/api/v1

Default namespace

GET

/books

Parameters

Cancel

Name Description

includeISBN

boolean

(query)

Include in the results the ISBN numbers. Default=false

true

sortBy

string

(query)

Sort results using the specified parameter. Default=id

title

author

string

(query)

Return only books by the given Author.

author - Return only books by the given Author.

page

integer

(query)

To save resources and bandwidth, larger replies might be broken down into smaller pages with 10 reconds per page. The RFC5988 (Web Linking) standard in HTTP Reply Headers is used for pagination.

The “page” parameter is then used to specify the page number.

page - To save resources and bandwidth, larger replies might be broken down into smaller pages with 10 reconds per page. The <a href="http://tools.ietf.org/html/rfc5988">RFC5988 (Web Linking)</a> standard in HTTP Reply Headers is used for pagination.<br/> The "page" parameter is then used to specify the page number.

Execute

Clear

Responses

Response content type

application/json

Curl

curl -X GET "http://library.demo.local/api/v1/books?includeISBN=true&sortBy=title" -H "accept: application/json"

Request URL

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

Server response

Code Details

200

Response body

Download

[

{

"id": 0,

"title": "IP Routing Fundamentals",

"author": "Mark A. Sportack",

"isbn": "978-1578700714"

},

{

"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"

},

{

"id": 1,

"title": "Python for Dummies",

"author": "Stef Maruch Aahz Maruch",

"isbn": "978-0471778646"

}

]

Response headers

access-control-allow-origin: \*

content-length: 599

content-type: application/json

date: Fri, 11 Dec 2020 04:19:10 GMT

server: Werkzeug/0.14.1 Python/3.8.2

Responses

Code Description

200

Success

POST

/books

Parameters

Cancel

Name Description

payload \*

(body)

Edit Value

Model

{

"id": 4,

"title": "IPv6 Fundamentals",

"author": "Rick Graziani"

}

Cancel

Parameter content type

application/json

Execute

Clear

Responses

Response content type

application/json

Curl

curl -X POST "http://library.demo.local/api/v1/books" -H "accept: application/json" -H "X-API-KEY: cisco|0numfF6jS70q0CZixqn185fyHjYjhNaL9BN3p59wjSk" -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

Response body

Download

{

"id": 4,

"title": "IPv6 Fundamentals",

"author": "Rick Graziani"

}

Response headers

access-control-allow-origin: http://library.demo.local

content-length: 81

content-type: application/json

date: Fri, 11 Dec 2020 05:13:28 GMT

server: Werkzeug/0.14.1 Python/3.8.2

vary: Origin

Responses

Code Description

200

Success

Example Value

Model

{

"id": 0,

"title": "string",

"author": "string"

}

DELETE

/books/{id}

Parameters

Cancel

Name Description

id \*

integer

(path)

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|0numfF6jS70q0CZixqn185fyHjYjhNaL9BN3p59wjSk"

Request URL

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

Server response

Code Details

200

Response body

Download

{

"id": 4,

"title": "IPv6 Fundamentals",

"author": "Rick Graziani"

}

Response headers

access-control-allow-origin: http://library.demo.local

content-length: 81

content-type: application/json

date: Fri, 11 Dec 2020 05:03:26 GMT

server: Werkzeug/0.14.1 Python/3.8.2

vary: Origin

Responses

Code Description

200

Success

Example Value

Model

{

"id": 0,

"title": "string",

"author": "string"

}

GET

/books/{id}

Parameters

Cancel

Name Description

id \*

integer

(path)

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

Response body

Download

{

"id": 4,

"title": "IPv6 Fundamentals",

"author": "Rick Graziani"

}

Response headers

access-control-allow-origin: \*

content-length: 81

content-type: application/json

date: Fri, 11 Dec 2020 04:21:36 GMT

server: Werkzeug/0.14.1 Python/3.8.2

Responses

Code Description

200

Success

Example Value

Model

{

"id": 0,

"title": "string",

"author": "string"

}

PUT

/books/{id}

Parameters

Cancel

Name Description

payload \*

(body)

Edit Value

Model

{

"id": 4,

"title": ""IPv6 Fundamentals",

"author": "Rick Graziani"

}

Cancel

Parameter content type

application/json

id \*

integer

(path)

4

Execute

Clear

Responses

Response content type

application/json

Curl

curl -X PUT "http://library.demo.local/api/v1/books/4" -H "accept: application/json" -H "X-API-KEY: cisco|0numfF6jS70q0CZixqn185fyHjYjhNaL9BN3p59wjSk" -H "Content-Type: application/json" -d "{ \"id\": 4, \"title\": \"\"IPv6 Fundamentals\", \"author\": \"Rick Graziani\"}"

