

REST API with Flask, Docker, PyCharm, Postman and MySQL

GitHub repository URL:

<https://github.com/Sowmyak91/RestAPIDockerFlaskPycharm>

1) postman listing all records for cities data

The screenshot shows the Postman interface with a GET request to `http://0.0.0.0:5000/api/v1/cities`. The response is a JSON array of two city objects. The status is 200 OK, time is 24 ms, and size is 41.83 KB.

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "fldName": "Tokyo",
5     "fldLat": 35.6850,
6     "fldLong": 139.7514,
7     "fldCountry": "Japan",
8     "fldAbbreviation": "JPN",
9     "fldCapitalStatus": "primary",
10    "fldPopulation": 35676000
11  },
12  {
13    "id": 2,
14    "fldName": "New York",
15    "fldLat": 40.6943,
16    "fldLong": -73.9249,
17    "fldCountry": "United States",
18    "fldAbbreviation": "USA",
19    "fldCapitalStatus": "NA",
```

2) postman showing one record with a get request.

The screenshot displays the Postman interface for a GET request. The request is configured with the method 'GET' and the URL 'http://0.0.0.0:5000/api/v1/cities/200'. The 'Body' tab is selected, showing a JSON response. The response status is '200 OK' with a time of '28 ms' and a size of '336 B'. The JSON body contains details for a city with ID 200, named Jacksonville.

Search Postman

Overview [CONFLICT] GET MyRe... Release Notes GET CitiesBrowse + ... No Environment

PyCharmCities / CitiesBrowse Save ...

GET http://0.0.0.0:5000/api/v1/cities/200 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

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

1

Body Cookies Headers (4) Test Results Status: 200 OK Time: 28 ms Size: 336 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 200,
3   "fldName": "Jacksonville",
4   "fldLat": 30.3322,
5   "fldLong": -81.6749,
6   "fldCountry": "United States",
7   "fldAbbreviation": "USA",
8   "fldCapitalStatus": "NA",
9   "fldPopulation": 1156291
10 }
11
12
```

3) postman creating a record by showing the post request and the results in the record listing:

Created a new record using post request, with new name and country.

The screenshot displays the Postman interface for a POST request. The request is directed to `http://0.0.0.0:5000/api/v1/cities`. The body is formatted as JSON, containing the following data:

```
{
  "fldName": "Pittsburg",
  "fldLat": 19.4424,
  "fldLong": -99.1310,
  "fldCountry": "United States",
  "fldAbbreviation": "US",
  "fldCapitalStatus": "primary",
  "fldPopulation": 19028000
}
```

The response status is **201 CREATED**, with a time of 18 ms and a size of 150 B. A notification box states: "201 CREATED. The request has been fulfilled and resulted in a new resource being created."

Results in record listing after adding new record using put request.

The screenshot shows the Postman interface for a GET request to `http://0.0.0.0:5000/api/v1/cities`. The request is successful with a status of 200 OK, a time of 16 ms, and a size of 42.01 KB. The response body is a JSON array of city records, displayed in a pretty-printed format. The records include details such as id, fldName, fldLat, fldLong, fldCountry, fldAbbreviation, fldCapitalStatus, and fldPopulation.

```
2353     "id": 236,  
2354     "fldName": "Qinhuangdao",  
2355     "fldLat": 39.9304,  
2356     "fldLong": 119.6200,  
2357     "fldCountry": "China",  
2358     "fldAbbreviation": "CHN",  
2359     "fldCapitalStatus": "NA",  
2360     "fldPopulation": 1003000  
2361   },  
2362   {  
2363     "id": 237,  
2364     "fldName": "Pittsburg",  
2365     "fldLat": 19.4424,  
2366     "fldLong": -99.1310,  
2367     "fldCountry": "United States",  
2368     "fldAbbreviation": "US",  
2369     "fldCapitalStatus": "primary",  
2370     "fldPopulation": null  
2371   }  
2372 ]
```

4) postman creating a record by showing the put request and the results in the record listing:

Edited the record using put request. fldPopulation is edited in the below record.

