Лекция

ПСКП, ПОИТ-3

**Node.js: Nest**

1. **Nest:** это фреймворк для создания серверных веб-приложений на языке JavaScript. Он значительно расширяет функциональность Express.js, добавляет декларативности и помогает разработчику строить приложение в соответствии с лучшими архитектурными практиками.
2. **Особенности**:

- декларативное программирование при помощи декораторов;

- встроенный DI-контейнер;

- мощные специализированные middleware;

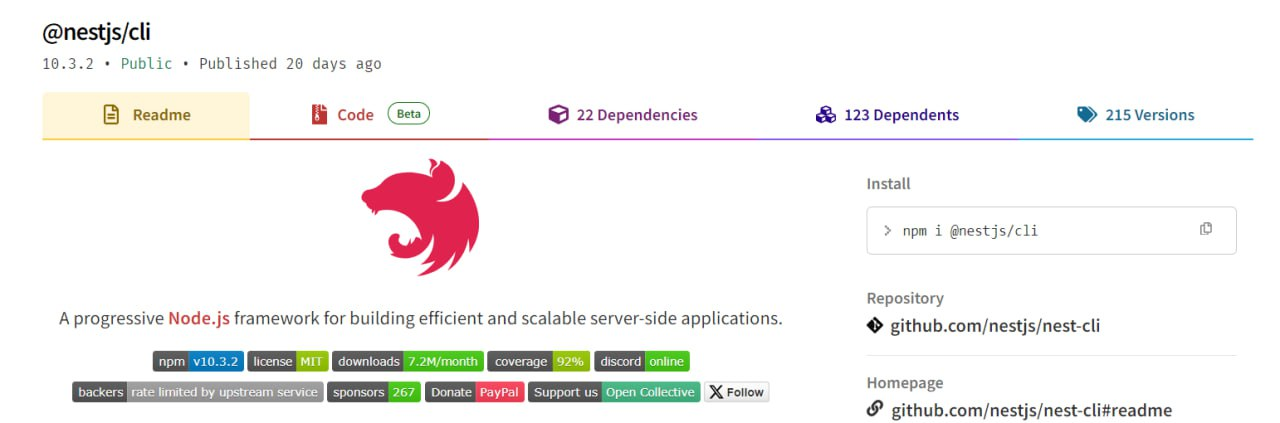
- механизм обработки ошибок «из коробки»;

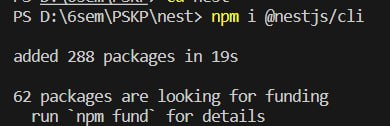
- поддержка любых протоколов, помимо HTTP.

1. **Nest:** основные характеристики:

* Nest.js написан на TypeScript и полностью поддерживает его, это обеспечивает:
* Проверку на тип данных: как простых, так и сложных.
* Декораторы – функционал декларативного программирования, позволяющий расширить любые методы, какие только пожелаем.
* Классические импорты TypeScript. Побочным эффектом послужит ускорение разработки, а также возможность писать код большой командой, не мешая друг другу.
* Четкое разделение функциональной нагрузки по элементам (controllers, middleware, exeption filters, modules)

