

# WEB. Разработка с Nest JS.

## Урок 4. Практика. Модули Nest.js.

### **Задача: Разработка системы управления пользователями и задачами.**

Вам нужно разработать простую систему управления пользователями и задачами с использованием NestJS.

Система должна состоять из двух функциональных модулей: UsersModule и TasksModule. Каждый модуль будет отвечать за управление своей областью ответственности: пользователи и задачи.

### **UsersModule.**

Отвечает за управление пользователями системы.

Содержит сервис UsersService и контроллер UsersController.

UsersService должен иметь методы для создания, получения списка пользователей и поиска пользователя по ID.

UsersController должен иметь маршруты для взаимодействия с пользователями:

- POST /users — создание нового пользователя.
- GET /users — получение списка всех пользователей.
- GET /users/:id — получение данных пользователя по ID.

Для хранения данных пользователей используйте в памяти массив (вместо полноценной базы данных).

### **TasksModule.**

Отвечает за управление задачами, которые могут быть назначены пользователям.

Содержит сервис TasksService и контроллер TasksController.

TasksService должен иметь методы для создания, получения списка задач и поиска задач по пользователю.

TasksController должен иметь маршруты для взаимодействия с задачами:

- POST /tasks — создание новой задачи.
- GET /tasks — получение списка всех задач.
- GET /tasks/user/:userId — получение списка задач, назначенных конкретному пользователю.

Задачи должны содержать информацию о том, какому пользователю они назначены.

**Взаимодействие модулей.**

TasksModule должен иметь доступ к UsersModule, чтобы проверять, существует ли пользователь, которому назначается задача.

При создании задачи в TaskService, перед назначением задачи пользователю, необходимо проверить, существует ли пользователь с указанным ID в UsersService.