The screenshot shows the Postman interface for a PUT request. The request is sent to `http://0.0.0.0:5000/api/v1/cities/237`. The body is a JSON object with the following fields: `id` (237), `fldName` (Pittsburg), `fldLat` (19.4424), `fldLong` (-99.1310), `fldCountry` (United States), `fldAbbreviation` (US), `fldCapitalStatus` (primary), and `fldPopulation` (123456). The response status is 200 OK, with a time of 11 ms and a size of 145 B. The response body is shown in the 'Body' tab, displaying the updated JSON object.

```
1 {
2   "id": 237,
3   "fldName": "Pittsburg",
4   "fldLat": 19.4424,
5   "fldLong": -99.1310,
6   "fldCountry": "United States",
7   "fldAbbreviation": "US",
8   "fldCapitalStatus": "primary",
9   "fldPopulation": 123456
10 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 11 ms Size: 145 B Save Response

Pretty Raw Preview Visualize JSON

```
1
```

Bootcamp Runner Trash

Results in the record listing after editing the record using put request

The screenshot shows the Postman interface with a GET request to `http://0.0.0.0:5000/api/v1/cities`. The response status is 200 OK, and the body is a JSON array of city records. The records include fields for `id`, `fldName`, `fldLat`, `fldLong`, `fldCountry`, `fldAbbreviation`, `fldCapitalStatus`, and `fldPopulation`.

```
[{"id": 235, "fldName": "Pittsburg", "fldLat": 39.9304, "fldLong": 119.6200, "fldCountry": "China", "fldAbbreviation": "CHN", "fldCapitalStatus": "NA", "fldPopulation": 1003000}, {"id": 237, "fldName": "Pittsburg", "fldLat": 19.4424, "fldLong": -99.1310, "fldCountry": "United States", "fldAbbreviation": "US", "fldCapitalStatus": "primary", "fldPopulation": 123456}]
```

5) postman delete a record by showing the delete request and the results in the record listing:

Deleted record of id: 237 using delete request.

The screenshot displays the Postman interface for a DELETE request. The request is configured with the method 'DELETE' and the URL 'http://0.0.0.0:5000/api/v1/cities/237'. The 'Body' tab is selected, showing a JSON payload for a city record with id 237. The response status is '200 OK'.

Request Details:

- Method: DELETE
- URL: http://0.0.0.0:5000/api/v1/cities/237
- Body (JSON):

```
{  "id": 237,  "fldName": "Pittsburg",  "fldLat": 19.4424,  "fldLong": -99.1310,  "fldCountry": "United States",  "fldAbbreviation": "US",  "fldCapitalStatus": "primary",  "fldPopulation": 123456}
```

Response Details:

- Status: 200 OK
- Time: 13 ms
- Size: 145 B

Results in the record listing after deleting the record using delete.

The screenshot shows the Postman interface for a REST client. The request is a GET to `http://0.0.0.0:5000/api/v1/cities`. The response status is 200 OK, with a time of 20 ms and a size of 41.83 KB. The response body is displayed in JSON format, showing a list of two cities.

```
2342 {
2343   "id": 235,
2344   "fldName": "Cologne",
2345   "fldLat": 50.9300,
2346   "fldLong": 6.9500,
2347   "fldCountry": "Germany",
2348   "fldAbbreviation": "DEU",
2349   "fldCapitalStatus": "NA",
2350   "fldPopulation": 1004000
2351 },
2352 {
2353   "id": 236,
2354   "fldName": "Qinhuangdao",
2355   "fldLat": 39.9304,
2356   "fldLong": 119.6200,
2357   "fldCountry": "China",
2358   "fldAbbreviation": "CHN",
2359   "fldCapitalStatus": "NA",
2360   "fldPopulation": 1003000
2361 }
2362 ]
```