

Flask programmer documentation

This is web application that has an assistant database.

This application is written in python.

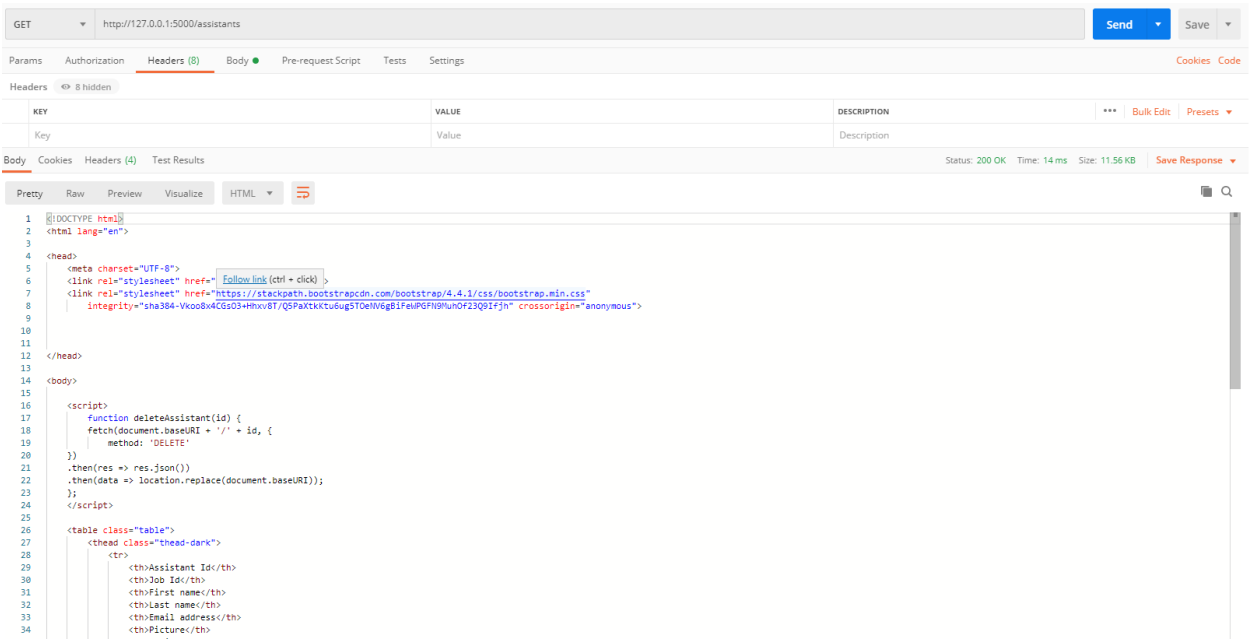
MVT(Model View Template) software design pattern was used in this application.

This application has crud endpoints:



- GET /assistants – displaying source,
- GET /assistants/create – displaying create form,
- POST /assistants – storing source,
- GET /assistants/update – displaying update form,
- PUT /assistants/{id} – updates source,
- DELETE /assistants/{id} – deletes source

I will test them in postman and describe them.

Get/assistants- this endpoint just display all assistants from database.

