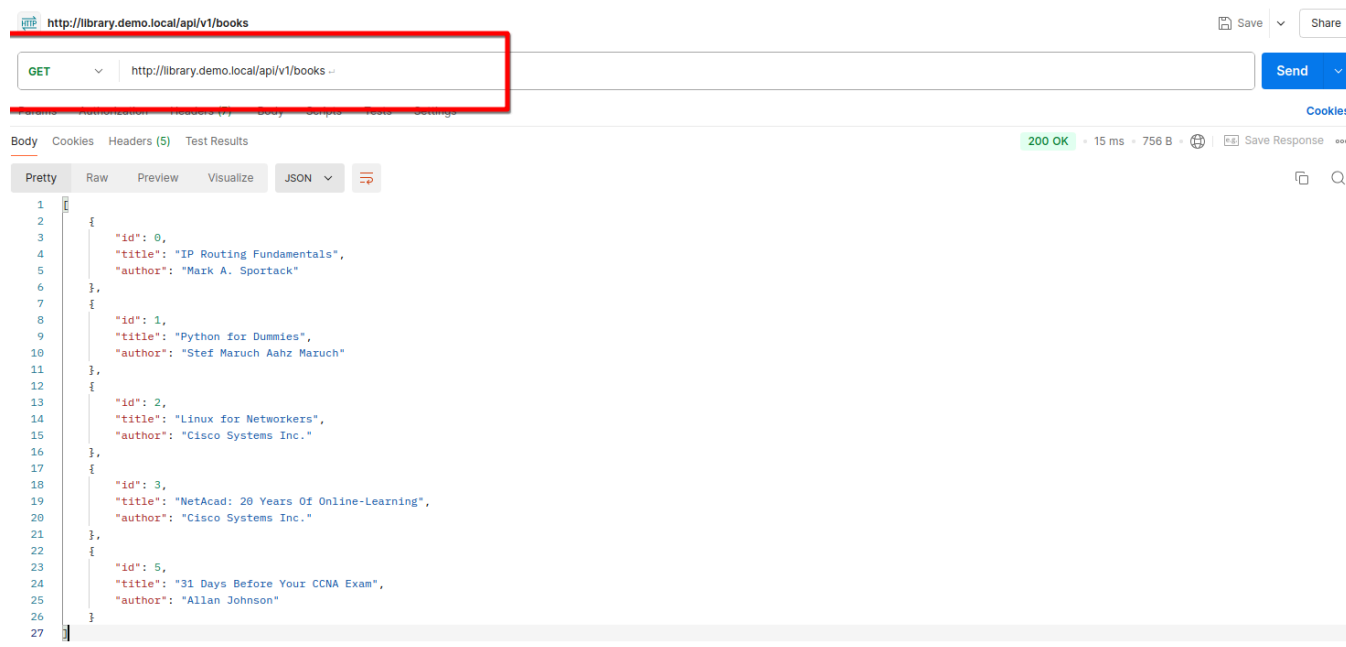
Veamos si realmente se eliminó. En efecto, al hacer nuevamente un GET, ya no

está el libro 4.



## Parte 3: Ahora usaremos POSTMAN

Empezamos con GET



Obtenemos otro token, pero ahora con POSTMAN:



http://library.demo.local/api/v1/loginViaBasic

POST http://library.demo.local/api/v1/loginViaBasic

Auth Type: Basic Auth

Username: cisco

Password: Cisco123!