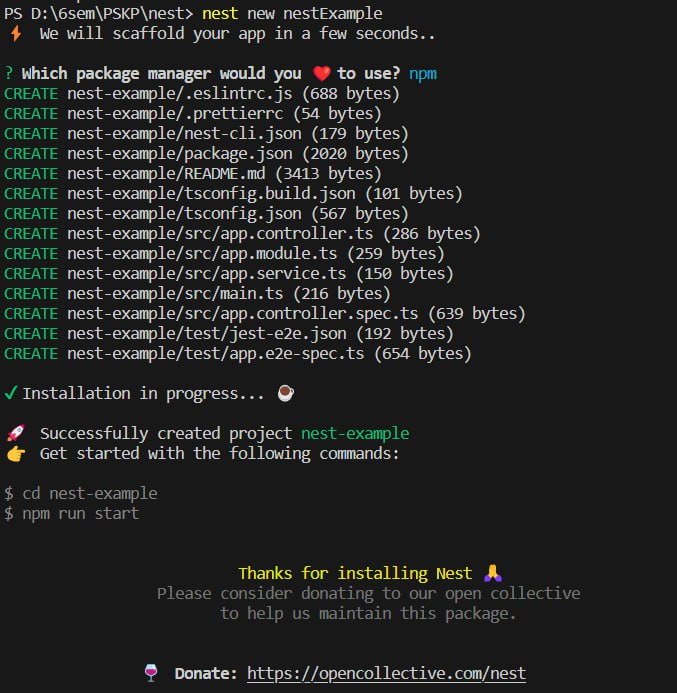
1. **Nest:** npm install -g @nestjs/cli



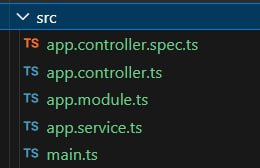


1. **Nest:** создание и настройка проекта Nest

Инициализация проекта:



Выполнение команды nest new приводит к генерации следующих файлов:



**app.controller.ts** — базовый (basic) контроллер с одним роутом (обработчиком маршрута или пути);

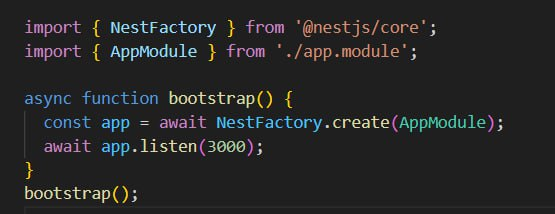
**app.controller.spec.ts** — юнит-тесты для контроллера;

**app.module.ts** — корневой (root) модуль приложения;

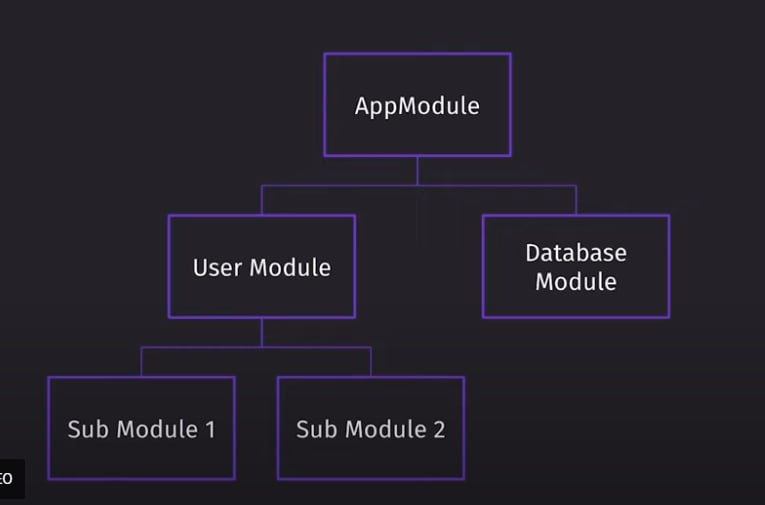
**app.service.ts** — базовый сервис с одним методом;

**main.ts** — точка входа приложения, в котором используется класс *NestFactory* для создания экземпляра приложения *Nest*.

1. **Nest:** main.ts содержит асинхронную функцию, которая инициализирует приложения:

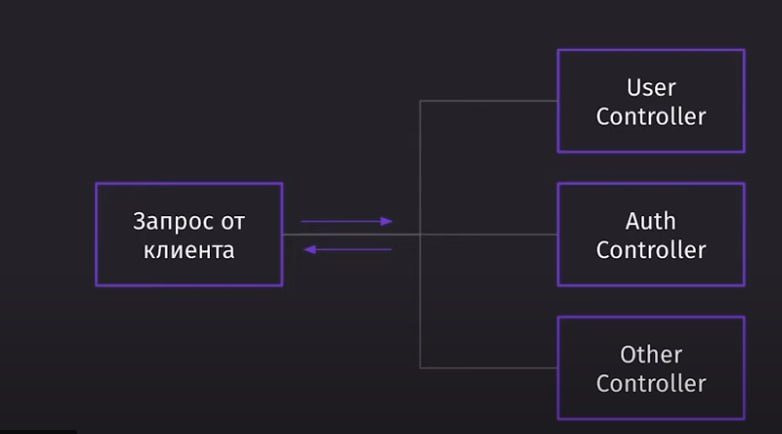


1. **Nest**: Modules – некоторый строительный блок, который инкапсулирует у себя внутри всю информацию из определенной доменной области, по сути все, что мы можем отделить это модуль.

