

GREENDECK

TASK 1

Task: Create a RESTful API
using Python and host it locally
using Docker.

CRUD OPERATIONS POSTMAN
SCREENSHOTS

1. READ/LIST Operation

- API Endpoint: /api/Products
- Method: GET

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

- Launchpad:** A green button to start a new request.
- Environment:** A dropdown menu showing "No Environment".
- Request Bar:** Displays the method "GET" and the URL "http://localhost:5500/api/Products". It includes "Send" and "Save" buttons.
- Tabs:** A row of tabs for "Params", "Authorization", "Headers (6)", "Body", "Pre-request Script", "Tests", and "Settings".
- Query Params Table:** A table with columns "KEY", "VALUE", and "DESCRIPTION". It contains one row with "Key" and "Value".
- Response Section:** Includes tabs for "Body", "Cookies", "Headers (4)", and "Test Results". It shows a status of "200 OK", a time of "4.59 s", and a size of "343.33 KB".
- Response Body:** A JSON array of product data, displayed in "Pretty" format. The JSON is as follows:

```
1 {
2   "Result": [
3     {
4       "Offer Price": 12,
5       "Product Name": "Jellycat Blossom Tulip Bunny Grabber, Pink"
6     },
7     {
8       "Offer Price": 1080,
9       "Product Name": "iCandy Peach All Terrain Pushchair and Carrycot, Forest"
10    },
11    {
12      "Offer Price": 50,
13      "Product Name": "Simon Carter Laser Engraved Button Cufflinks, Gunmetal"
14    },
15    {
16      "Offer Price": 15,
17      "Product Name": "Oasis Broderie Short Sleeve Top, Black"
18    }
19  ]
20 }
```
- Footer:** Includes a "Bootcamp" logo and buttons for "Build", "Browse", and other utility icons.

2. CREATE/ADD/INSERT Operation

- API Endpoint: /api/Products/add
- Method: POST

The screenshot shows the Postman interface with a POST request to `http://localhost:5500/api/Products/add`. The request body is a JSON object with the following fields:

```
{  "Product Name" : "Multi Pocket Joggers",  "Brand Name" : "Allen Solley",  "Regular Price" : 12,  "Offer Price" : 6,  "Currency" : "GBP",  "Classification_11" : "Mens",  "Classification_12" : "Cloths",  "Classification_13" : "",  "Classification_14" : "",  "Image URL" : ""}
```

The response is a JSON object indicating success:

```
{  "Object Added Successfully",  "201"}
```

The status is 200 OK, Time is 508 ms, and Size is 179 B.

The screenshot shows the MongoDB Compass interface. The left sidebar shows the database structure:

- FirstDB
 - Titanic

The main panel displays the collection **FirstDB.Titanic** with a collection size of 1.85MB, 5001 total documents, and 84KB total index size. The **Find** tab is active, showing a filter: `{name : "Multi Pocket Joggers"}`. The query results show 1 document:

```
{  "_id": ObjectId("60007c89b372b9a4806710be"),  "name": "Multi Pocket Joggers",  "brand_name": "Allen Solley",  "regular_price_value": 12,  "offer_price_value": 6,  "currency": "GBP",  "classification_11": "Mens",  "classification_12": "Cloths",  "classification_13": "",  "classification_14": "",  "image_url": ""}
```

3. UPDATE Operation

- API Endpoint: /api/Products/update/<id>
- Method: PUT

The screenshot displays the MyFirstCluster web interface. At the top, the header shows 'MyFirstCluster' with a logo, and on the right, 'VERSION 4.2.11' and 'REGION AWS Singapore (ap-southeast-1)'. Below the header is a navigation bar with tabs: Overview, Real Time, Metrics, Collections (selected), Search, Profiler, Performance Advisor, Online Archive, and Command Line Tools. Under the 'Collections' tab, it shows 'DATABASES: 1' and 'COLLECTIONS: 1'. On the left sidebar, there is a '+ Create Database' button and a search bar labeled 'NAMESPACES'. Below the search bar, a tree view shows 'FirstDB' expanded, with 'Titanic' listed underneath. The main content area is titled 'FirstDB.Titanic' and shows 'COLLECTION SIZE: 1.85MB', 'TOTAL DOCUMENTS: 5001', and 'INDEXES TOTAL SIZE: 84KB'. Below this, there are tabs for 'Find' (selected), 'Indexes', 'Schema Anti-Patterns 0', 'Aggregation', and 'Search Indexes 0'. A 'FILTER' input field contains the JSON query '{ "filter": "example" }'. To the right of the filter are 'Find' and 'Reset' buttons. Above the results is an 'INSERT DOCUMENT' button. The results section is titled 'QUERY RESULTS 1-20 OF MANY' and displays a single document in JSON format:

```
{  "_id": ObjectId("5ff0996395ddf785590c5207"),  "name": "Jellycat Blossom Tulip Bunny Grabber, Pink",  "brand_name": "jellycat",  "regular_price_value": 12,  "offer_price_value": 12,  "currency": "GBP",  "classification_l1": "baby & child",  "classification_l2": "soft toys",  "classification_l3": "",  "classification_l4": "",  "image_url": "https://johnlewis.scene7.com/is/image/JohnLewi/237070760?"}
```

 A green chat bubble icon is visible in the bottom right corner of the interface.

Offer Price before Update is “12”.