Body: {"token": "cisco|mgpxiMGqvT4x28SqQYj3hy9zNjwzS250eCyBq5KmFhQ"}

Ahora, jugamos con el POST para agregar un libro

http://library.demo.local/api/v1/books

POST http://library.demo.local/api/v1/books

Auth Type: API Key

Key: X-API-KEY

Value: cisco|mgpxiMGqvT4x28SqQYj3hy9zNjwzS250eCyBq5KmFhQ

Body: {  
 "id": 4,  
 "title": "IPv6 Fundamentals",  
 "author": "Rick Graziani"  
}

Ahora verificaremos si realmente se agregó, además, añadimos un par de queries a la petición.

http://library.demo.local/api/v1/books?includeISBN=true&sortBy=author

GET http://library.demo.local/api/v1/books?includeISBN=true&sortBy=author

Auth Type: No Auth

Body: [  
 {  
 "id": 5,  
 "title": "31 Days Before Your CCNA Exam",  
 "author": "Allan Johnson"  
 },  
 {  
 "id": 2,  
 "title": "Linux for Networkers",  
 "author": "Cisco Systems Inc.",  
 "isbn": "000-0000000123"  
 },  
 {  
 "id": 3,  
 "title": "MetAcad: 20 Years Of Online-Learning",  
 "author": "Cisco Systems Inc.",  
 "isbn": "000-00000001123"  
 },  
 {  
 "id": 0,  
 "title": "IP Routing Fundamentals",  
 "author": "Mark A. Sportack",  
 "isbn": "978-1578700714"  
 },  
 {  
 "id": 4,  
 "title": "IPv6 Fundamentals",  
 "author": "Rick Graziani"  
 }  
]

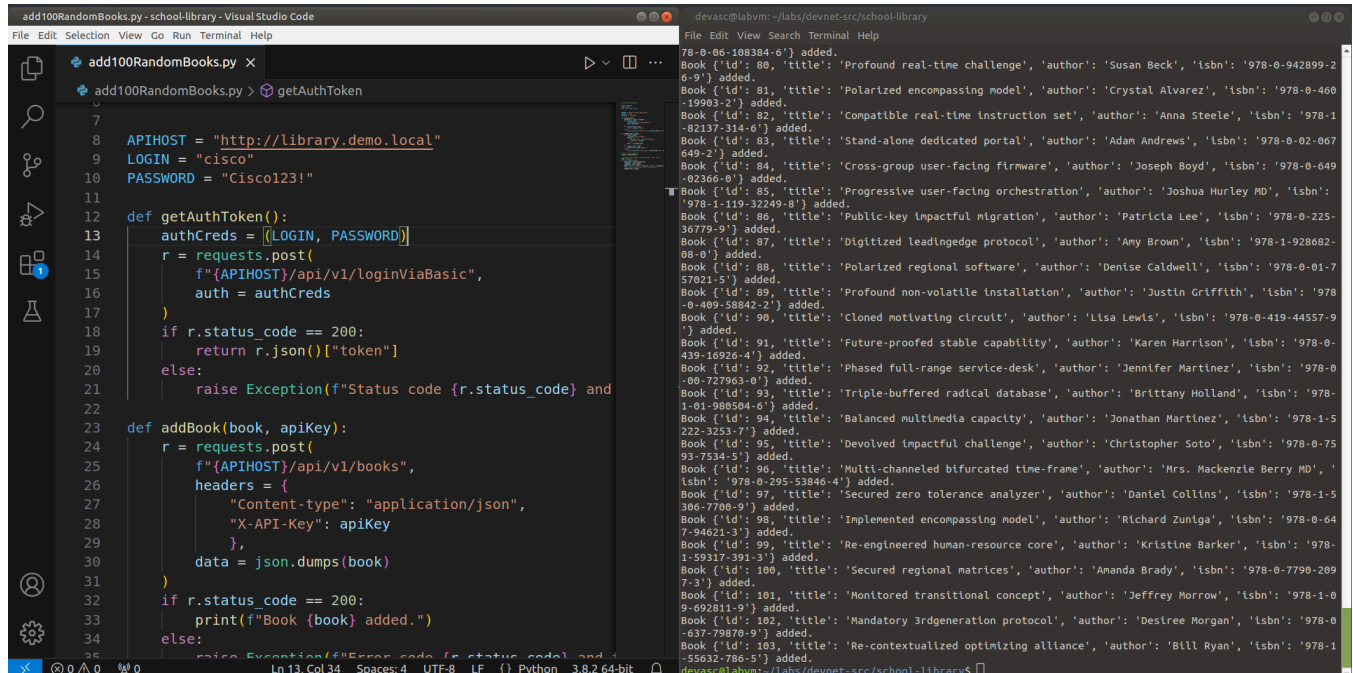
## Parte 4: Usando Python para agregar 100 libros

