

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

GitHub repository URL:

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

GitHub repository URL for new CSV:

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

### 1) postman listing all records

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, and the response size is 41.83 KB.

```
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",
```

Overview GET OscarBrowse + ... No Environment

PyCharmOscar / OscarBrowse Save ...

GET http://localhost:5000/api/v1/oscars Send

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

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<p>Body Cookies Headers (4) Test Results Status: 200 OK Time: 22 ms Size: 11.55 KB Save Response</p> <p>Pretty Raw Preview Visualize JSON</p> <pre> 1  { 2    { 3      "id": 1, 4      "Index": 1, 5      "Year": 1928, 6      "Age": 22, 7      "Name": "Janet Gaynor", 8      "Movie": "Seventh Heaven", 9      "Column_6": "Street Angel and Sunrise: A Song of Two Humans" 10   }, 11   { 12     "id": 2, 13     "Index": 2, 14     "Year": 1929, 15     "Age": 37, 16     "Name": "Mary Pickford", 17     "Movie": "Coquette", 18     "Column_6": null 19   }, 20   { 21     "id": 3,</pre>				

## 2) postman showing one record with a get request.

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
```

Overview GET OscarBrowse + ... No Environment

PyCharmOscar / OscarBrowse Save ...

GET http://localhost:5000/api/v1/oscars/79 Send

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

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
-----	-------	-------------	-----	-----------

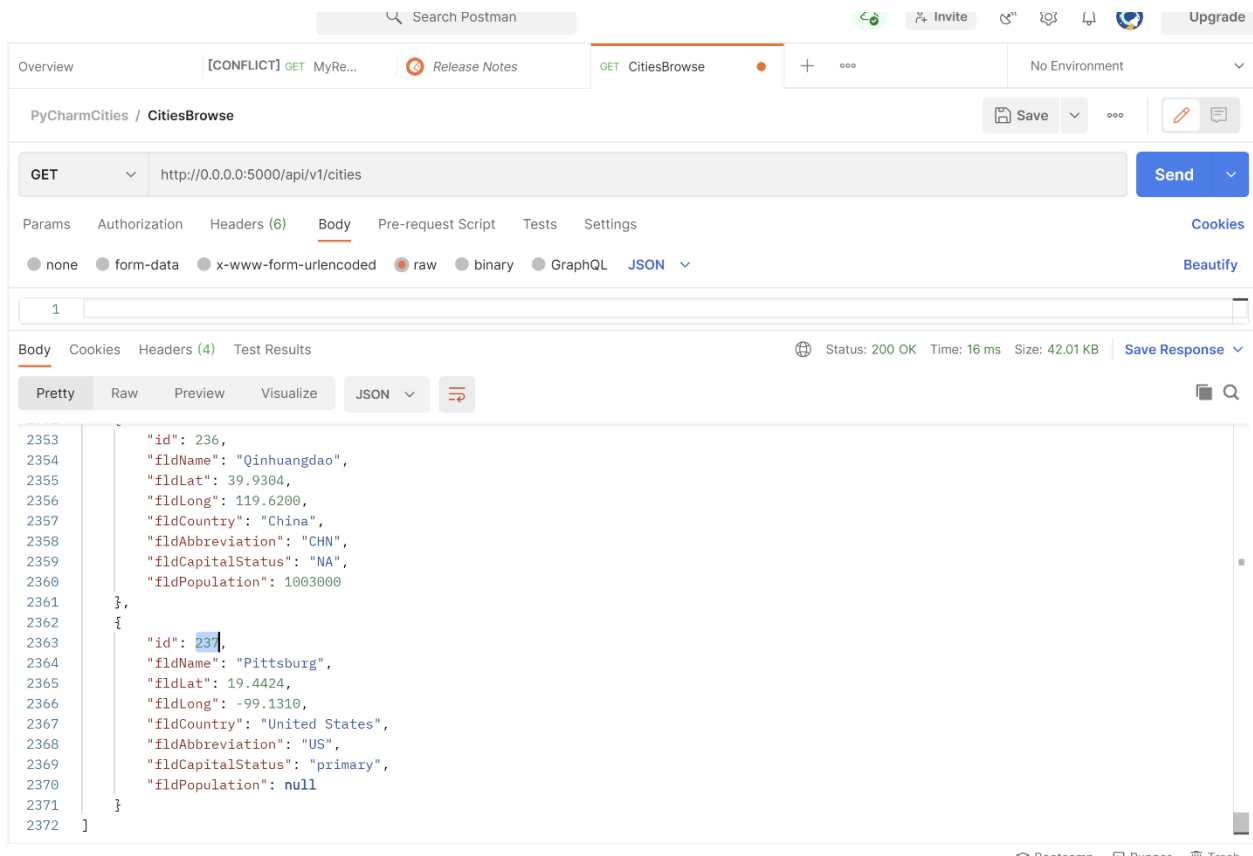
Body Cookies Headers (4) Test Results Status: 200 OK Time: 20 ms Size: 280 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 79,
3   "Index": 79,
4   "Year": 2006,
5   "Age": 29,
6   "Name": "\"Reese Witherspoon\"",
7   "Movie": "\"Walk the Line\"",
8   "Column_6": null
9 }
10
11
```

### 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.



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

The screenshot displays the Postman application interface. At the top, the 'Overview' tab is selected, showing the request details for a POST method to the endpoint `http://localhost:5000/api/v1/oscars`. The request body is set to 'JSON' and contains a JSON object with the following data:

```
{
  "Index": 90,
  "Year": 2020,
  "Age": 22,
  "Name": "\Janet Gaynor",
  "Movie": "\Seventh Heaven",
  "Column_6": "Street Angel and Sunrise: A Song of Two Humans"
}
```

Below the request, the 'Body' tab is selected, showing the response body. The response is a JSON object with the following data:

```
{
  "id": 89,
  "Index": 89,
  "Year": 2016,
  "Age": 26,
  "Name": "\Brie Larson",
  "Movie": "\Room",
  "Column_6": null
}
```

The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 15 ms, and a size of 11.55 KB.

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 response is a JSON array of two city records. The first record is for Qinhuangdao, China, with a population of 1003000. The second record is for Pittsburgh, United States, with a population of null. The status is 200 OK, time is 16 ms, and size is 42.01 KB.

```
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:

```
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 }
```

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 same JSON object as the request body.

At the bottom right, there are links for 'Bootcamp', 'Runner', and '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 first record is for China, and the second is for Pittsburg, United States.

```
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": 123456  
2371 ]  
2372 ]
```



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, and the response body is empty, indicating successful deletion.

**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
- Response Body: (Empty)

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 a JSON array containing two city records.

```
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 ]
```