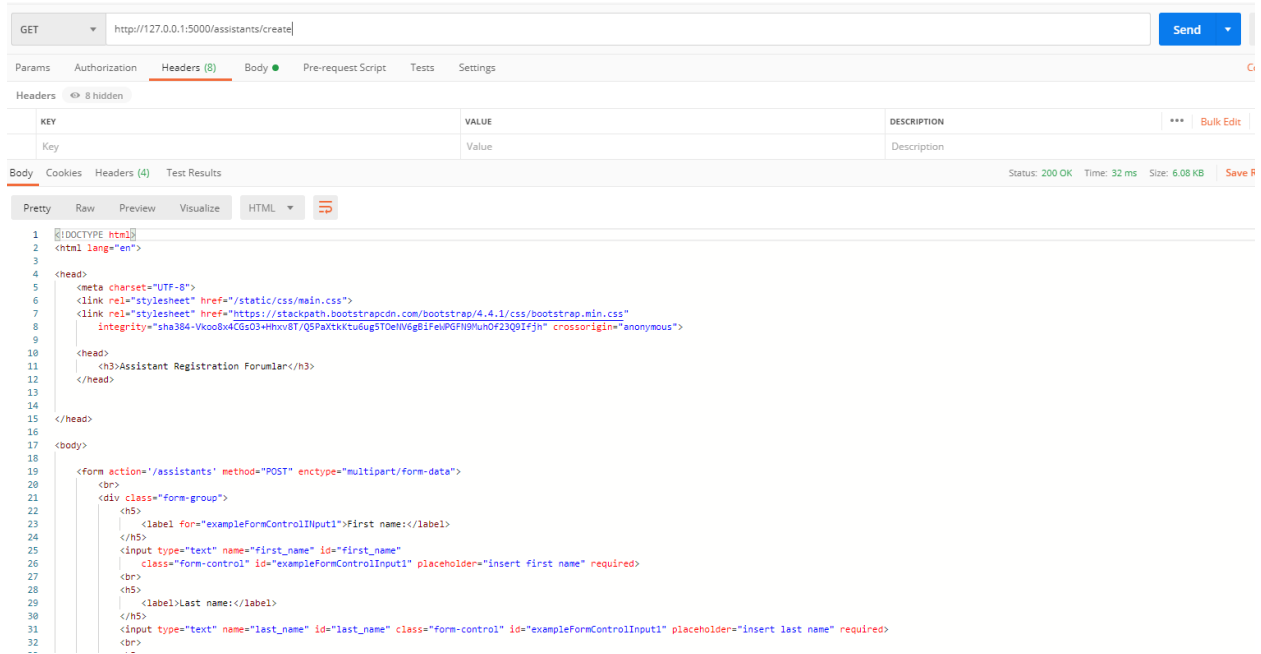


Here is ss from web browser:

Assistant Id	Job Id	First name	Last name	Email address	Picture	Action
1	4th Grade Teacher	Iza	Piechowiak	iza@gmail.com		<div>Delete</div> <div>Update</div>
2	4th Grade Teacher	Janusz	Kowalski	janusz@gmail.com		<div>Delete</div> <div>Update</div>
3	4th Grade Teacher	Jacek	Kaczmarek	kaczmarek.jacek10@gmail.com		<div>Delete</div> <div>Update</div>

Get/assistants/create- this endpoint just shows page where you can enter new assistant data and then save them.

SS form postman:



SS form web browser:

Assistant Registration Forumlar

First name:

Last name:

Email:

Job name:

Upload picture:

Not choosing any file will casue file to generate automaticly

Here all data fields are required, so if u don't insert all needed data then you won't be able to save them and add to database.

First name and Last name are text fields.

Email is email field so u need type there email@gmail.com or another correct email.

You can choose job from jobs list. Jobs are from json file.

Last option, you can upload your own photo or photo will be automatically generated from website: <https://randomuser.me/api>

Post/assistants- this happens when you press save button and all data are correct.

Then new assistant is added to database.

SS from postman:

POST http://127.0.0.1:5000/assistants

Send Save

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

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







KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> inputfile	tommy.jpg	
<input checked="" type="checkbox"/> first_name	Bartek	
<input checked="" type="checkbox"/> last_name	Kaczmarek	
<input checked="" type="checkbox"/> email_address	bartek@gmail.com	
<input checked="" type="checkbox"/> job_id	15	
<input checked="" type="checkbox"/> pic_name	tommy	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 39 ms Size: 57.76 KB Save Response

