

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 2

Выполнил: Зайцев Кирилл
Дмитриевич

Группа К33402

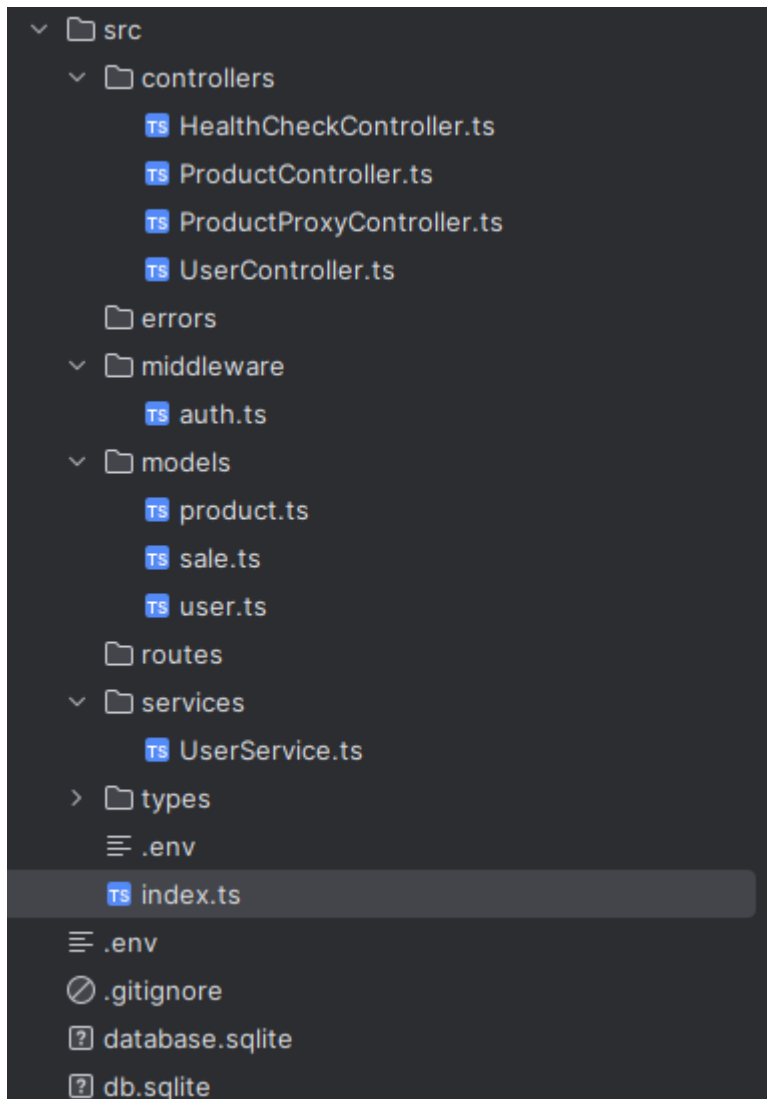
Проверил:
Добряков Д. И.

Санкт-Петербург

Задача:

По выбранному варианту необходимо будет реализовать RESTful API средствами express + typescript (используя ранее написанный boilerplate).

1. По выбранному варианту было составлено минимальное приложение с простым функционалом пользователей и товаров.



2. Создадим модели user.ts:

```
import { Table, Column, Model, Unique, PrimaryKey, AutoIncrement, Default } from
'sequelize-typescript';

@Table
export class User extends Model {
  @Unique
  @PrimaryKey
  @AutoIncrement
  @Column
  declare id: number;

  @Column
  declare firstName: string;
```

```

    @Column
    declare lastName: string;

    @Column
    declare email: string;

    @Column
    declare password: string;

    @Default(false)
    @Column
    declare isAdmin: boolean;
}

```

sale.ts:

```

import {
  Table,
  Column,
  Model,
  PrimaryKey,
  AutoIncrement,
  ForeignKey,
  BelongsTo
} from 'sequelize-typescript'
import { User } from './user.js'
import { Product } from './product.js'

@Table
export class Sale extends Model {
  @PrimaryKey
  @AutoIncrement
  @Column
  declare id: number

  @ForeignKey(() => User)
  @Column
  declare userId: number

  @ForeignKey(() => Product)
  @Column
  declare productId: number

  @Column
  declare quantity: number

  @Column
  declare totalPrice: number

  @BelongsTo(() => User)
  user: User

  @BelongsTo(() => Product)
  product: Product
}

```

product.ts:

```
import {
  Table,
  Column,
  Model,
  PrimaryKey,
  AutoIncrement,
  Default,
} from 'sequelize-typescript'

@Table
export class Product extends Model {
  @PrimaryKey
  @AutoIncrement
  @Column
  declare id: number

  @Column
  declare name: string

  @Column
  declare category: string

  @Column
  declare price: number

  @Default(0)
  @Column
  declare stock: number

  @Default(false)
  @Column
  declare discount: number
}
```