Request URL

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

Server response

Code Details

400

Undocumented

Error: BAD REQUEST

Response body

Download

{

"message": "Failed to decode JSON object: Expecting ',' delimiter: line 3 column 14 (char 26)"

}

Response headers

access-control-allow-origin: http://library.demo.local

content-length: 103

content-type: application/json

date: Fri, 11 Dec 2020 04:20:37 GMT

server: Werkzeug/0.14.1 Python/3.8.2

vary: Origin

Responses

Code Description

200

Success

Example Value

Model

{

"id": 0,

"title": "string",

"author": "string"

}

POST

/loginViaBasic

Parameters

Cancel

No parameters

Execute

Clear

Responses

Response content type

application/json

Curl

curl -X POST "http://library.demo.local/api/v1/loginViaBasic" -H "accept: application/json"

Request URL

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

Server response

Code Details

200

Response body

Download

{

"token": "cisco|S4rwHoXkM0HJt1Yw2UX-BbduGJuY5jXnhsPr9RkQLq4"

}

Response headers

access-control-allow-origin: http://library.demo.local

content-length: 69

content-type: application/json

date: Fri, 11 Dec 2020 04:36:03 GMT

server: Werkzeug/0.14.1 Python/3.8.2

vary: Origin

Responses

Code Description

200

Success

Headers:

Name Description Type

Authorization: Basic BASE64 encoded username:password string

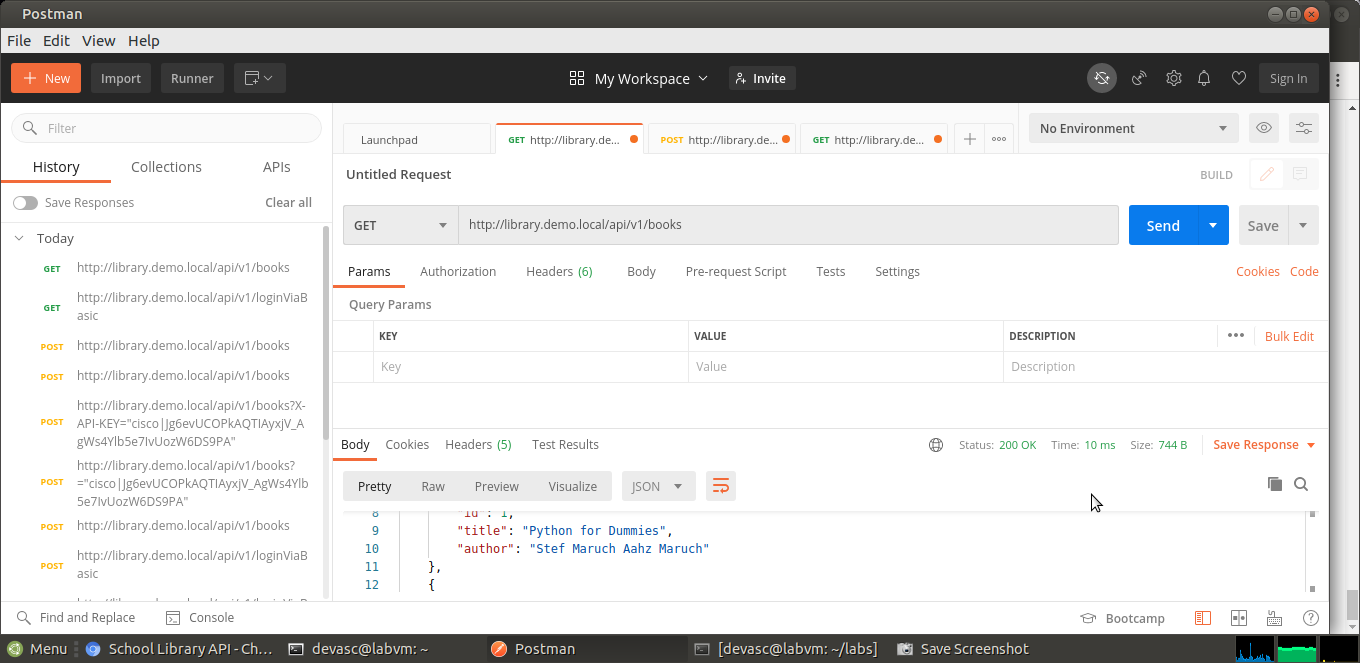
POST

