

[MY GitHub](#)

Flask REST API project

Hi there! In this project I created RESTful API services, two separated microservices where one is a service for users, and another one is for products. It's developed in Python's framework Flask. In addition, I used HTML, CSS, BOOTSTRAP, and JAVASCRIPT for modest frontend pages. My chosen theme for this site is "Ducati Motor Holding".

Let's start and take a look at this project

This explanation is for Windows OS users

To make this project easier to start, I packed it to Docker. First of all, you have to start your "Docker Desktop" on the side, it just has to be started, and that's all (if you don't have one, [download](#) it from the official site). After that, run PowerShell as administrator, and move to the directory where is project. *for example:*

```
cd "C:\Users\milja\VS Code Projects\FlaskAPI"
```

Then, start Docker

```
docker-compose up
```

Congratulations! Project is successfully started!

Now, go to your web browser, at URL field type localhost:5000, and the login page will be there. This is the first microservice for users. We assume you are on your first time-on-site and don't have your account, and if you attempt to log in anyway, it won't pass. You should go to the Registration page, register and now you are a legal customer :)

After successfully login in, site gonna relocate you to the next microservice where are all dream Ducati Motorcycles. This page is not hard coded, all products are displayed directly from the database, with the help of API and JavaScript.

Products side API

And now, there is a point of this project, easy and fast maintaining site. With the program, which is called "Postman" (of course, if you don't have one, go [here](#) and catch it) you can preview, add, delete and update all those angry machines. Open Postman, at URL field, type and try next commands, respectively url links:

- To preview motorcycles (*choose GET method*)

```
GET    http://127.0.0.1:5001/getmotorcycle
```

Body should be empty.

- To add motorcycle (*choose POST method*)

```
POST    http://127.0.0.1:5001/newmotorcycle
```

In body, choose "raw" radio button, and for example type this:

```
{
  "id": 55,
  "name": "KAWASAKI",
  "power": "150 hp",
  "torque": "68 lb-ft",
  "dry_weight": "200 kg",
  "img_url": "https://autoblog.rs/gallery/108/kawasaki-zx-25r.jpg"
}
```

and click SEND

- To update motorcycle (*choose PUT method*)

```
PUT    http://127.0.0.1:5001/updatemotorcycle/55
```

That last number 55 in URL means ID of product, which you want to update. In body, choose "raw" radio button, and for example type this:

```
{
  "power": "250 hp",
  "torque": "90 lb-ft"
}
```

- To delete motorcycle (*choose DELETE method*)

```
DELETE http://127.0.0.1:5001/deletemotorcycle/55
```

Also, that last number 55 in URL means ID of product, which you want to delete. Body should be empty.

Users side API

As I showed at the beginning, you can register, respectively POST new user in database via HTML forms. Also there are GET and DELETE available methods from POSTMAN for this API.

- To preview users (*choose GET method*)

```
GET    http://127.0.0.1:5000/getusers
```

Body should be empty.

- To delete user (*choose DELETE method*)

```
DELETE http://127.0.0.1:5000/deleteuser/9
```

Also, that last number 9 in URL means ID of user, which you want to delete. Body should be empty.