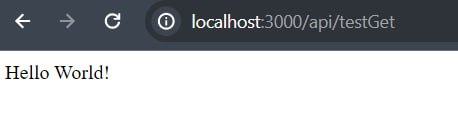




1. **Nest**: Controllers это точки, куда приходит внешний запрос в наше приложение







1. **Nest**: middleware - прослойка между контроллером и клиентом

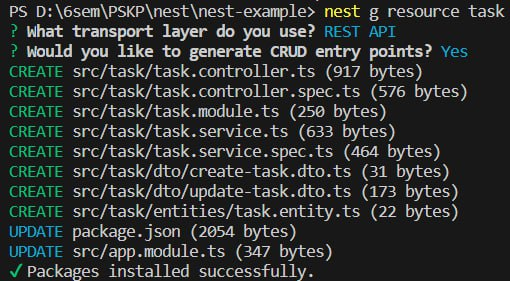


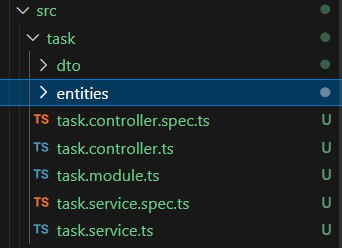


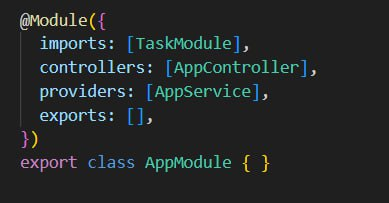
1. **Nest**: Service(provider) – выделенные сервисы, репозитории, которые инкапсулируют в себя функциональность.