Pretty Raw Preview Visualize HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" href="/static/css/main.css">
7   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
8     integrity="sha384-Vkoo8x4CGs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gB1FeWPGFN9MuhOf23Q9Ifjh"
9     crossorigin="anonymous">
10
11
12 </head>
13
14 <body>
15
16 <script>
```

Result in browser:

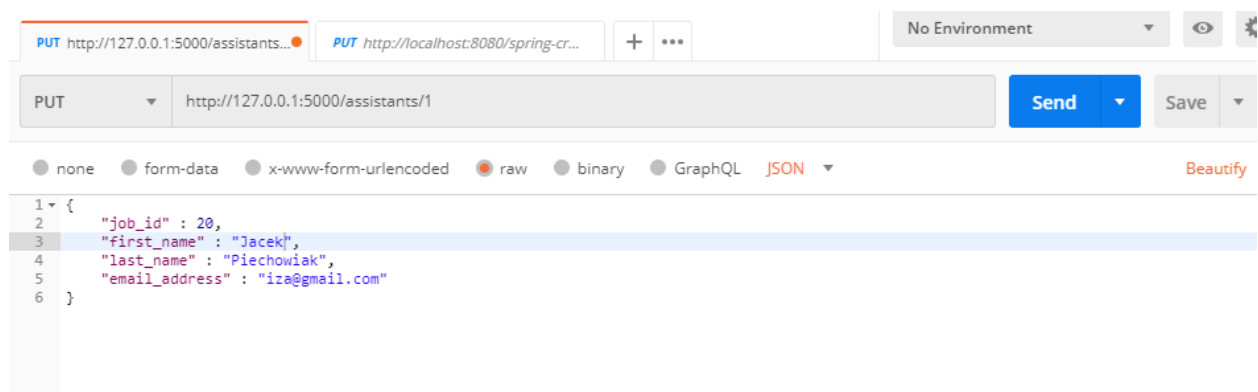
Assistant Id	Job Id	First name	Last name	Email address	Picture	Action
1	4th Grade Teacher	Iza	Piechowiak	iza@gmail.com		<button>Delete</button> <button>Update</button>
2	4th Grade Teacher	jacek	Kowalskaa	janusz@gmail.com		<button>Delete</button> <button>Update</button>
3	3D Animator	Jacek	Kaczmarek	kaczmarek.jacek10@gmail.com		<button>Delete</button> <button>Update</button>
4	2 Year Olds Preschool Teacher	Jacek	Kaczmarek	kaczmarek.jacek10@gmail.com		<button>Delete</button> <button>Update</button>
5	21 Dealer	Jacek	Kaczmarek	kaczmarek.jacek10@gmail.com		<button>Delete</button> <button>Update</button>
6	3rd Grade Teacher	Bartek	Kaczmarek	bartek@gmail.com		<button>Delete</button> <button>Update</button>

Get/assistants/update


This endpoint just shows update form of selected assistant. It is same as create assistant but current data are already inserted and you can update them if you want.

PUT /assistants/{id} – this do all updating work and changes database data.

SS from postman:



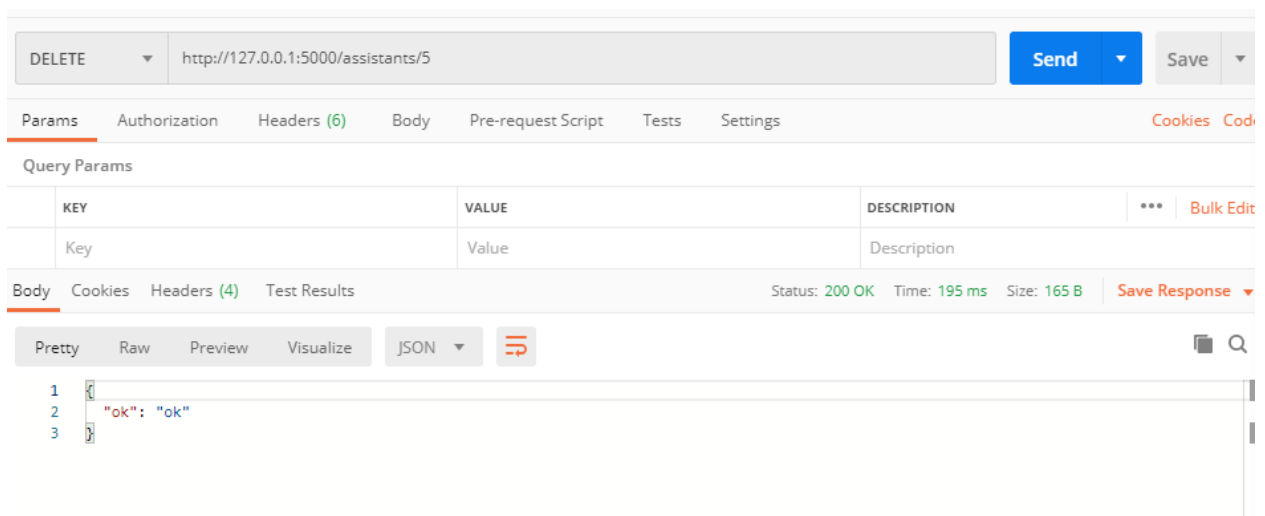
Result in web browser:

Assistant Id	Job Id	First name	Last name	Email address	Picture	Action
1	4th Grade Teacher	Jacek	Piechowiak	iza@gmail.com		<div>Delete</div> <div>Update</div>

DELETE /assistants/{id}- this endpoint removes one assistant data from database.

You can remove assistant by clicking red button 'Delete'.





SS from postman:



The screenshot shows the Postman interface for a DELETE request. The URL is `http://127.0.0.1:5000/assistants/5`. The response status is 200 OK, with a time of 195 ms and a size of 165 B. The response body is displayed in JSON format:

```
1 {
2   "ok": "ok"
3 }
```

SS from web browser:

Assistant Id	Job Id	First name	Last name	Email address	Picture	Action
1	4th Grade Teacher	Iza	Kaczmarek	iza@gmail.com		<div>Delete</div> <div>Update</div>
2	4th Grade Teacher	Janusz	Kowalski	janusz@gmail.com		<div>Delete</div> <div>Update</div>
3	4th Grade Teacher	Jacek	Kaczmarek	kaczmarek.jacek10@gmail.com		<div>Delete</div> <div>Update</div>
4	3D Artist	Jacek	Kaczmarek	kaczmarek.jacek10@gmail.com		<div>Delete</div> <div>Update</div>

As you can see there is no assistant with id number equal to 5 because it was just deleted. But remember to insert correct id number. If the assistant you want to delete does not exist, then the program won't remove anything from the database.