

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №2

Выполнил:
Жаров Александр
К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

Задача

В рамках данной лабораторной работы Вам предложено выбрать один из нескольких вариантов. Выбранный вариант останется единственным на весь курс и будет использоваться в последующих лабораторных работах.

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

Ход работы

1. Для разработки был выбран вариант с интернет-магазином одежды и для него были определены необходимые модели и методы для api.

User	Product	Product in cart	Cart
id email password full name cart_id refresh token	id name description price brand categorys_id[] count discount_id	id cart_id name description price brand categorys_id[] discount_id count in cart	id user_id products in cart[] total price
Category	Discount		
id title products_id[]	products_id[] value description		

User Login Registration Logout me	Category get all	Product get products (byName, byCategory, byPrice, byBrand) get product by id
Cart get cart add product to cart (create product in cart) update cart delete product in cart		Discount get product by discount.

2. Для начала мы взяли boilerplate на nestjs, к которому подключили prisma. Этот инструмент позволил нам использовать встроенную админку, что упростило взаимодействие с бд.
3. Далее мы создали необходимые модели и добавили все виды связей

```

You, 3 дня назад | 1 author (You)
model User {
  id          Int          @id @default(autoincrement())
  email       String       @unique
  password    String
  fullName    String?
  cart        Cart?
  cartId      Int?
  refreshToken String?
}

You, 3 дня назад | 1 author (You)
model Cart {
  id          Int          @id @default(autoincrement())
  User        User         @relation(fields: [userId], references: [id])
  totalPrice  Int
  productsInCart ProductInCart[]
  userId      Int          @unique
}

You, 3 дня назад | 1 author (You)
model Product {
  id          Int          @id @default(autoincrement())
  title       String
  description  String
  price       Int
  categories  Category[]
  Discount    Discount?   @relation(fields: [discountId], references: [id])
  discountId  Int?
}

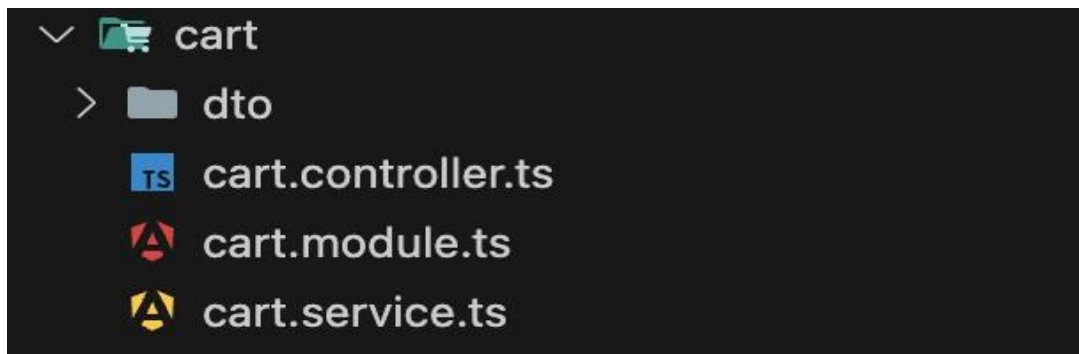
You, 3 дня назад | 1 author (You)
model ProductInCart {
  id          Int          @id @default(autoincrement())
  productId   Int
  Cart        Cart?       @relation(fields: [cartId], references: [id])
  cartId      Int?
  count       Int
}

You, 3 дня назад | 1 author (You)
model Category {
  id          Int          @id @default(autoincrement())
  title       String
  Products    Product[]
}
You, 3 дня назад * feature: add bd models

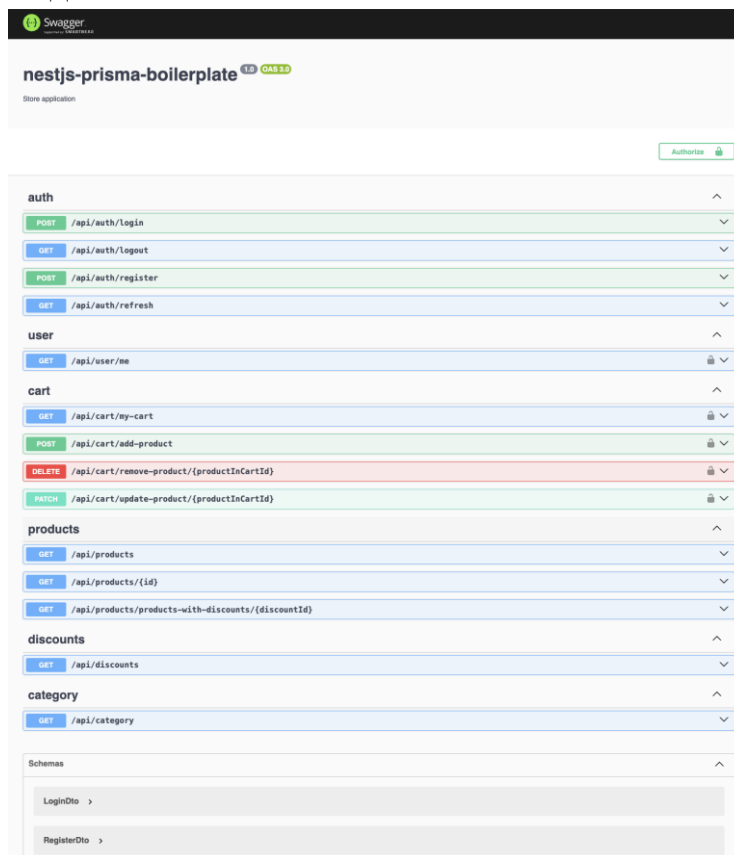
You, 3 дня назад | 1 author (You)
model Discount {
  id          Int          @id @default(autoincrement())
  value       Int
  description  String
  Products    Product[]
}

```

4. Для каждой модели был создан отдельный модуль, который состоит из:
 - a. controller - для обработки API запросов
 - b. service - для реализации основной логики
 - c. module - для объединения всех составных частей модуля.
 - d. dto - data to transfer models



5. В результате был получен сервис с готовым api и документацией для онлайн магазина одежды с регистрацией и всеми необходимыми моделями.



Вывод

В ходе работы я освоил фреймворк nestjs и научился подключать к нему такие инструменты, как swagger и prisma.