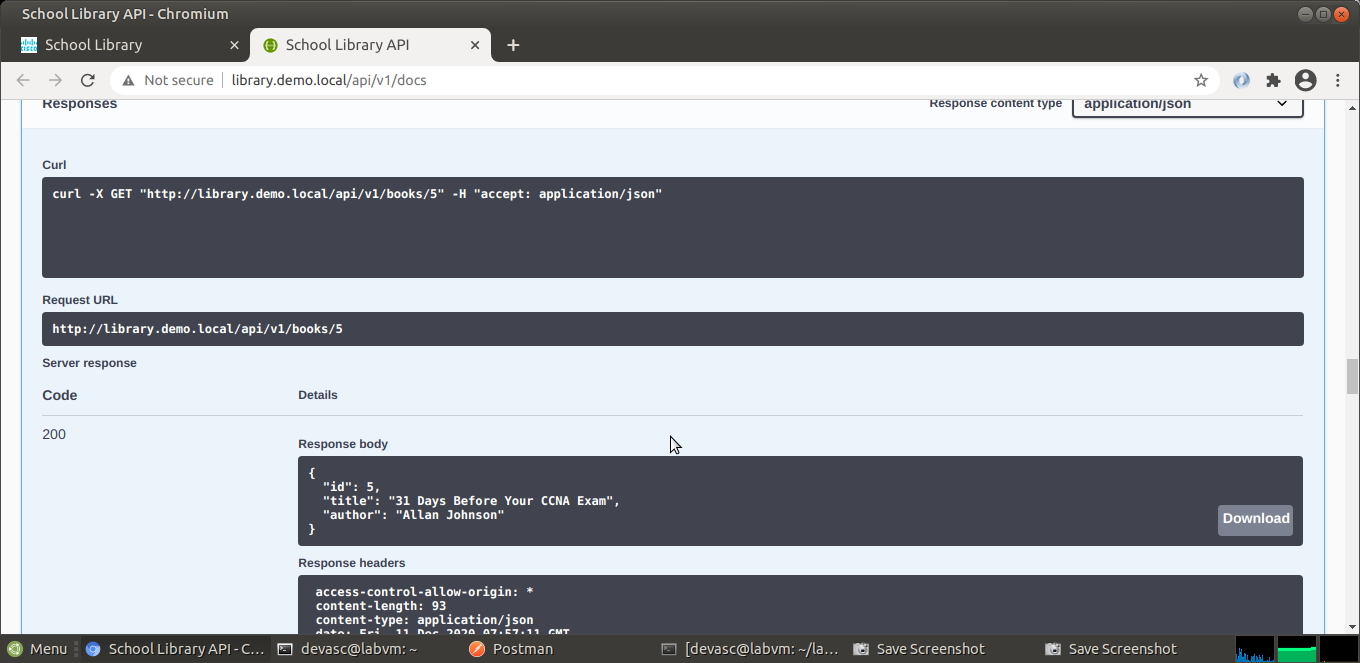
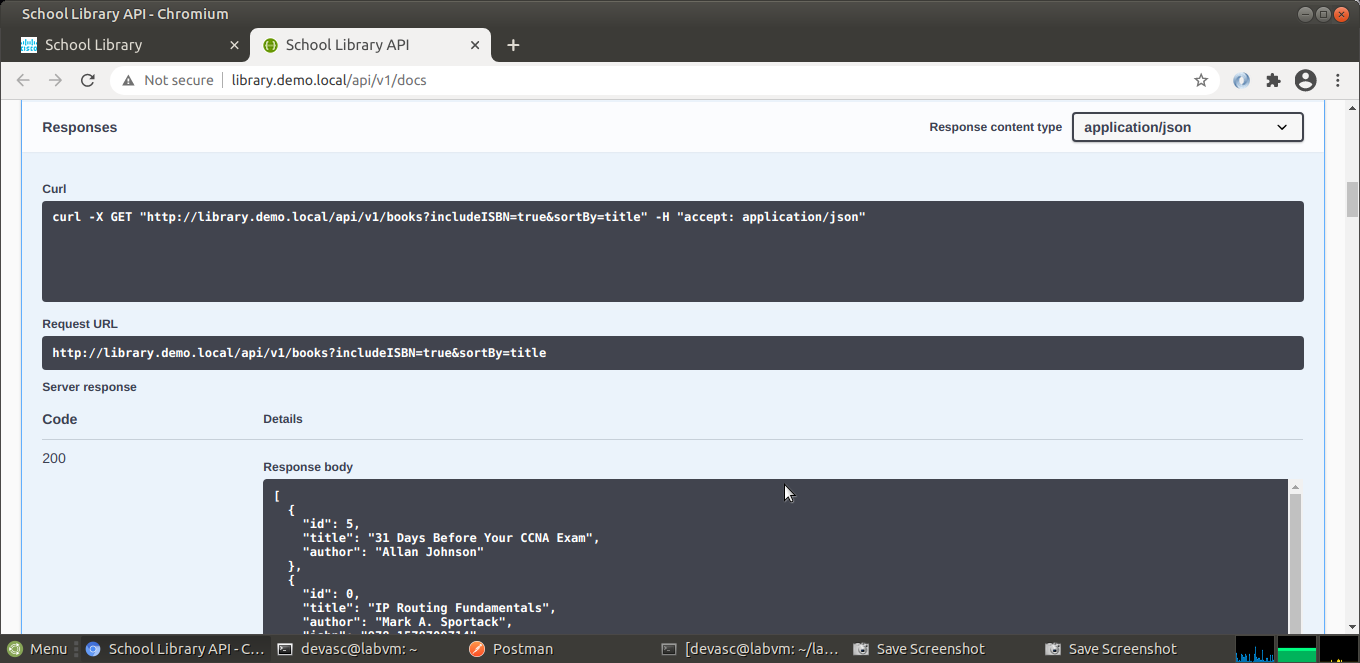
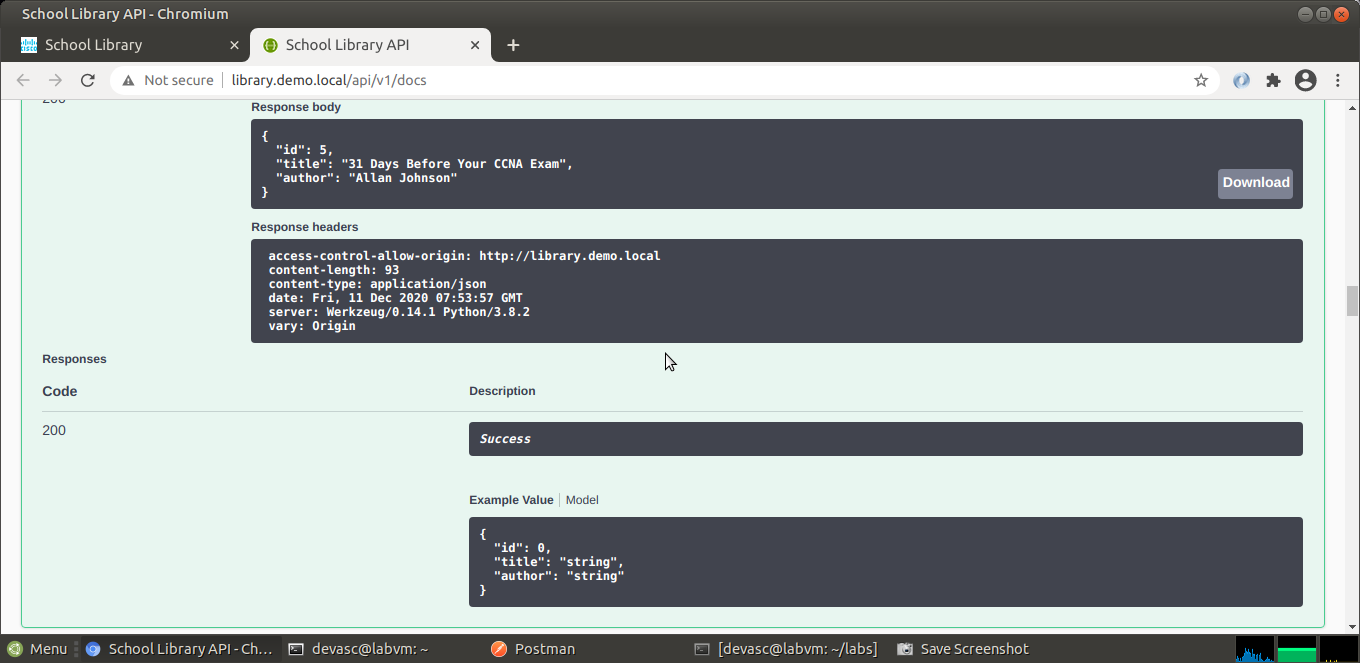
/loginViaJSON

