**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

# **Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа 3

Выполнили:

Никитин Павел

Группа

К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

## Задача

Необходимо реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций вашего приложения.

## Ход работы

Для этой задачи, было решено выделить сервис корзины из основного приложения. Поэтому была полностью переделана структура. Репозиторий проекта превратился в моно репозиторий содержащий сразу несколько сервисов для удобства в качестве утилиты был использован nx.

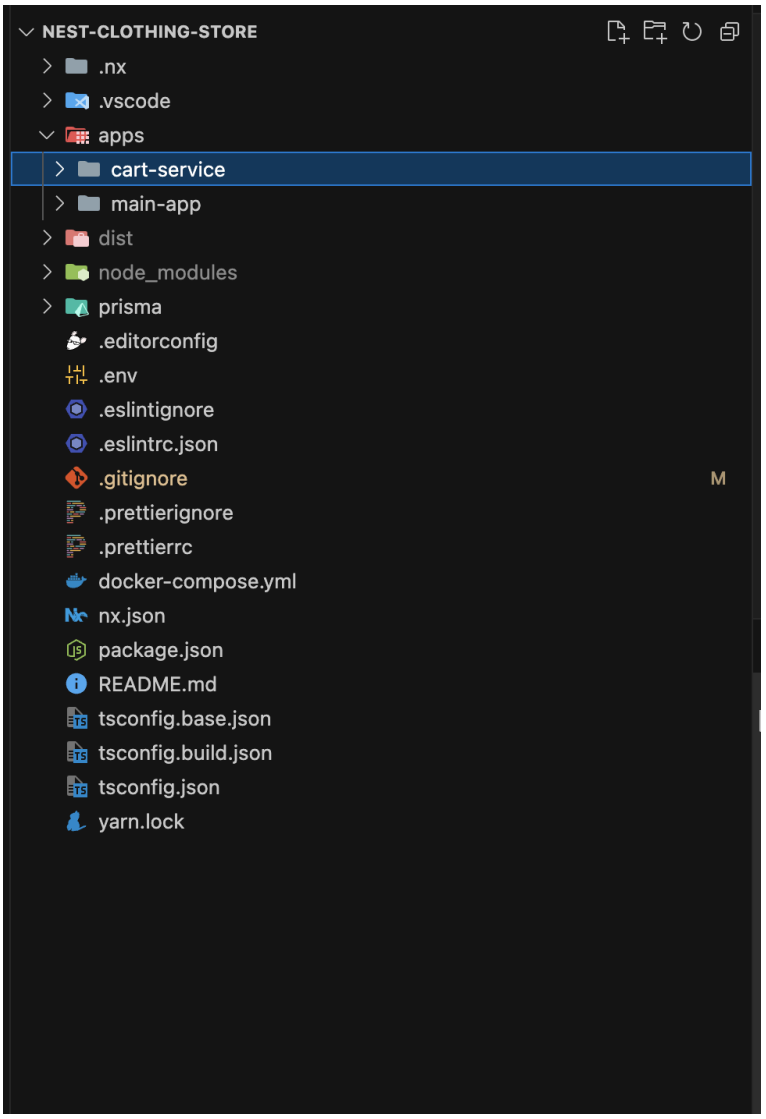
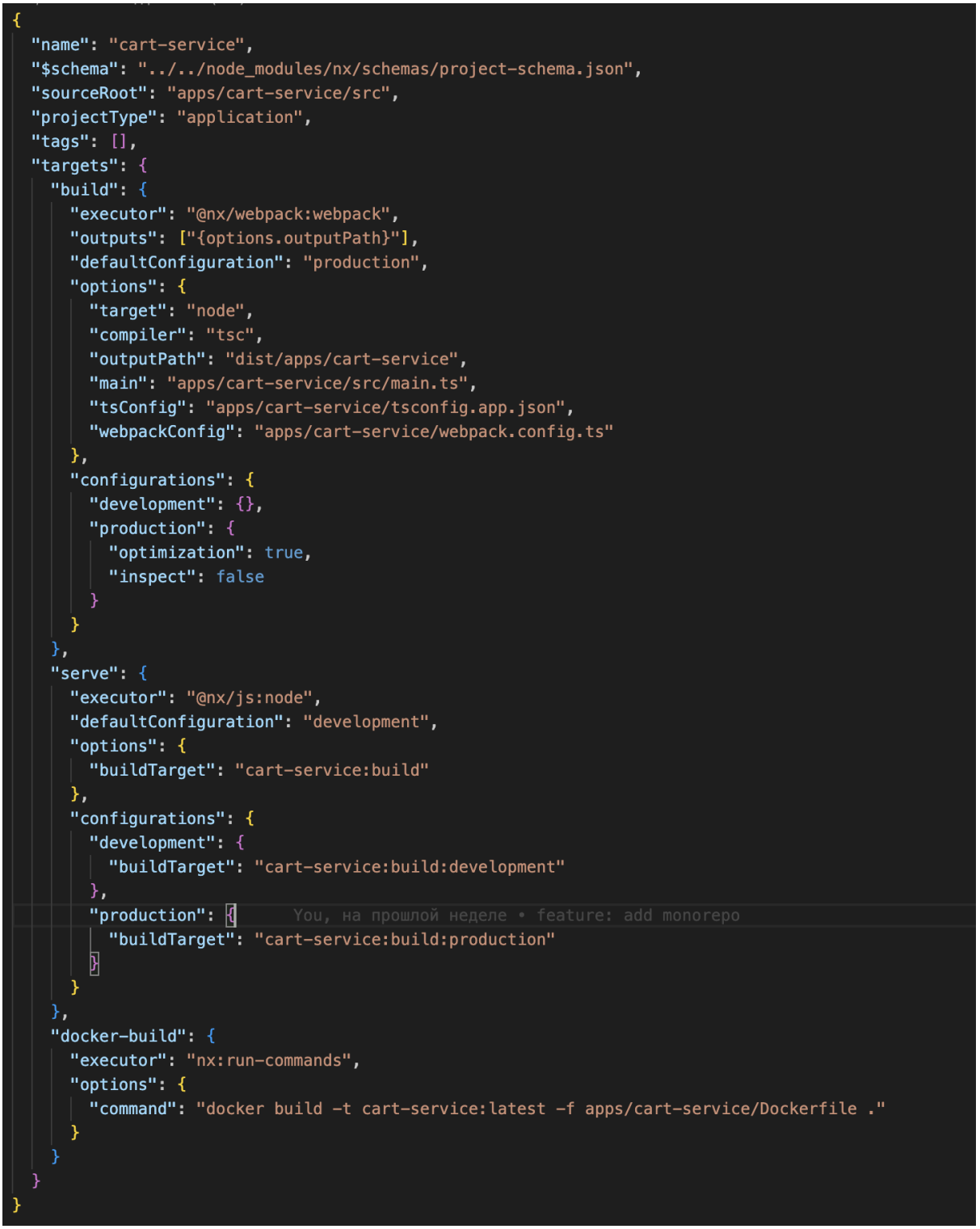