Launchpad GET http://localhost:5500/api/Produ... PUT http://localhost:5500/api/Produ... No Environment

Untitled Request BUILD

PUT http://localhost:5500/api/Products/update/5ffd996395ddf705590c5207 Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "Regular Price" : 12,
3   "Offer Price" : 8
4 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 467 ms Size: 181 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "Object Successfully Updated",
3   201
4 }
```

Bootcamp Build Browse

MyFirstCluster VERSION 4.2.11 REGION AWS Singapore (ap-southeast-1)

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Command Line Tools

DATABASES: 1 COLLECTIONS: 1 VISUALIZE YOUR DATA REFRESH

+ Create Database

NAMESPACES

FirstDB

Titanic

FirstDB.Titanic

COLLECTION SIZE: 1.85MB TOTAL DOCUMENTS: 5001 INDEXES TOTAL SIZE: 84KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER {"filter": "example"} Find Reset

QUERY RESULTS 1-20 OF MANY

```
_id: ObjectId("5ffd996395ddf705590c5207")
name: "Jellycat Blossom Tulip Bunny Grabber, Pink"
brand_name: "jellycat"
regular_price_value: 12
offer_price_value: 8
currency: "GBP"
classification_l1: "baby & child"
classification_l2: "soft toys"
classification_l3: ""
classification_l4: ""
image_url: "https://johnlewis.scene7.com/is/image/JohnLewis/237070760?"
```

Offer Price after the update is “8”.

4. DELETE Operation

- API Endpoint: `/api/Products/delete/<id>`
- Method: DELETE

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Command Line Tools

DATABASES: 1 COLLECTIONS: 1

+ Create Database

NAMESPACES

FirstDB

Titanic

FirstDB.Titanic

COLLECTION SIZE: 1.85MB TOTAL DOCUMENTS: 5001 INDEXES TOTAL SIZE: 84KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER {name: "Multi Pocket Joggers"}

Find Reset

QUERY RESULTS 1-1 OF 1

```
{
  "_id": "ObjectId('5ffeb4db9cd1e11f45827421')",
  "name": "Multi Pocket Joggers",
  "brand_name": "Allen Solley",
  "regular_price_value": 20,
  "offer_price_value": 15,
  "currency": "GBP",
  "classification_11": "Mens",
  "classification_12": "Cloths",
  "classification_13": "",
  "classification_14": "",
  "image_url": ""
}
```

Products is present before deletion operation.

Launchpad GET http://localhost:5500/api/Produ... DEL http://localhost:5500/api/Produ... + ...

No Environment

Untitled Request

BUILD

DELETE http://localhost:5500/api/Products/delete/5ffeb4db9cd1e11f45827421

Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 160 ms Size: 182 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2 "Object Deleted Successfully!",
3 204
4
```

Product deleted successfully.

FirstDB.Titanic

COLLECTION SIZE: 1.85MB

TOTAL DOCUMENTS: 5001

INDEXES TOTAL SIZE: 84KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes ●

INSERT DOCUMENT

FILTER {name : "Multi Pocket Joggers"}

Find

Reset

QUERY RESULTS 0

Atlas Blog

Contact Sales

5. FIND Operation

- API Endpoint: `/api/Products/find/<id>`
- Method: GET

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

- Request:** Method: GET, URL: `http://localhost:5500/api/Products/find/5ffd996395ddf705590c5207`
- Response:** Status: 200 OK, Time: 269 ms, Size: 548 B
- Body:** JSON response showing product details for ID `5ffd996395ddf705590c5207`.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
{
  "_id": {
    "$oid": "5ffd996395ddf705590c5207"
  },
  "name": "Jellycat Blossom Tulip Bunny Grabber, Pink",
  "brand_name": "jellycat",
  "regular_price_value": 12,
  "offer_price_value": 12,
  "currency": "GBP",
  "classification_11": "baby & child",
  "classification_12": "soft toys",
  "classification_13": "",
  "classification_14": "",
  "image_url": "https://johnlewis.scene7.com/is/image/JohnLewis/237070760?"
}
```

Product found.

The Command Prompt window shows the following output:

```
Command Prompt - docker run -p 5500:5500 flaskapp1
172.17.0.1 - - [14/Jan/2021 17:05:25] "GET /api/Products/find/asdf345fsdfs HTTP/1.1" 500 -
172.17.0.1 - - [14/Jan/2021 17:08:52] "PUT /api/Products/update/5ffd996395ddf705590c5207 HTTP/1.1" 400 -
172.17.0.1 - - [14/Jan/2021 17:09:12] "PUT /api/Products/update/5ffd996395ddf705590c5207 HTTP/1.1" 200 -
172.17.0.1 - - [14/Jan/2021 17:11:48] "DELETE /api/Products/delete/5ffeb4db9cd1e11f45827421 HTTP/1.1" 200 -
Flask_RestAPI.py:79: DeprecationWarning: insert is deprecated. Use insert_one or insert_many instead.
  add = mongo.db.Titanic.insert({'name' : name, 'brand_name' : brand, 'regular_price_value' : price, 'offer_price_value' : offer,
172.17.0.1 - - [14/Jan/2021 17:16:57] "POST /api/Products/add HTTP/1.1" 200 -
```


Please correct me or let me know, If I've made any mistake or something is not clear through the screenshots.

Thanks,

Aditya Sisodiya