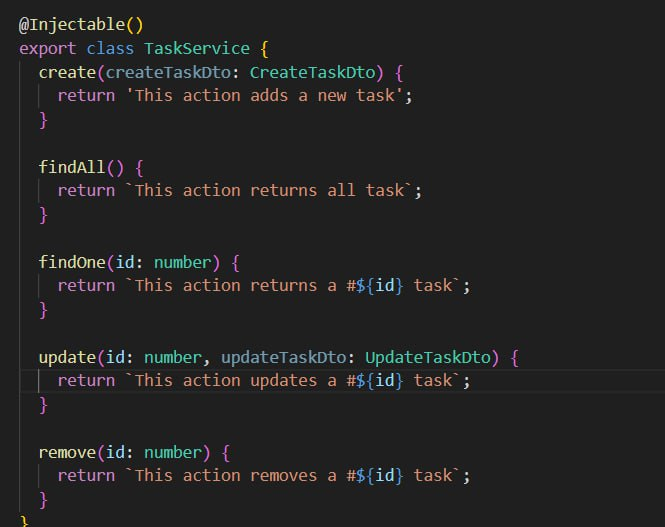


1. **Nest:** создание новой сущности

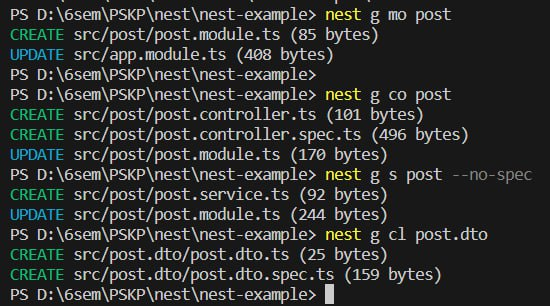




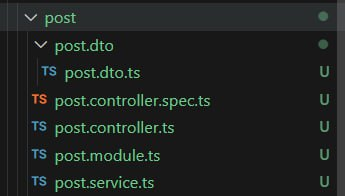




1. **Nest:** nest g mo, nest g co, nest g s