Рисунок 1 - новая структура проекта

Теперь каждый микросервис имеет свой project.json, который описывает как правильно собирать проект



```
{
  "name": "cart-service",
  "$schema": "../../node_modules/nx/schemas/project-schema.json",
  "sourceRoot": "apps/cart-service/src",
  "projectType": "application",
  "tags": [],
  "targets": {
    "build": {
      "executor": "@nx/webpack:webpack",
      "outputs": ["{options.outputPath}"],
      "defaultConfiguration": "production",
      "options": {
        "target": "node",
        "compiler": "tsc",
        "outputPath": "dist/apps/cart-service",
        "main": "apps/cart-service/src/main.ts",
        "tsConfig": "apps/cart-service/tsconfig.app.json",
        "webpackConfig": "apps/cart-service/webpack.config.ts"
      },
      "configurations": {
        "development": {},
        "production": {
          "optimization": true,
          "inspect": false
        }
      }
    },
    "serve": {
      "executor": "@nx/js:node",
      "defaultConfiguration": "development",
      "options": {
        "buildTarget": "cart-service:build"
      },
      "configurations": {
        "development": {
          "buildTarget": "cart-service:build:development"
        },
        "production": {      You, на прошлой неделе • feature: add monorepo
          "buildTarget": "cart-service:build:production"
        }
      }
    },
    "docker-build": {
      "executor": "nx:run-commands",
      "options": {
        "command": "docker build -t cart-service:latest -f apps/cart-service/Dockerfile ."
      }
    }
  }
}
```

