

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Русинов Василий

Группа К3340

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

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

Вариант: сайт для поиска работы.

Ход работы

Большая часть RESTful API в рамках выполнения ЛР1, в рамках выполнения данной лабораторной работы были добавлены статусы ответов HTTP:

- 201 Created → при успешном создании ресурса
- 200 OK → стандартный успешный ответ
- 204 No Content → успешно удалён, но ничего не возвращается
- 404 Not Found → ресурс не найден
- 400 Bad Request → если клиент отправил неправильные данные
- 500 Internal Server Error → на случай непредвиденных ошибок

Пример UserController:

```
export class UserController {
  async create(req: Request, res: Response) {
    try {
      const user = await service.create(req.body);
      res.status(201).json(user);
    } catch (err: unknown) {
      res.status(400).json({ error: (err as Error).message });
    }
  }

  async getAll(req: Request, res: Response) {
    try {
      const users = await service.getAll();
      res.status(200).json(users);
    } catch (err: unknown) {
      res.status(500).json({ error: (err as Error).message });
    }
  }

  async getById(req: Request, res: Response) {
    try {
      const user = await service.getById(Number(req.params.id));
      if (!user) return res.status(404).json({ error: "User not found" });
      res.status(200).json(user);
    } catch (err: unknown) {
      res.status(500).json({ error: (err as Error).message });
    }
  }
}
```

```

    async getByEmail(req: Request, res: Response) {
      try {
        const user = await service.getByEmail(req.params.email);
        if (!user) return res.status(404).json({ error: "User not found" });
        res.status(200).json(user);
      } catch (err: unknown) {
        res.status(500).json({ error: (err as Error).message });
      }
    }

    async update(req: Request, res: Response) {
      try {
        const user = await service.update(Number(req.params.id), req.body);
        if (!user) return res.status(404).json({ error: "User not found" });
        res.status(200).json(user);
      } catch (err: unknown) {
        res.status(400).json({ error: (err as Error).message });
      }
    }

    async delete(req: Request, res: Response) {
      try {
        const ok = await service.delete(Number(req.params.id));
        if (!ok) return res.status(404).json({ error: "User not found" });
        res.status(204).send();
      } catch (err: unknown) {
        res.status(500).json({ error: (err as Error).message });
      }
    }
  }
}

```

Вывод

В ходе выполнения лабораторной работы был реализован RESTful API средствами express + typescript (используя ранее написанный boilerplate).