САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 2

Выполнил: Рашевский Вячеслав Васильевич

Группа K33402

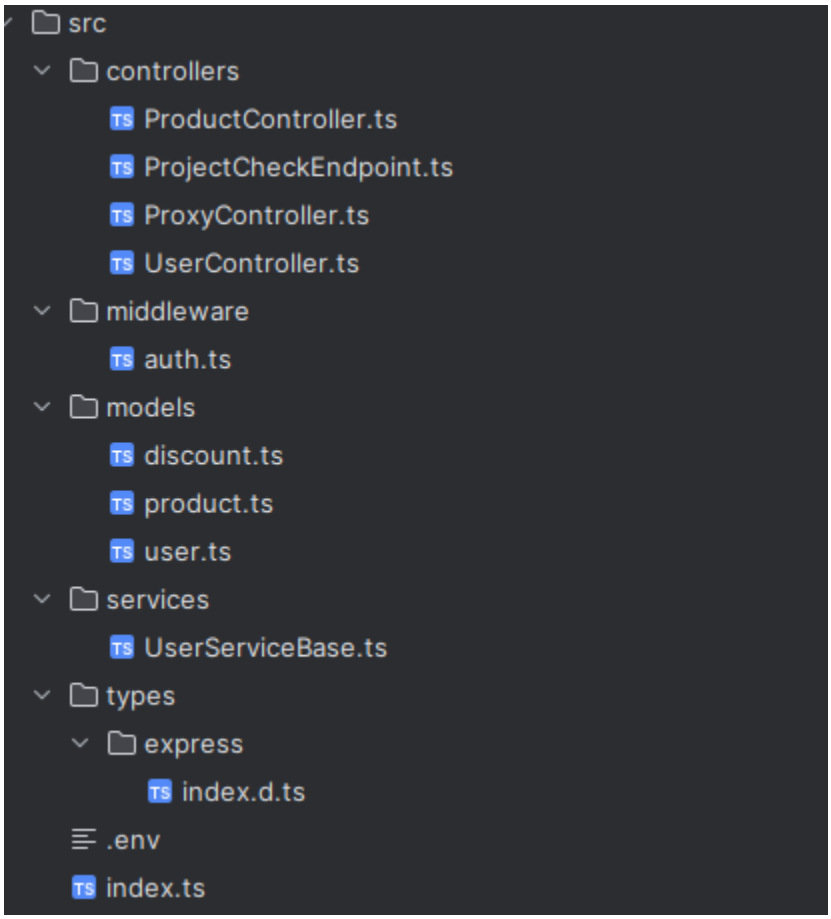
Проверил:
Добряков Д. И.