Рисунок 2 - project.json мокросервиса

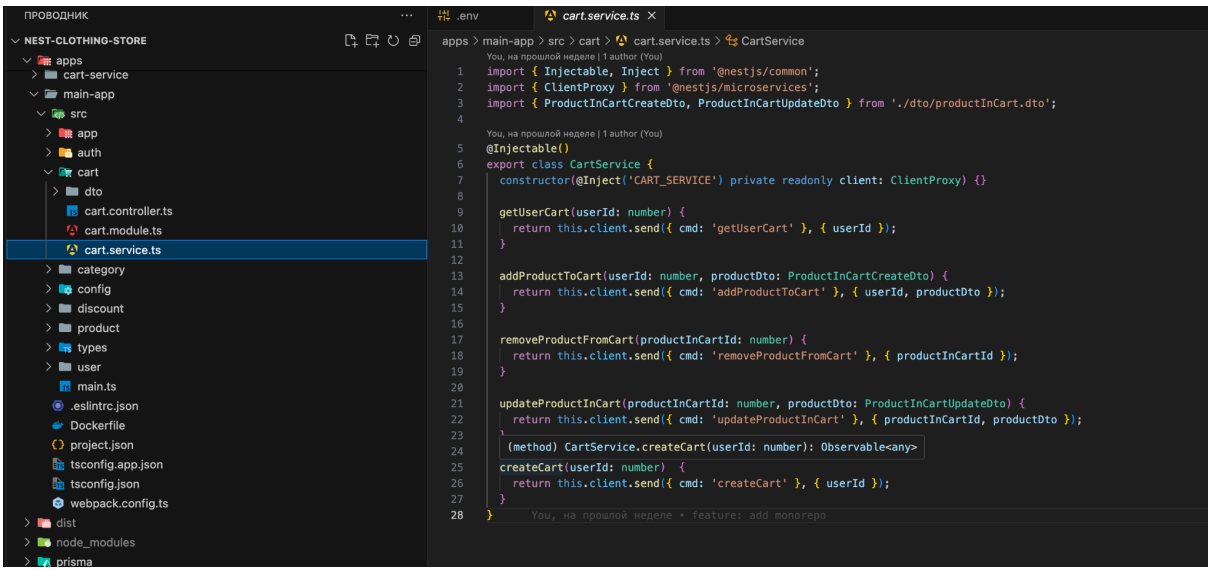



Рисунок 3 - реализация cartService в главном приложении

Если раньше мы в сервисе обращались к базе данных то теперь просим получить эту информацию от другого микросервиса.

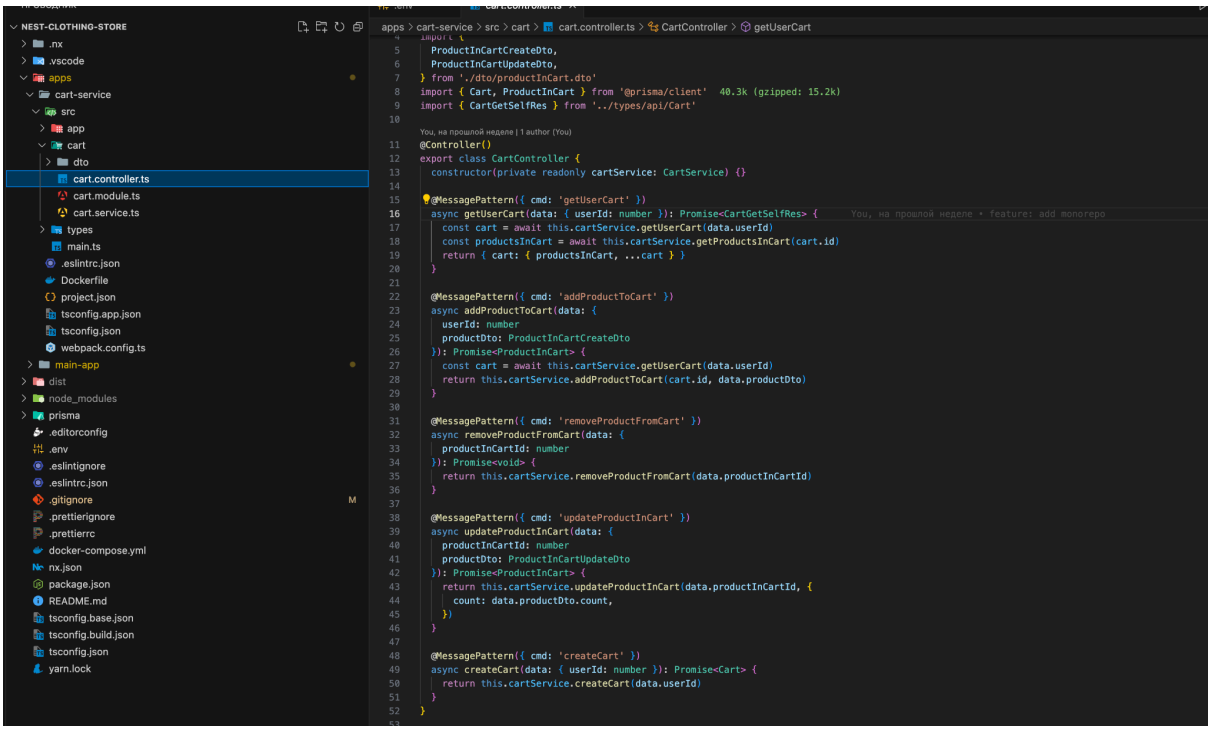



Рисунок 4 - контроллер в микросервисе корзины


Рисунок 5 - скрипты для запуска проекта

**Ссылка на результат -** https://github.com/pavel-nikitin-2022/nest-clothing-store/tree/pavelnikitin/feature/microservices

**Вывод**

В ходе этой работы я научился писать микросервисы на nestjs, хоть в идеале это должны быть разные репозитории с разными базами данных, текущая архитектура намного лучше предыдущей и позволяет легко вносить изменения в проект.