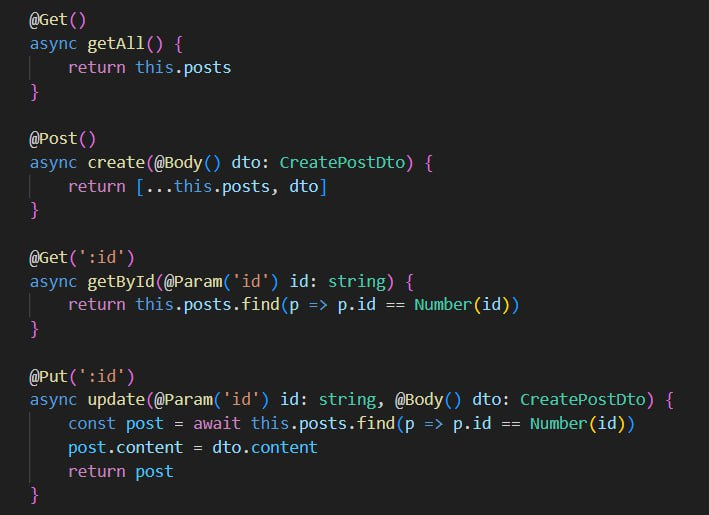


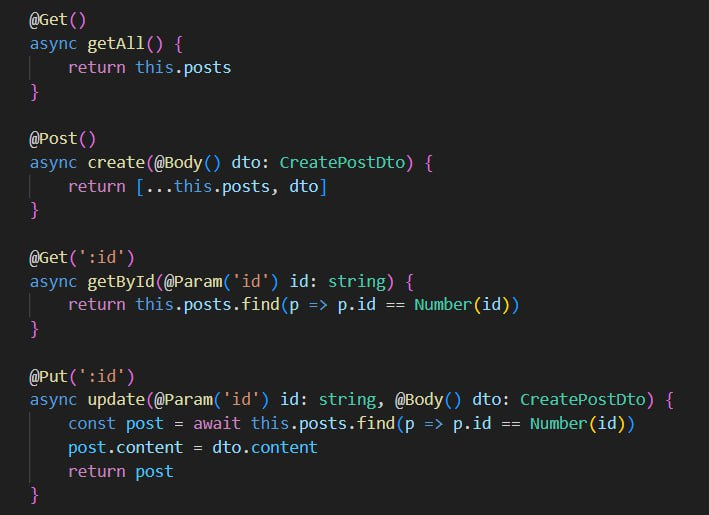
1. **Nest:** post entity



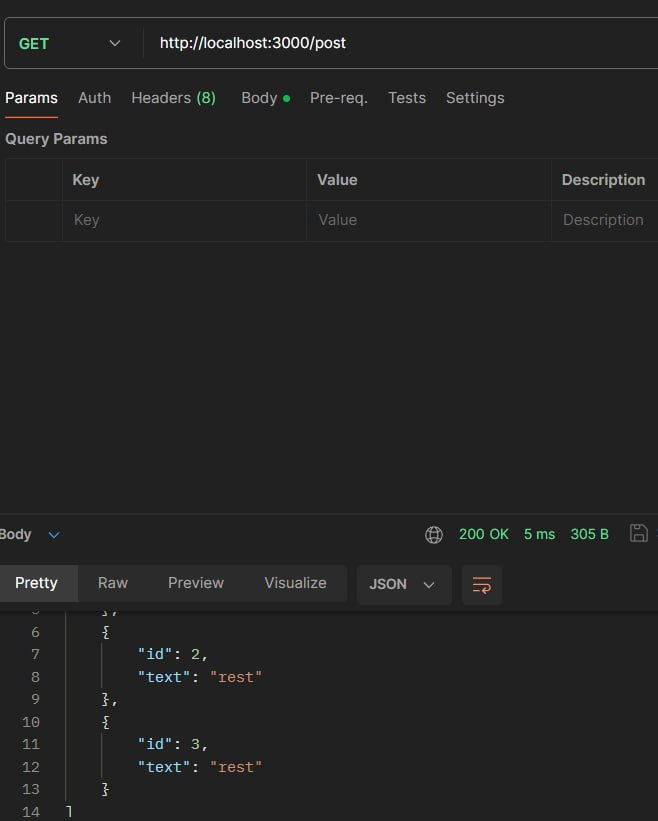
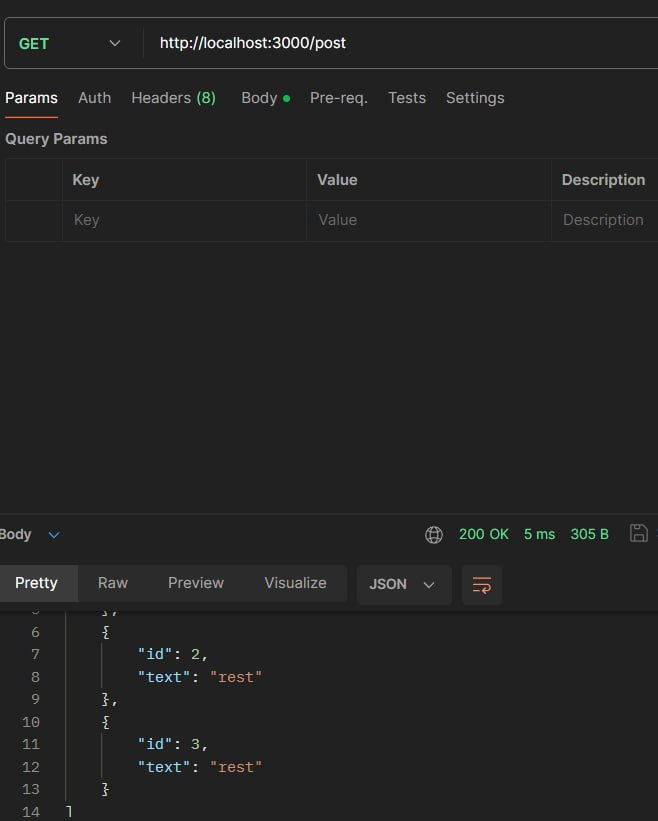