Models

Credentials

Book



/usr/bin/env python3 /home/devasc/.vscode/extensions/ms468/pythonFiles/lib/python/debugpy/launcher 45951 -- /home/devasc/labsdevasc@labvm:~/labs/devnet-src/python$ /udevasc@labvm:~/labs/devnet-src/python$ /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 45951 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Mary Odom.

Gregory Avery

James Chavez

Krista Medina

Brent Mason

Chad Jimenez

Mary Myers

Charles Navarro

Roberto Cortez

Paul Turner

Loretta Wood

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 44327 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Sheila Webb.

Erik Adams

Timothy Rivera

Dr. Eric Callahan

Sarah Buchanan

Elizabeth Cisneros

Tanner Blair

Anthony Hanna

Luis Smith

Kendra Chapman

Allison Bryant

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 37955 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Alyssa Rodriguez.

Miranda Smith

Gregory Smith

Jasmine Rogers

Jeffrey Wood

Tiffany Tucker

Drew Doyle

Jonathan Cook

Philip Kaiser

Jacob Williams

Raymond Logan

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 32891 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Kimberly Gillespie.

Kimberly Lynn

Robert Little

Cody Powell

Hannah Barton

Thomas Smith

Adrian Wilson

Paul Riddle

Scott Hunt

Andrew Taylor

Devin Henry

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 35445 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Cole Phillips.

Clayton Carson

Lauren Russell

Roger Sims

Donna Roach

David Wright

Sherry Cunningham

Kenneth Fischer

Jacqueline Martin

Jennifer Flores

Tina Bailey

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 40525 -- /home/devasc/labs/devnet-src/python/books-add.py

File "/home/devasc/labs/devnet-src/python/books-add.py", line 4

//for i in range(10):

^

SyntaxError: invalid syntax

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 43599 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Fernando Kelley.

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 38323 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Aaron Rich.

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 42999 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Steven Wood.

Traceback (most recent call last):

File "/home/devasc/labs/devnet-src/python/books-add.py", line 17, in <module>

getAuthToken()

File "/home/devasc/labs/devnet-src/python/books-add.py", line 11, in getAuthToken

r = requests.post(f"{APIHOST}/api/v1/loginViaBasic", auth = authCreds )

NameError: name 'requests' is not defined

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 44725 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is John Nelson.

success

devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 40289 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Richard Martin.

Traceback (most recent call last):

File "/home/devasc/labs/devnet-src/python/books-add.py", line 19, in <module>

getAuthToken()

File "/home/devasc/labs/devnet-src/python/books-add.py", line 18, in getAuthToken

raise Exception(f"Status code {r.status\_code} and text {r.text}, while trying to Auth.")

Exception: Status code 401 and text Unauthorized Access, while trying to Auth.

devasc@labvm:~/labs/devnet-src/python$ ^C

devasc@labvm:~/labs/devnet-src/python$ devasc@labvm:~/labs/devnet-src/python$ cd /home/devasc/labs/devnet-src/python ; /usr/bin/env python3 /home/devasc/.vscode/extensions/ms-python.python-2020.6.88468/pythonFiles/lib/python/debugpy/launcher 35931 -- /home/devasc/labs/devnet-src/python/books-add.py

My name is Lawrence Blair.

success

devasc@labvm:~/labs/devnet-src/python$ import requests

import json

from faker import Faker

fake=Faker()

print('My name is {}.'.format(fake.name()))

#for i in range(10):

# print(fake.name())

APIHOST = "http://library.demo.local"

LOGIN = "cisco"

PASSWORD = "Cisco123!"

def getAuthToken():

authCreds = (LOGIN, PASSWORD)

r = requests.post(f"{APIHOST}/api/v1/loginViaBasic", auth = authCreds )

if r.status\_code == 200:

print("success")

return r.json()["token"]

else:

raise Exception(f"Status code {r.status\_code} and text {r.text}, while trying to Auth.")

getAuthToken()

def addBook(book, apiKey):

r = requests.post(

f"{APIHOST}/api/v1/books",

headers = {

"Content-type": "application/json",

"X-API-Key": apiKey

},

data = json.dumps(book)

)

if r.status\_code == 200:

print(f"Book {book} added.")

else:

raise Exception(f"Error code {r.status\_code} and text {r.text}, while tryingto add book {book}.")

apiKey = getAuthToken()

fake = Faker()

for i in range(6, 106):

fakeTitle = fake.catch\_phrase()

fakeAuthor = fake.name()

fakeISBN = fake.isbn13()

book = {"id":i, "title": fakeTitle, "author": fakeAuthor, "isbn": fakeISBN}

# add the new random "fake" book using the API

addBook(book, apiKey)