Primero jugamos con el módulo faker.

```
devasc@labvm:~/labs/devnet-src/school-library$ python3
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from faker import Faker
>>> fake = Faker()
>>> print('My name is {}'.format(fake.name()))
My name is Charles Lewis.
>>> print('My name is {} and I wrote "{}" (ISBN {}).'.format(fake.name(),fake.catch_phrase(),fake.isbn13()))
My name is Cynthia Rodriguez and I wrote "Object-based 24hour throughput" (ISBN 978-1-965763-98-8).
```

```
>>> for i in range(10):
...     print(fake.name())
...
Samuel Galvan
Joseph Barnes
Christopher Reed
Richard Lopez
Amy Richardson
Kathleen Villa
Angela Weaver
David Ortiz
Shaun Bishop
Christopher Johnson
>>>
```

Analizamos el código de Python y agregamos los 100 libros:



```
add100RandomBooks.py - school-library - Visual Studio Code
File Edit Selection View Go Run Terminal Help

add100RandomBooks.py X
8 APIHOST = "http://library.demo.local"
9 LOGIN = "cisco"
10 PASSWORD = "Cisco123!"
11
12 def getAuthToken():
13     authCreds = [(LOGIN, PASSWORD)]
14     r = requests.post(
15         f"(APIHOST)/api/v1/loginViaBasic",
16         auth = authCreds
17     )
18     if r.status_code == 200:
19         return r.json()["token"]
20     else:
21         raise Exception(f"Status code {r.status_code} and
22
23 def addBook(book, apiKey):
24     r = requests.post(
25         f"(APIHOST)/api/v1/books",
26         headers = {
27             "Content-type": "application/json",
28             "X-API-Key": apiKey
29         },
30         data = json.dumps(book)
31     )
32     if r.status_code == 200:
33         print(f"Book {book} added.")
34     else:
35         raise Exception(f"Error code {r.status_code} and
36
devasc@labvm:~/labs/devnet-src/school-library$
78-0-06-108384-6') added.
Book {'id': 80, 'title': 'Profound real-time challenge', 'author': 'Susan Beck', 'isbn': '978-0-942899-2
6-9') added.
Book {'id': 81, 'title': 'Polarized encompassing model', 'author': 'Crystal Alvarez', 'isbn': '978-0-460
-19003-2') added.
Book {'id': 82, 'title': 'Compatible real-time instruction set', 'author': 'Anna Steele', 'isbn': '978-1
-82137-314-6') added.
Book {'id': 83, 'title': 'Stand-alone dedicated portal', 'author': 'Adam Andrews', 'isbn': '978-0-02-067
649-2') added.
Book {'id': 84, 'title': 'Cross-group user-facing firmware', 'author': 'Joseph Boyd', 'isbn': '978-0-649
-02360-0') added.
Book {'id': 85, 'title': 'Progressive user-facing orchestration', 'author': 'Joshua Hurley MD', 'isbn':
'978-1-119-32249-8') added.
Book {'id': 86, 'title': 'Public-key impactful migration', 'author': 'Patricia Lee', 'isbn': '978-0-225-
30779-9') added.
Book {'id': 87, 'title': 'Digitized leadingedge protocol', 'author': 'Amy Brown', 'isbn': '978-1-928682-
08-0') added.
Book {'id': 88, 'title': 'Polarized regional software', 'author': 'Denise Caldwell', 'isbn': '978-0-01-7
57021-5') added.
Book {'id': 89, 'title': 'Profound non-volatile installation', 'author': 'Justin Griffith', 'isbn': '978
-0-409-50842-2') added.
Book {'id': 90, 'title': 'Cloned motivating circuit', 'author': 'Lisa Lewis', 'isbn': '978-0-419-44557-9
') added.
Book {'id': 91, 'title': 'Future-proofed stable capability', 'author': 'Karen Harrison', 'isbn': '978-0-
439-16926-4') added.
Book {'id': 92, 'title': 'Phased full-range service-desk', 'author': 'Jennifer Martinez', 'isbn': '978-0
-00-727963-0') added.
Book {'id': 93, 'title': 'Triple-buffered radical database', 'author': 'Brittany Holland', 'isbn': '978-
1-01-980504-6') added.
Book {'id': 94, 'title': 'Balanced multimedia capacity', 'author': 'Jonathan Martinez', 'isbn': '978-1-5
222-3253-7') added.
Book {'id': 95, 'title': 'Devolved impactful challenge', 'author': 'Christopher Soto', 'isbn': '978-0-75
93-7534-5') added.
Book {'id': 96, 'title': 'Multi-channelled bifurcated time-frame', 'author': 'Mrs. Mackenzie Berry MD', '
isbn': '978-0-295-53846-4') added.
Book {'id': 97, 'title': 'Secured zero tolerance analyzer', 'author': 'Daniel Collins', 'isbn': '978-1-5
306-7700-9') added.
Book {'id': 98, 'title': 'Implemented encompassing model', 'author': 'Richard Zuniga', 'isbn': '978-0-64
7-94021-3') added.
Book {'id': 99, 'title': 'Re-engineered human-resource core', 'author': 'Kristine Barker', 'isbn': '978-
1-59317-391-3') added.
Book {'id': 100, 'title': 'Secured regional matrices', 'author': 'Amanda Brady', 'isbn': '978-0-7790-209
7-3') added.
Book {'id': 101, 'title': 'Monitored transitional concept', 'author': 'Jeffrey Morrow', 'isbn': '978-1-0
9-692811-9') added.
Book {'id': 102, 'title': 'Mandatory 3rdgeneration protocol', 'author': 'Destree Morgan', 'isbn': '978-0
-637-79870-9') added.
Book {'id': 103, 'title': 'Re-contextualized optimizing alliance', 'author': 'Bill Ryan', 'isbn': '978-1
-55632-780-5') added.
```