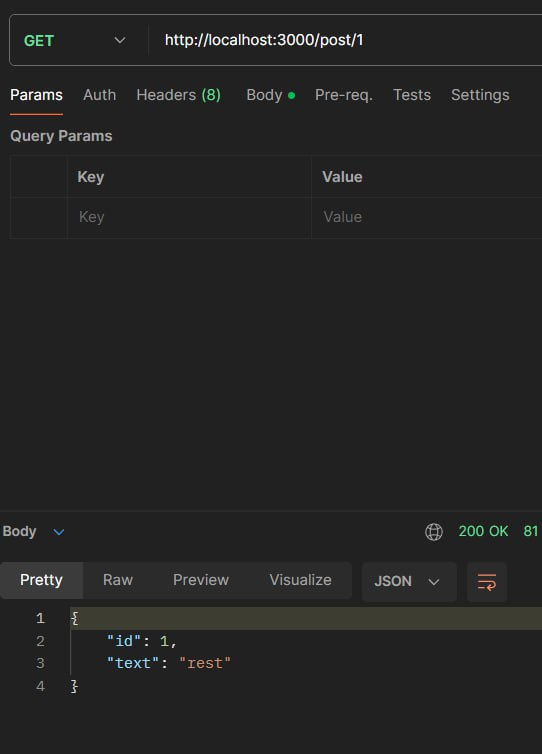
1. **Nest:** GET

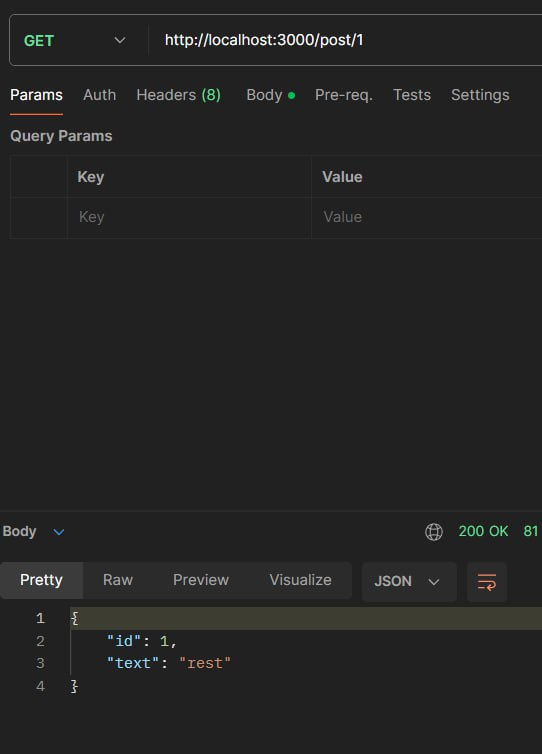




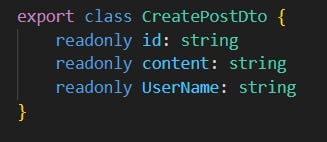
1. **Nest:** результат работы GET

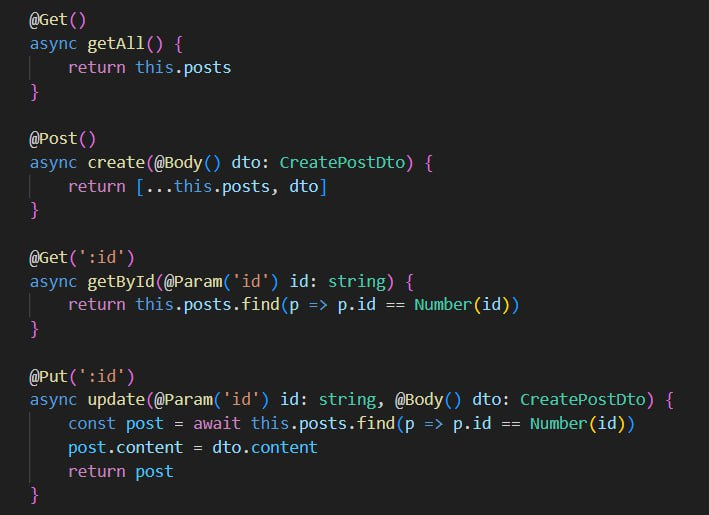




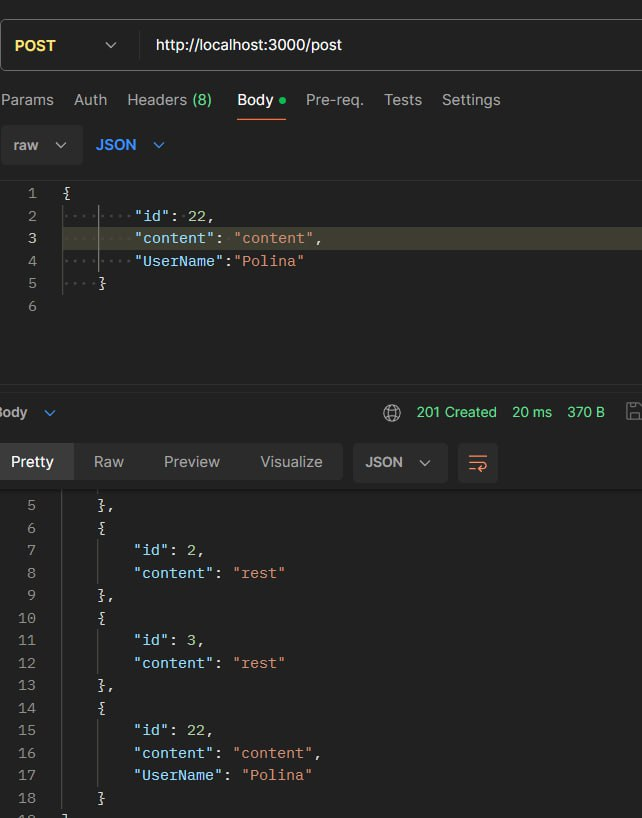
1. **Nest:** DTO – data transform object.



