**LAB 2 – EXPLORE REST APIs WITH API SIMULATOR AND POSTMAN**

LAB NETACAD : CISCO DEVNET 4.5.5-lab---explore-rest-apis-with-api-simulator-and-postman

Part 1: Explore API Documentation Using the API Simulator

Document your findings in 3 steps:

* Task preparation and implementation:

Step 1-5 are straight forward

Text

Description automatically generated

First API call

Text

Description automatically generated

API call with curl

Post a book response:

Text

Description automatically generated

Code 200 > Success!

Table

Description automatically generatedVisible in our books nowText

Description automatically generated

Also Get books shows these

curl -X DELETE "http://library.demo.local/api/v1/books/4" -H "accept: application/json" -H "X-API-KEY: cisco|7F8RljSITKTpnaa\_YJp8fCcgkUrmauZ4wlx6vopD1yk"

Deleting books by id through curl instead of dashboard

* Task troubleshooting: curl api call wasn’t working -> worked after restart & sudo apt update
* Task verification: Text

  Description automatically generated

Part 2: Use Postman to Make API Calls to the API Simulator

Document your findings in 3 steps:

* Task preparation and implementation:

Graphical user interface, text, application, email

Description automatically generated

POST request with api key

Graphical user interface, text, application, email

Description automatically generatedConfirm by using GET books, id 4 is back in there now. We removed it in the last excerciseGraphical user interface, text, application, email

Description automatically generated

Using parameters to get the ISBN and sort by author, its now in the request url aswell

* Task troubleshooting: Method not allowed but was using get instead of post when trying to retrieve API Key
* Task verification: Screenshots in task execution

Part 3: Use Python to Add 100 Books to the API Simulator

Document your findings in 3 steps:

* Task preparation and implementation:

A computer screen capture

Description automatically generated with low confidence

All faker methods, some of them are used in the python script to add books, like isbn, catch phrase, name

Exercise:

print('My name is {}'.format(fake.name()) + 'and I wrote "{}"'.format(fake.catch\_phrase()) + ' (ISBN {})'.format(fake.isbn13()) )

Result:



* Task troubleshooting: None
* Task verification: Graphical user interface, application

  Description automatically generated