

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Бархатова Наталья

Группа К3339

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

## Задача

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

## Ход работы

Все круды были ранее реализованы в ДЗ2, поэтому в этой работе я лишь добавила более четкое разделение на http коды. Добавлены коды ответа для создания (201 Created), успешного удаления без тела (204 No Content).

Ошибки поиска ресурса теперь возвращают 404 Not Found.

## Пример Rest-controller

```
@JsonController("/workouts")
@UseBefore(AuthMiddleware)
export class WorkoutController {
  @Post()
  @HttpCode(201)
  async create(@Body() data: Partial<Workout>) {
    const workout = workoutRepo.create(data);
    return await workoutRepo.save(workout);
  }

  @Get()
  @HttpCode(200)
  async getAll() {
    return await workoutRepo.find();
  }

  @Get("/:id")
  @OnUndefined(404)
  @HttpCode(200)
  async getOne(@Param("id") id: string) {
    return await workoutRepo.findOne({
      where: { id },
      relations: ["trainingPlans"],
    });
  }

  @Put("/:id")
  @OnUndefined(404)
  @HttpCode(200)
  async update(@Param("id") id: string, @Body() data: Partial<Workout>) {
    const workout = await workoutRepo.findOne({ where: { id } });
    if (!workout) return undefined;
    workoutRepo.merge(workout, data);
    return await workoutRepo.save(workout);
  }

  @Delete("/:id")
```

```
@OnUndefined(404)
@HttpCode(204)
async delete(@Param("id") id: string) {
    const result = await workoutRepo.delete(id);
    if (result.affected === 0) return undefined;
    return;
}
}
```

## **Вывод**

В контроллерах реализован корректный и единообразный подход к использованию HTTP-статусов. Такой подход соответствует REST-стандартам и улучшает читабельность и поддержку кода за счет централизованной обработки ошибок.