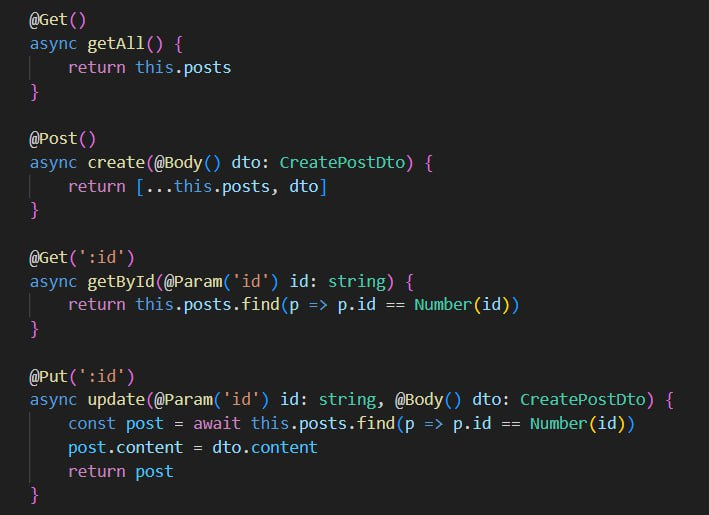
1. **Nest:** POST



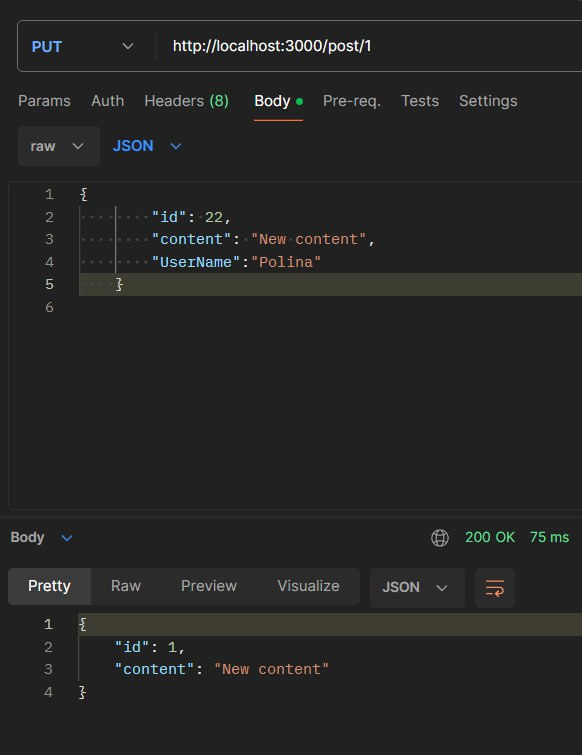
1. **Nest:** результат работы POST

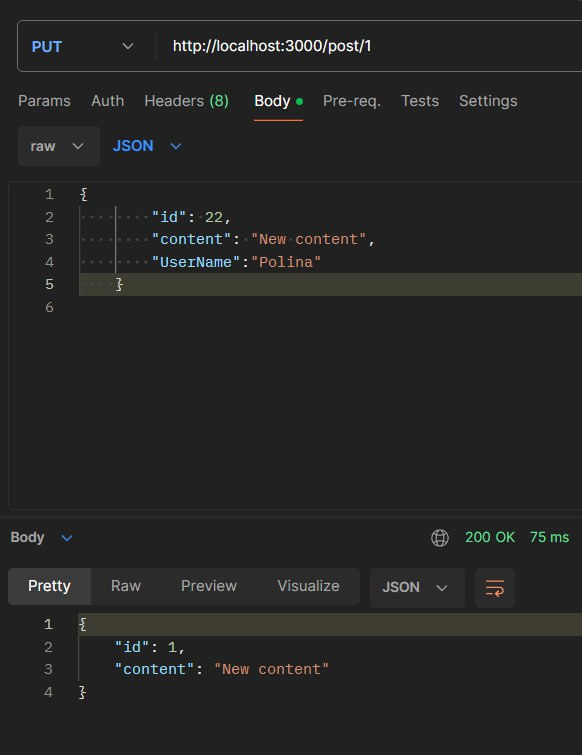


1. Nest: PUT

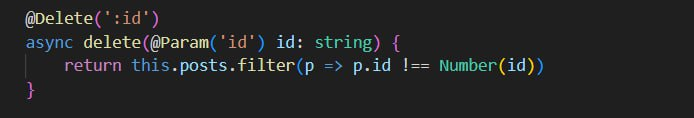


1. Nest: результат работы PUT

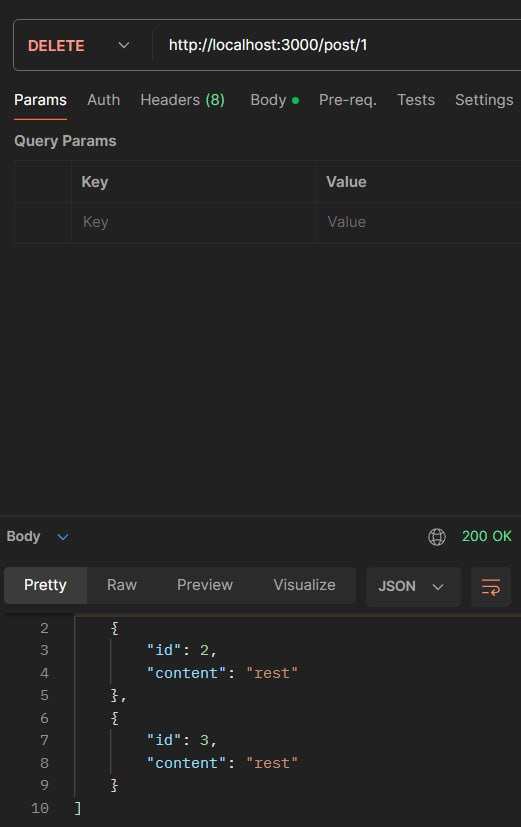


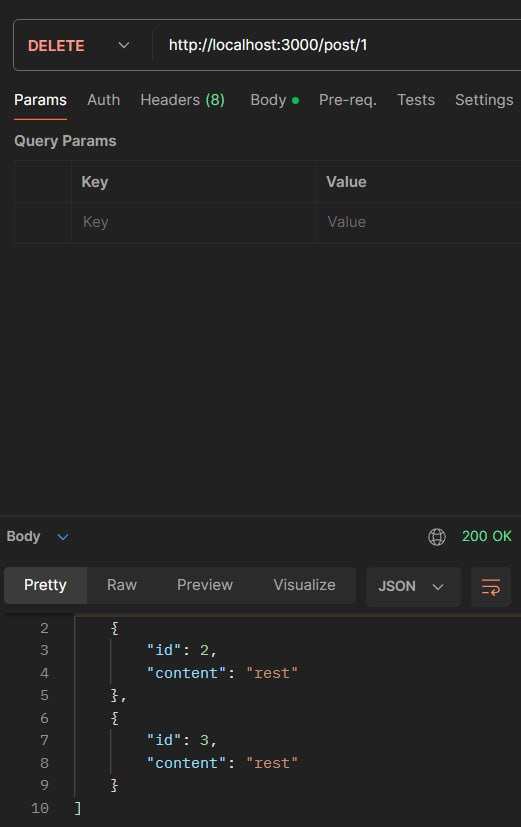