Санкт-Петербург

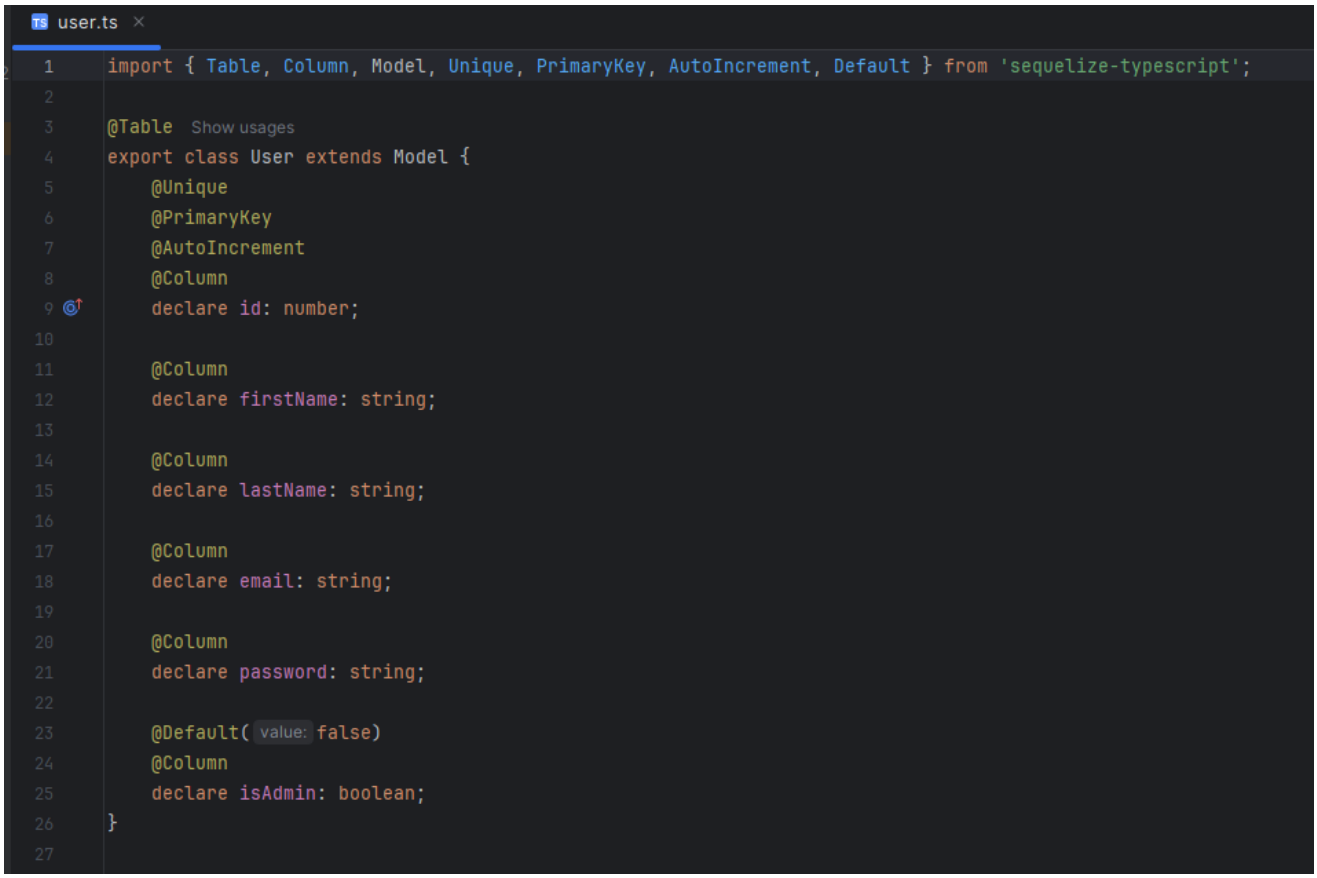
Задача:

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:

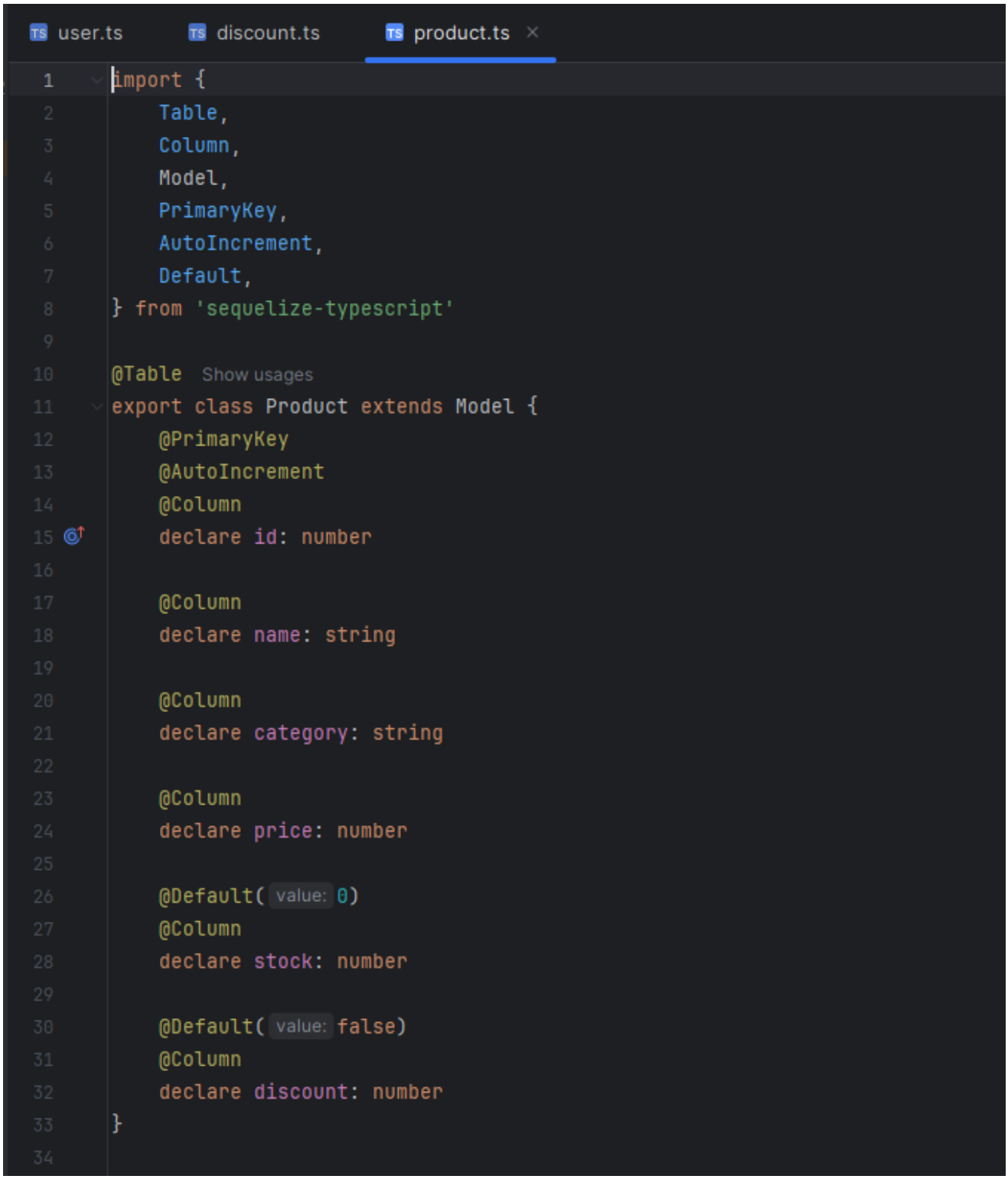


discount.ts:

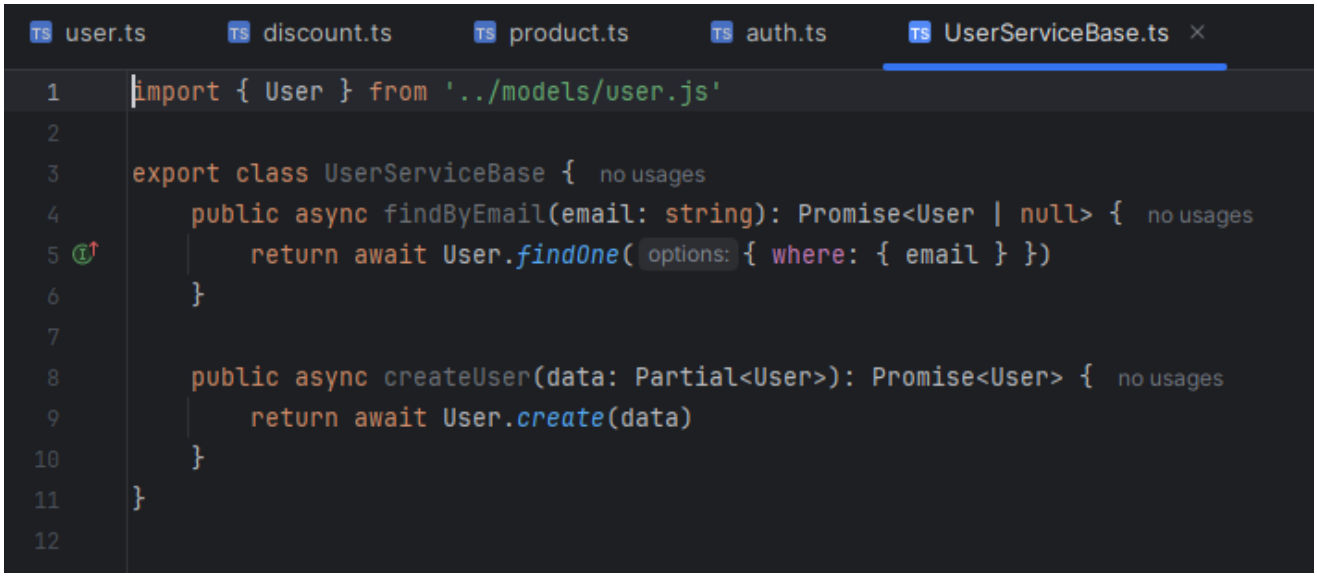
user.ts discount.ts

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

@Table  Show usages
export class Discount extends Model {
    @PrimaryKey
    @AutoIncrement
    @Column
    declare id: number

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

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

    @Column
    declare quantity: number

    @Column
    declare totalPrice: number

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

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

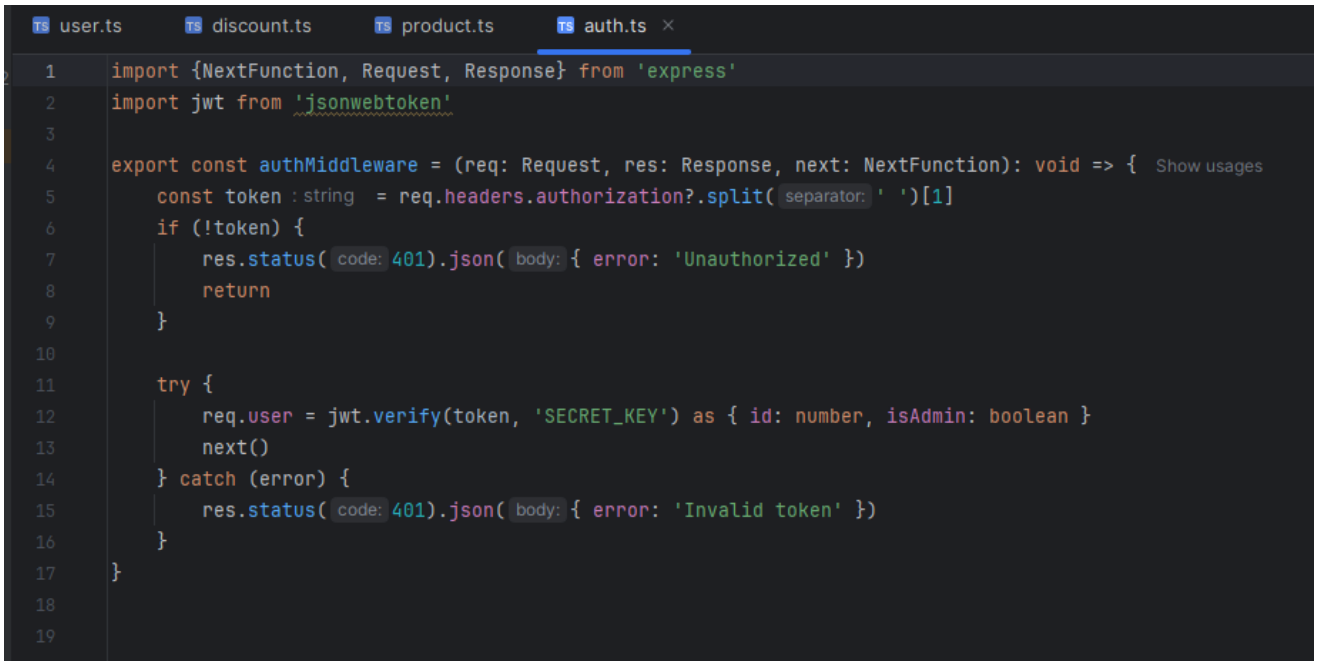
product.ts:


user.ts
discount.ts
product.ts
import {
Table,
Column,
Model,
PrimaryKey,
AutoIncrement,
Default,
} from 'sequelize-typescript'
@Table Show usages
export class Product extends Model {
@PrimaryKey
@AutoIncrement
@Column
declare id: number
@Column
declare name: string
@Column
declare category: string
@Column
declare price: number
@Default(value: 0)
@Column
declare stock: number
@Default(value: false)
@Column
declare discount: number
}

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

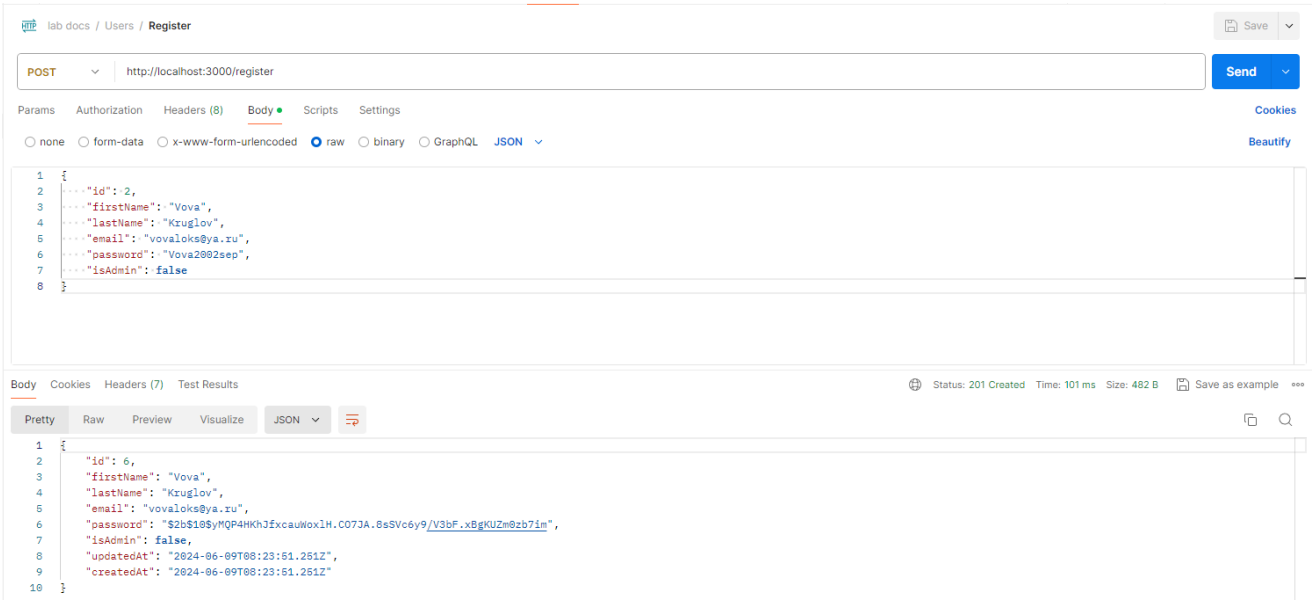


Auth.ts:

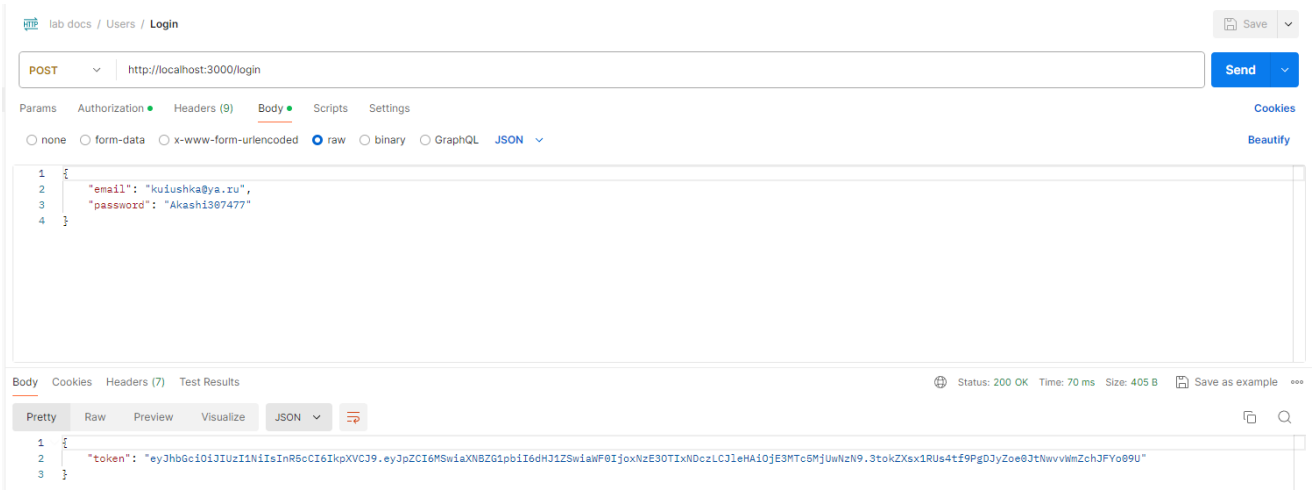


Примеры выполнения запросов в postman:

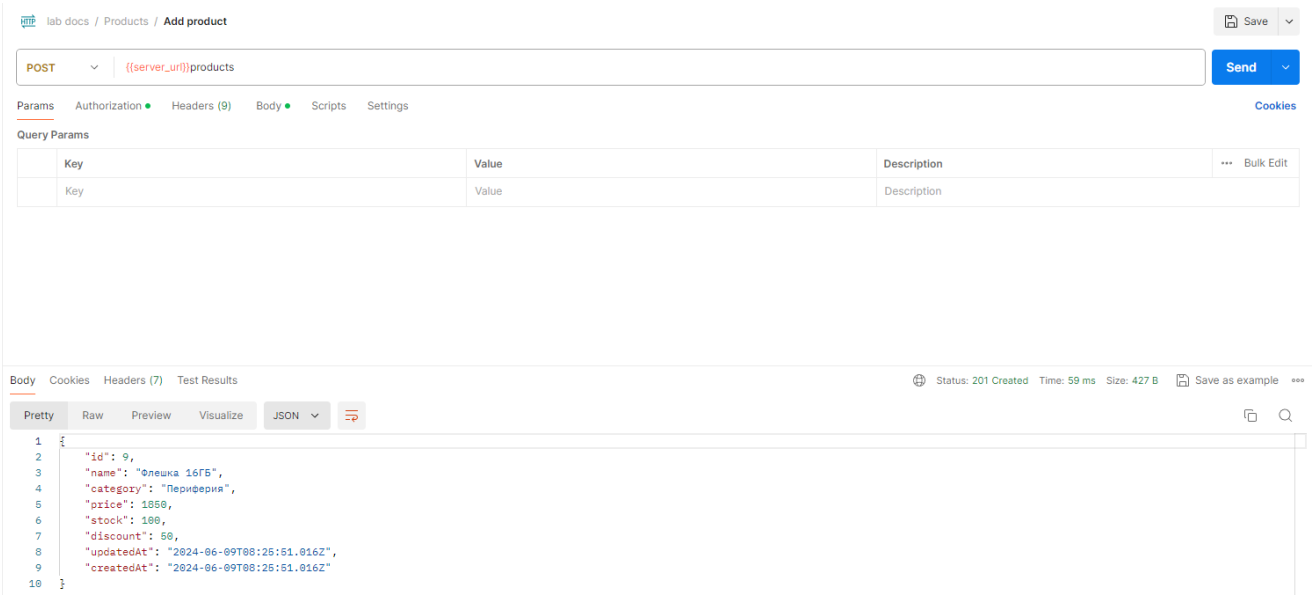
registration



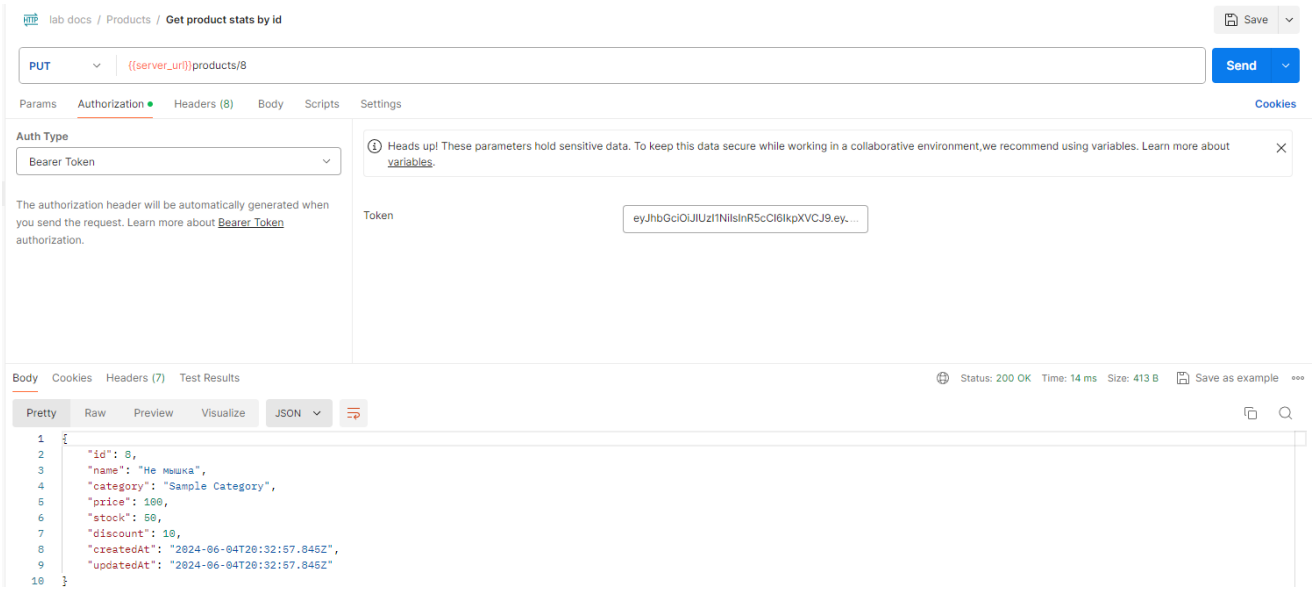
Login:



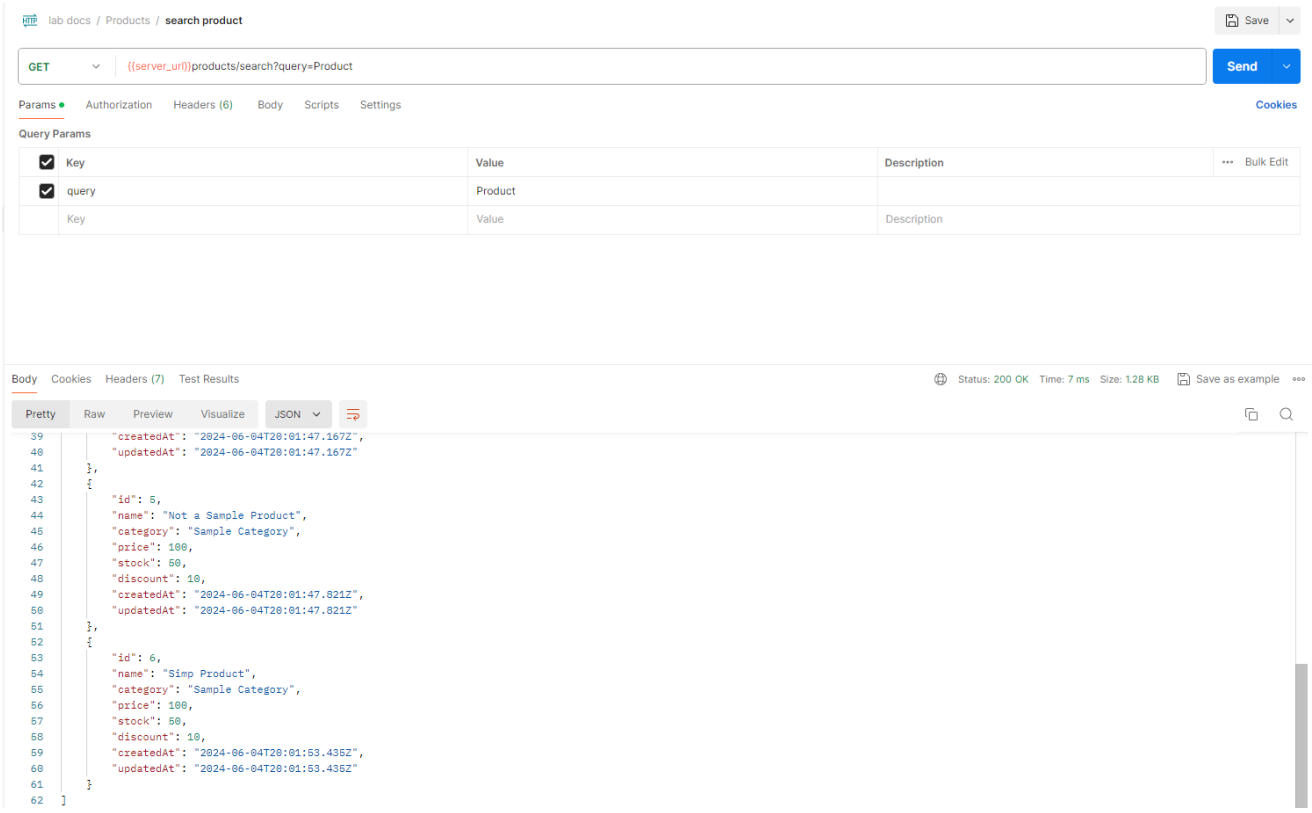
Post products:



Get product by id:



Search:


lab docs / Products / search product
Save
GET
{{server_url}}products/search?query=Product
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
query
Product
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
"createdAt": "2024-06-04T20:01:47.167Z",
"updatedAt": "2024-06-04T20:01:47.167Z"
},
{
"id": 5,
"name": "Not a Sample Product",
"category": "Sample Category",
"price": 100,
"stock": 50,
"discount": 10,
"createdAt": "2024-06-04T20:01:47.821Z",
"updatedAt": "2024-06-04T20:01:47.821Z"
},
{
"id": 6,
"name": "Simp Product",
"category": "Sample Category",
"price": 100,
"stock": 50,
"discount": 10,
"createdAt": "2024-06-04T20:01:53.435Z",
"updatedAt": "2024-06-04T20:01:53.435Z"
}
]

Вывод:
В рамках этой лабораторной работы был создан сервис для интернет-магазина. В качестве основы для разработки использовался шаблон, созданный в ходе предыдущей лабораторной работы. Этот шаблон включал в себя такие технологии, как TypeScript, Sequelize и Express. Благодаря уже готовому шаблону удалось значительно сократить время на начальную настройку проекта и сосредоточиться на разработке функционала, специфичного для интернет-магазина. Сервис включает в себя функциональные возможности для управления товарами, обработки заказов, а также взаимодействия с базой данных, обеспечивая надежную и эффективную работу интернет-магазина.