5. Создадим сервис для работы с юзером и veddleware для ограничения доступа к функционалу:

```
1  import { User } from '../models/user.js'
2
3  export class UserService { no usages
4    public async findByEmail(email: string): Promise<User | null> { no usages
5      return await User.findOne( options: { where: { email } })
6    }
7
8    public async createUser(data: Partial<User>): Promise<User> { no usages
9      return await User.create(data)
10   }
11 }
12
```

Auth.ts:

```
import { NextFunction, Request, Response } from 'express'
import jwt from 'jsonwebtoken'
```

```
export const authMiddleware = (req: Request, res: Response, next: NextFunction):
void => {
  const token = req.headers.authorization?.split(' ')[1]
  if (!token) {
    res.status(401).json({ error: 'Unauthorized' })
    return
  }

  try {
    req.user = jwt.verify(token, 'SECRET_KEY') as { id: number, isAdmin:
boolean }
    next()
  } catch (error) {
    res.status(401).json({ error: 'Invalid token' })
  }
}
```

Примеры выполнения запросов в postman:

register

lab docs / Users / Register

POST http://localhost:3000/register

Params Authorization Headers (8) Body Scripts Settings

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

```
1 {
2   "id": 2,
3   "firstName": "Vova",
4   "lastName": "Kruglov",
5   "email": "vovaloks@ya.ru",
6   "password": "Vova2002sep",
7   "isAdmin": false
8 }
```

Body Cookies Headers (7) Test Results

Status: 201 Created Time: 101 ms Size: 482 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 6,
3   "firstName": "Vova",
4   "lastName": "Kruglov",
5   "email": "vovaloks@ya.ru",
6   "password": "$2b$10$yMQ4HKHJfxcawoxlH.C07JA.8sSVc6y9/V3bF.xBgKUZm8zb7im",
7   "isAdmin": false,
8   "updatedAt": "2024-06-09T08:23:51.251Z",
9   "createdAt": "2024-06-09T08:23:51.251Z"
10 }
```

Login:

lab docs / Users / Login

POST http://localhost:3000/login

Params Authorization Headers (9) Body Scripts Settings

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

```
1 {
2   "email": "kuiushka@ya.ru",
3   "password": "Akashi307477"
4 }
```

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 70 ms Size: 405 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaXNzI6IiE6dHJlZSwiaWF0IjoxNzE3OTI1NDczLCJleHAiOiE3MTc6MTUwNzN9.3tokZXsx1RU54tf9PgDJyZoe03tNvvvWmZchJFY089U"
3 }
```

Post product:

lab docs / Products / Add product

POST

{{server_url}}products

Send

ParamsAuthorizationHeaders (9)BodyScriptsSettings

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

BodyCookiesHeaders (7)Test Results

Status: 201 CreatedTime: 59 msSize: 427 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 9,
3   "name": "Флешка 16ГБ",
4   "category": "Периферия",
5   "price": 1850,
6   "stock": 100,
7   "discount": 50,
8   "updatedAt": "2024-06-09T08:25:51.816Z",
9   "createdAt": "2024-06-09T08:25:51.816Z"
10 }
```

Get product by id:

lab docs / Products / Get product stats by id

PUT

{{server_url}}products/8

Send

ParamsAuthorizationHeaders (8)BodyScriptsSettings

Auth Type

Bearer Token

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables.

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 14 msSize: 413 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "id": 8,
3   "name": "Не мышка",
4   "category": "Sample Category",
5   "price": 100,
6   "stock": 50,
7   "discount": 10,
8   "updatedAt": "2024-06-04T20:32:57.845Z",
9   "createdAt": "2024-06-04T20:32:57.845Z"
10 }
```


Search:

lab docs / Products / search product

OET

{{server_url}}products/search?query=Product

Send

Params Authorization Headers (6) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> query	Product			
<input type="checkbox"/> Key	Value	Description		

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 7 ms Size: 1.28 KB Save as example

Pretty Raw Preview Visualize JSON

```
39      "createdAt": "2024-06-04T20:01:47.167Z",
40      "updatedAt": "2024-06-04T20:01:47.167Z"
41    },
42    {
43      "id": 5,
44      "name": "Not a Sample Product",
45      "category": "Sample Category",
46      "price": 100,
47      "stock": 50,
48      "discount": 10,
49      "createdAt": "2024-06-04T20:01:47.021Z",
50      "updatedAt": "2024-06-04T20:01:47.021Z"
51    },
52    {
53      "id": 6,
54      "name": "Simp Product",
55      "category": "Sample Category",
56      "price": 100,
57      "stock": 50,
58      "discount": 10,
59      "createdAt": "2024-06-04T20:01:53.435Z",
60      "updatedAt": "2024-06-04T20:01:53.435Z"
61    }
62  ]
```

Вывод:

В этой лабораторной работе был разработан сервис для интернет-магазина, используя шаблон, который был создан в ходе предыдущей лабораторной работы.