

ДЗ 4. Система обработки заказов ресторана

Цель:

Разработать два отдельных микросервиса на основе RESTful API для системы обработки заказов в ресторане, первый из которых реализует авторизацию пользователей с различными ролями, а второй – управляет заказами и отслеживает запас блюд.

I. Микросервис авторизации пользователей

I.1. Требования к API микросервиса авторизации

I.1.1. Регистрация нового пользователя

-200, 400(email error)

- Есть

-CreatedAt updatedAt обновляются в БД автоматически при создании и изменении

Создание пользователя:



Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/users' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "Peter",
    "email": "email@gmail",
    "passwordHash": "password",
    "role": "string"
  }'
```

Request URL

http://localhost:8080/api/users

Server response

Code	Details
201	Response body

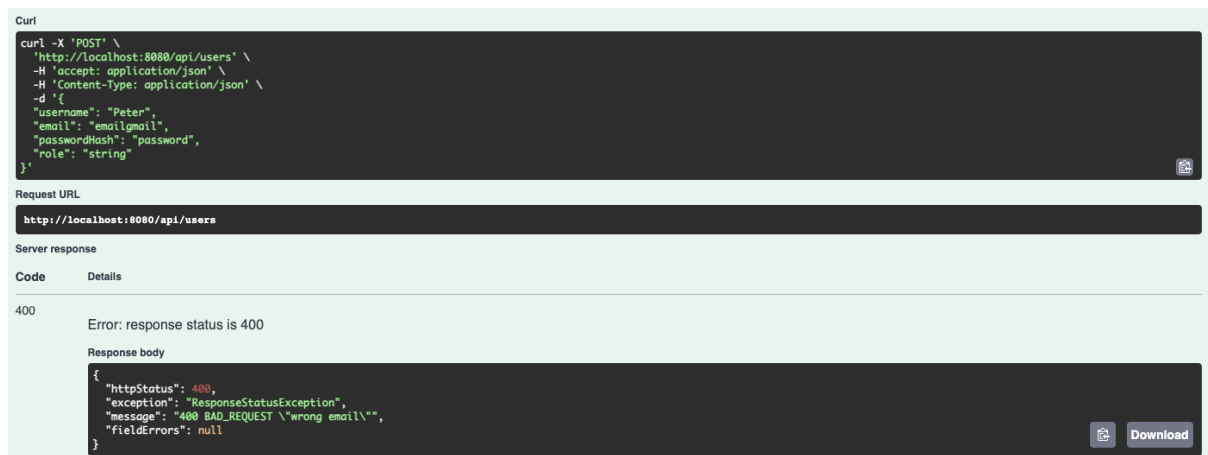
201
Undocumented

Response body

10033

Download

Неправильный email:



Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/users' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "Peter",
    "email": "email@gmail",
    "passwordHash": "password",
    "role": "string"
  }'
```

Request URL

http://localhost:8080/api/users

Server response

Code	Details
400	Error: response status is 400

400

Error: response status is 400

Response body

```
{
  "httpStatus": 400,
  "exception": "ResponseStatusException",
  "message": "400 BAD_REQUEST \"wrong email\"",
  "fieldErrors": null
}
```

Download

Удаление пользователя:

Curl

```
curl -X 'DELETE' \
'http://localhost:8080/api/users/10010' \
-H 'accept: */*'
```

Request URL

`http://localhost:8080/api/users/10010`

Server response

Code	Details
204	<p>Response headers</p> <pre>connection: keep-alive date: Sat, 27 May 2023 21:24:17 GMT keep-alive: timeout=60</pre>

Запрос всех пользователей:

Curl

```
curl -X 'GET' \
'http://localhost:8080/api/users' \
-H 'accept: application/json'
```

Request URL

`http://localhost:8080/api/users`

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "id": 10033, "username": "Peter", "email": "email@gmail", "passwordHash": "passwordHash", "role": "string", "createdAt": "2023-05-28T00:17:58.341064", "updatedAt": "2023-05-28T00:17:58.341146" }]</pre> <p>Download</p>

изменение пользователя

Curl

```
curl -X 'PUT' \
'http://localhost:8080/api/users/10033' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "username": "Frizer",
  "email": "str@ing",
  "passwordHash": "string",
  "role": "programmer"
}'
```

Request URL

`http://localhost:8080/api/users/10033`

Server response

Code	Details
200	<p>Response headers</p> <pre>connection: keep-alive content-length: 0 date: Sat, 27 May 2023 21:29:34 GMT keep-alive: timeout=60</pre>

I.1.2. Вход пользователя в систему (авторизация)

Не реализовал

I.1.3. Предоставление информации о пользователе

- 200,404(user not found)

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/users/10033' \
  -H 'accept: */*'
```

Request URL

<http://localhost:8080/api/users/10033>

Server response

Code	Details
200	Response body

Undocumented

```
{
  "id": 10033,
  "username": "Frizer",
  "email": "str@ing",
  "passwordHash": "stringHash",
  "role": "programmer",
  "createdAt": null,
  "updatedAt": "2023-05-28T00:29:34.004067"
}
```

Download

II. Микросервис обработки заказов

II.1. Требования к API микросервиса обработки заказов

II.1.1. Создание заказов

Внимание! Принимает список id OrderDishes. Это таблица в которой указаны соответствия заказов и id в меню. То есть до этого надо добавить заказ и id блюда туда. Сделать это можно с помощью свагера.

Создание блюда:

Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/dishes' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "soup",
    "description": "syp",
    "price": 10,
    "quantity": 10
  }'
```

Request URL

<http://localhost:8080/api/dishes>

Server response

Code	Details
201	Response body

10034

Download

Создание Заказа:

Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/orders' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "specialRequest": "string",
    "user": 10033,
    "dishSet": []
  }'
```

Request URL

<http://localhost:8080/api/orders>

Server response

Code	Details
201	Response body

Undocumented

10035

Download

Создание ряда id в меню - заказ:



Также заказы можно удалять и редактировать и получать по id.

II.1.2. Обработка заказов

Сделал обработку заказа

При создании заказа статус cooking после рандомного времени(6-10 сек) он изменяется на ready

II.1.3. Предоставление информации о заказе

Реализовал(выше)

Получение заказов



II.1.4. Управление блюдами

не реализовал

II.1.5. Предоставление меню

`localhost:8080/menu`

Также есть поддержка свагера, из которого можно получить API

<http://localhost:8080/v3/api-docs>

Критерии оценки ДЗ

1. Корректная реализация сервиса авторизации пользователей (2 балла):
 - 1.1. Регистрация пользователя (1 балл)
 - 1.2. Вход пользователя в систему (0.5 балла)
 - 1.3. Выдача информации о пользователе (0.5 балла)
2. Корректная реализация сервиса обработки заказов (5 баллов в случае реализации сервиса авторизации пользователей)
 - 2.1. Управление заказами (2 балла)
 - 2.2. Управление блюдами (2 балла)
 - 2.3. Предоставление информации о меню (1 балл)
3. Реализация коллекции Postman (или Swagger), которая должна демонстрировать функциональность реализованн(ого/ых) микросервис(а/ов), охватывая все API (1 балл в случае реализации сервис(а/ов)).
4. Качество кода и документация (2 балла):
 - 4.1. Хорошо организованный, модульный и поддерживаемый код.
 - 4.2. Качественная документация, включая краткое описание архитектуры системы и спецификацию API.