1. **Nest**: DELETE



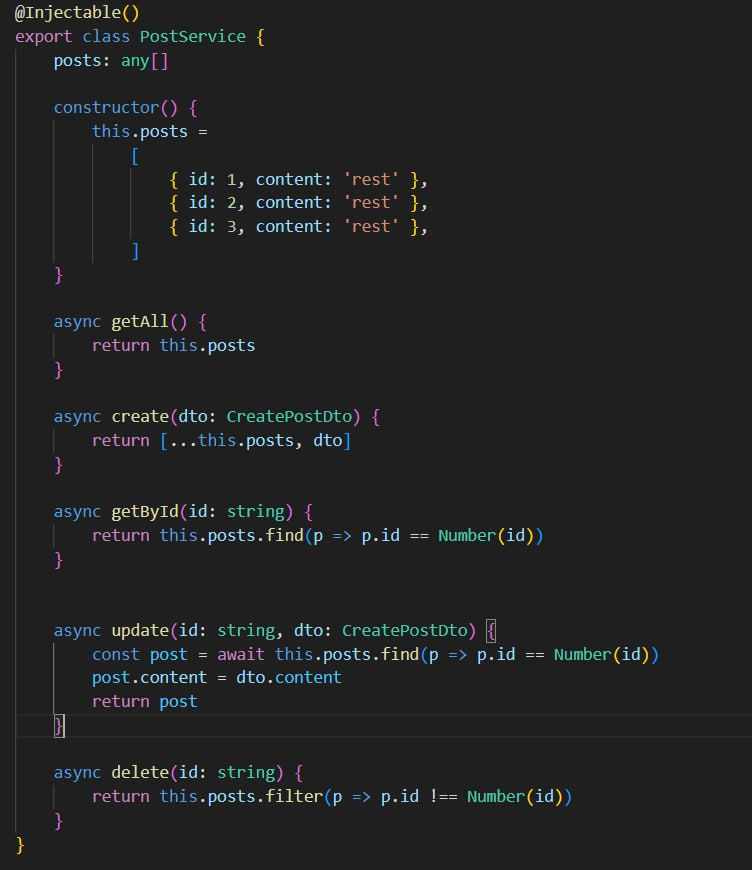
1. **Nest**: результат работы DELETE





1. **Nest:** controller -> service





1. **Nest:** позволяет хорошо писать тесты благодаря модульной структуре, где мы можем выбрать определенные контроллеры, модули и протестировать их изолированно, либо протестировать целиком.

