

README

Introduction

In this project is created Passenger's service with functionalities, like creating passenger, retrieving passengers or single passenger, updating existing passenger and deleting passengers.

Technologies

Java

Spring Boot

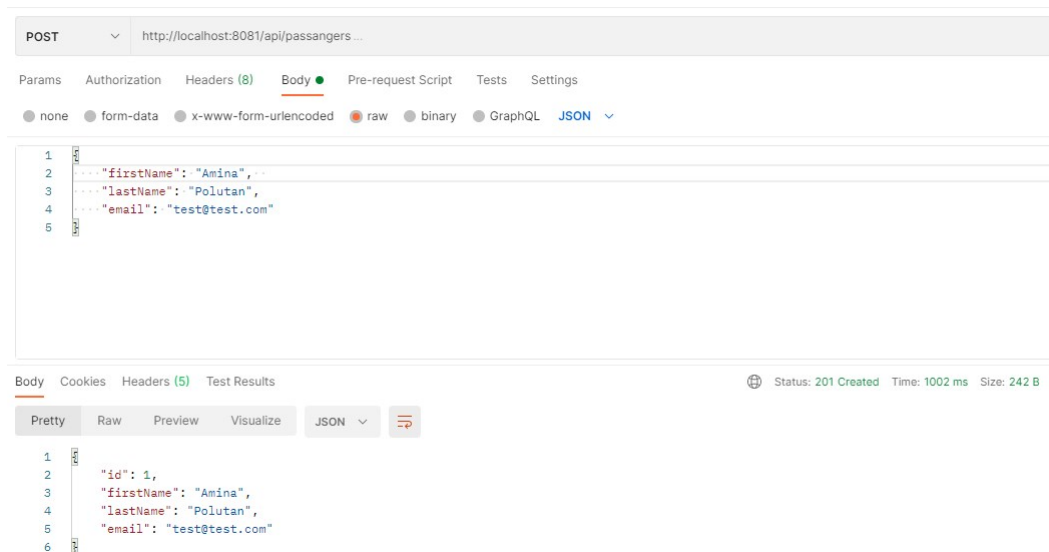
MySql

Maven

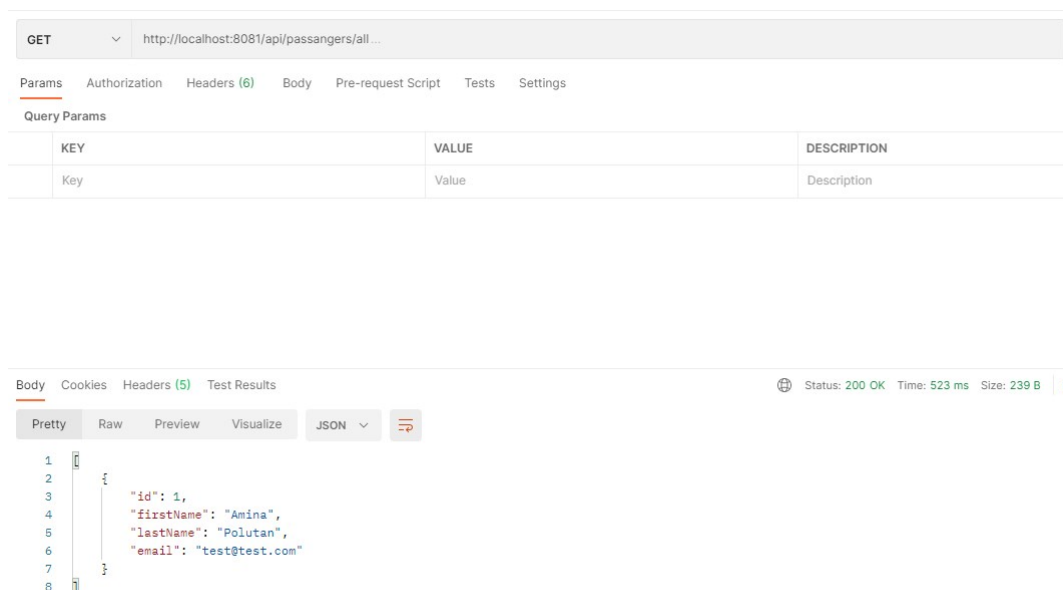
I created code in IntelliJ environment. Code is separated in different classes. Passenger class is for declaration informations about passengers, like first name, last name, email adress. This class have constructor and methods to get and set passenger properties. PassengerDTO class is for getting and setting properties from Postman application about passengers. PassengerService class moves code from controller to service component for creating passengers, retrieving passengers, updating passengers and deleting passengers. Then PassengerController is autowired with service, so we can invite constructors and methods from service. PassengerController class is responsible for processing incoming requests. Controller is mapped to the /api/passangers path.

How to run

First, you should open folder "passenger" in IntelliJ environment and run PassengerAplication, path is passenger/passenger/src/main/java, then in Postman application should be used url: <http://localhost:8081/api/passangers> for creating passenger.



For retrieving all passengers should be used url:
`http://localhost:8081/api/passangers/all`.



For retrieving passengers by id should be used url:
`http://localhost:8081/api/passangers/id?id=1`.

GET <http://localhost:8081/api/passangers?id=1>

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

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	id	1	
	Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 72 ms Size: 23

Pretty Raw Preview Visualize JSON

```

1  {
2    "id": 1,
3    "firstName": "Amina",
4    "lastName": "Polutan",
5    "email": "test@test.com"
6  }
```

For updating passengers should be used url:
<http://localhost:8081/api/passangers/1>.

PUT <http://localhost:8081/api/passangers/1>

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

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

```

1  {
2    "id": 1,
3    "firstName": "Elvin",
4    "lastName": "Polutan",
5    "email": "test@test.com"
6  }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 73 ms Size: 237 B

Pretty Raw Preview Visualize JSON

```

1  {
2    "id": 1,
3    "firstName": "Elvin",
4    "lastName": "Polutan",
5    "email": "test@test.com"
6  }
```

For deleting all passengers should be used url:
<http://localhost:8081/api/passangers> and method delete.
 And for deleting passengers by id should be used url:
<http://localhost:8081/api/passangers/1>.

Checkup in database: You will need to install MySQL Workbench and use connection with username=root and password=root or set in application.properties (path in IntelliJ: passenger/passenger/src/main/resources) yours username and password.

Query 1 x

Limit to 1000 rows

```
1 • SELECT * FROM mysql.passangers;
```

Result Grid Filter Rows: Edit: Export/Import

	id	email	first_name	last_name
▶	1	test@test.com	Amina	Polutan
*	NULL	NULL	NULL	NULL