САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 2

Выполнила:

Красюк Карина

Группа К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

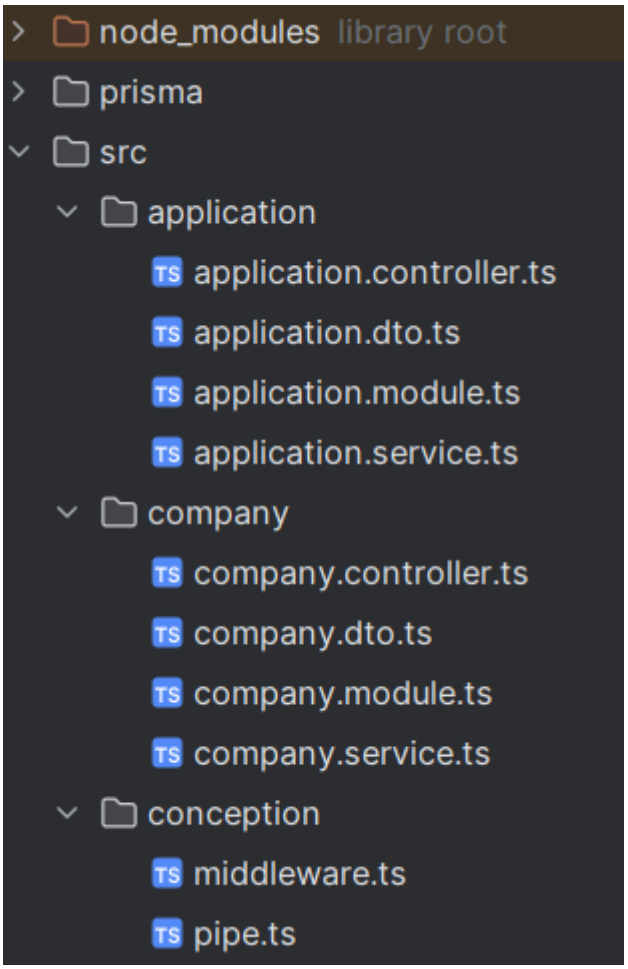
Реализовать RESTful API средствами express + typescript (используя ранее написанный boilerplate)

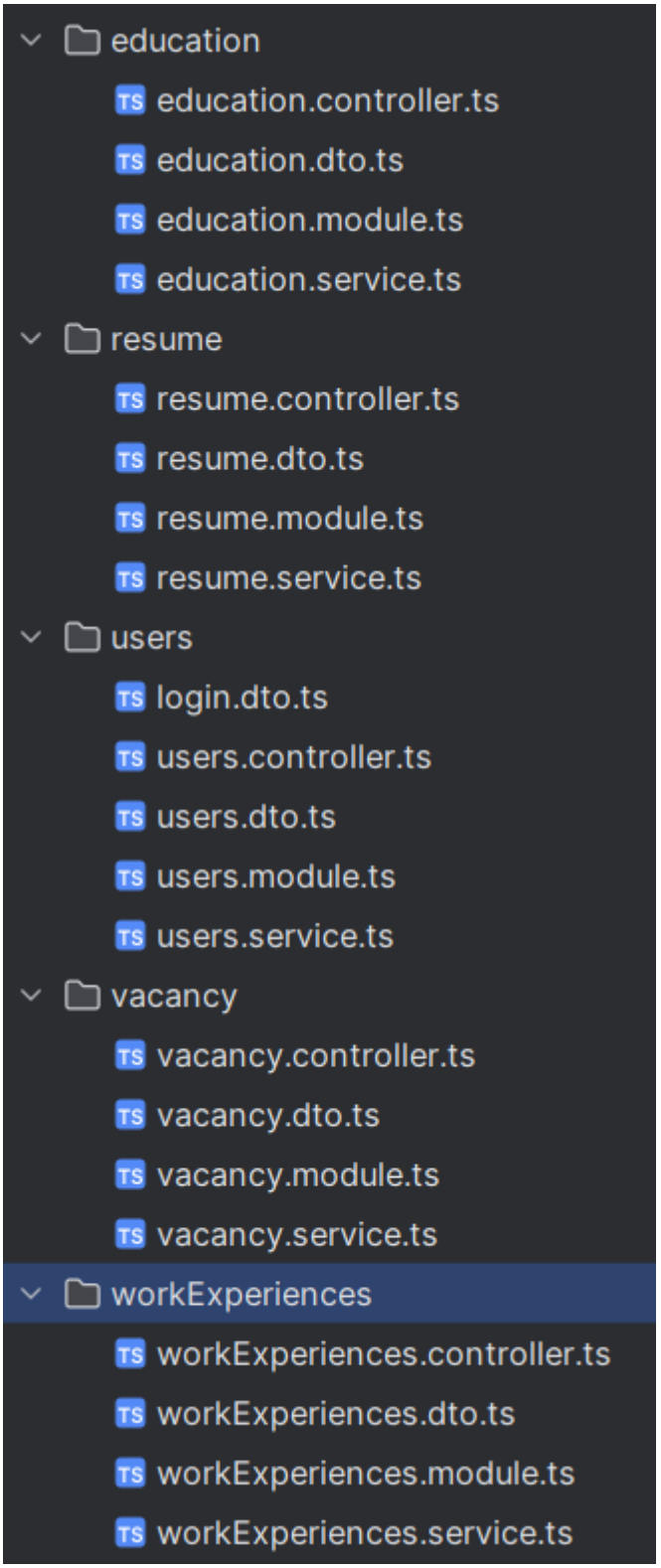
Ход работы

Реализованные контроллеры:

1. application
2. company
3. education
4. resume
5. users
6. vacancy
7. workExperiences

Для каждой сущности были реализованы контроллеры, сервисы и модули. Есть разделение на клиент сервер, а также не запоминаем состояние.





Пример кода education.controller

```
import {
  Body,
  Controller,
  Delete,
  Get,
  Param,
  Post,
  Put,
  Query,
  UseGuards,
```

```
  UsePipes,
  ValidationPipe,
} from '@nestjs/common';
import { ApiTags, ApiOperation, ApiResponse, ApiBearerAuth, ApiBody,
ApiParam } from '@nestjs/swagger';
import { EducationService } from './education.service';
import { ParseIntPipe } from '../conception/pipe';
import { CreateEducationsDto, TUpdateEducationsDto } from
'./education.dto';

@ApiTags('Education')
@Controller('educations')
export class EducationController {
  constructor(private readonly educationsService: EducationService) {}

  @Get()
  @ApiOperation({ summary: 'Получить все записи об образовании' })
  @ApiResponse({ status: 200, description: 'Список записей об образовании
успешно получен' })
  @ApiBearerAuth()
  findAll() {
    return this.educationsService.educationFindAll();
  }

  @Get(':id')
  @ApiOperation({ summary: 'Получить запись об образовании по ID' })
  @ApiResponse({ status: 200, description: 'Запись об образовании успешно
получена' })
  @ApiResponse({ status: 404, description: 'Запись об образовании не
найдена' })
  @ApiParam({ name: 'id', type: 'number', description: 'ID записи об
образовании' })
  getEducation(@Param('id', ParseIntPipe) id: number) {
    return this.educationsService.educationGetById(id);
  }

  @Post()
  @UsePipes(new ValidationPipe())
  @ApiOperation({ summary: 'Создать новую запись об образовании' })
  @ApiResponse({ status: 201, description: 'Запись об образовании успешно
создана' })
  @ApiResponse({ status: 400, description: 'Неверные данные' })
  @ApiBody({ type: CreateEducationsDto })
  @ApiBearerAuth()
  create(@Body() dto: CreateEducationsDto) {
    return this.educationsService.educationCreate(dto);
  }

  @Put(':id')
  @UsePipes(new ValidationPipe())
  @ApiOperation({ summary: 'Обновить запись об образовании' })
  @ApiResponse({ status: 200, description: 'Запись об образовании успешно
обновлена' })
  @ApiResponse({ status: 400, description: 'Неверные данные' })
  @ApiResponse({ status: 404, description: 'Запись об образовании не
найдена' })
  @ApiParam({ name: 'id', type: 'number', description: 'ID записи об
образовании' })
  @ApiBearerAuth()
  update(
      @Param('id', ParseIntPipe) id: number,
      @Body() dto: TUpdateEducationsDto,
  ) {
    return this.educationsService.educationUpdate(id, dto);
  }

  @Delete(':id')
```

```
  @UsePipes(new ValidationPipe())
  @ApiOperation({ summary: 'Удалить запись об образовании' })
  @ApiResponse({ status: 200, description: 'Запись об образовании успешно
удалена' })
  @ApiResponse({ status: 404, description: 'Запись об образовании не
найдена' })
  @ApiParam({ name: 'id', type: 'number', description: 'ID записи об
образовании' })
  @ApiBearerAuth()
  delete(@Param('id', ParseIntPipe) id: number) {
    return this.educationsService.educationDelete(id);
  }
}
```

Префикс АПИ

```
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/workExperiences, GET} route +0ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/workExperiences/:id, GET} route +1ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/workExperiences, POST} route +0ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/workExperiences/:id, PUT} route +1ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/workExperiences/:id, DELETE} route +1ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RoutesResolver] EducationController {/api/educations}: +0ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/educations, GET} route +1ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/educations/:id, GET} route +0ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/educations, POST} route +1ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/educations/:id, PUT} route +0ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/educations/:id, DELETE} route +2ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RoutesResolver] ApplicationController {/api/application}: +1ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/application, GET} route +1ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/application/:id, GET} route +0ms
[Nest] 19968  - 11.06.2025, 22:19:08     LOG [RouterExplorer] Mapped {/api/application, POST} route +1ms
```

Вывод

В данной лабораторной работе я реализовала RESTful API средствами express + typescript.