Ahora volvemos al swagger y verificamos, añadimos como query page=8, pues

por temas de cuidar el ancho de banda, se divide por grupos de 10, probemos con 8

The screenshot shows a REST client interface with the following sections:

- Request URL:** `http://library.demo.local/api/v1/books?page=8`
- Server response:** `200`
- Code:** A red box highlights the JSON response body, which is a list of 10 book objects (IDs 80-89).

```
{
  "id": 80,
  "title": "Profound real-time challenge",
  "author": "Susan Beck"
},
{
  "id": 81,
  "title": "Polarized encompassing model",
  "author": "Crystal Alvarez"
},
{
  "id": 82,
  "title": "Compatible real-time instruction set",
  "author": "Anna Steele"
},
{
  "id": 83,
  "title": "Stand-alone dedicated portal",
  "author": "Adam Andrews"
},
{
  "id": 84,
  "title": "Cross-group user-facing firmware",
  "author": "Joseph Boyd"
},
{
  "id": 85,
  "title": "Progressive user-facing orchestration",
  "author": "Joshua Hurley MD"
},
{
  "id": 86,
  "title": "Profound real-time challenge",
  "author": "Susan Beck"
},
{
  "id": 87,
  "title": "Polarized encompassing model",
  "author": "Crystal Alvarez"
},
{
  "id": 88,
  "title": "Compatible real-time instruction set",
  "author": "Anna Steele"
},
{
  "id": 89,
  "title": "Stand-alone dedicated portal",
  "author": "Adam Andrews"
}
```
- Response headers:**

```
access-control-allow-origin: *
content-length: 1169
content-type: application/json
date: Sat, 14 Sep 2024 18:27:39 GMT
link: <http://library.demo.local/api/v1/books?page=9>; rel="next"
server: Werkzeug/0.14.1 Python/3.8.2
```

Efectivamente, obtenemos los libros del 80 a 89.

Finalizado