

"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"

Факультет «Инфокоммуникационных технологий»
Направление подготовки «09.03.03 Прикладная информатика»
Бакалаврская программа «Мобильные и сетевые технологии»

ОТЧЕТ
по лабораторной работе 1
по дисциплине «Бэк-энд разработка»

Выполнила: Матрохина А.С., К33402

Проверил: Добряков Д.И.

Санкт-Петербург
2022

Цель работы:

Нужно написать свой boilerplate на express + sequelize / TypeORM + typescript.

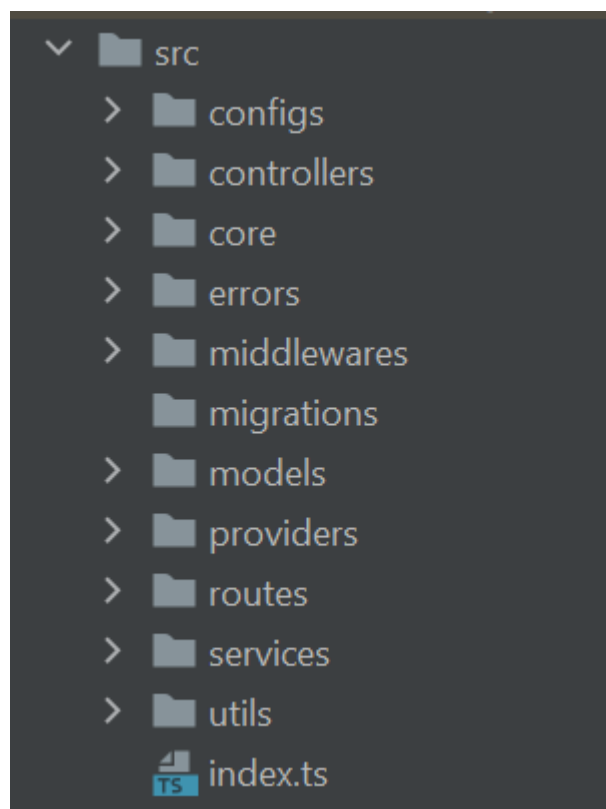
Должно быть явное разделение на:

- модели
- контроллеры
- роуты
- сервисы для работы с моделями (реализуем паттерн “репозиторий”)

Результаты:

За основу был взят boilerplate, приведенный для примера

<https://github.com/kantegory/express-sequelize-boilerplate>. Среди других подобных на Github, я сочла его наиболее понятным и подходящим под задачи дальнейших лабораторных работ. Шаблон имеет следующую структуру:



В ходе доработки были произведены некоторые улучшения. Во-первых, удалены лишние зависимости, а для оставшихся обновлены версии до самых актуальных:

```
"dependencies": {
  "bcrypt": "^5.0.1",
  "body-parser": "^1.20.0",
  "cors": "^2.8.5",
  "express": "^4.18.1",
  "ini": "^2.0.0",
  "passport": "^0.6.0",
  "passport-jwt": "^4.0.0",
  "reflect-metadata": "^0.1.13",
  "sequelize": "^6.19.2",
  "sequelize-typescript": "^2.1.3",
  "sqlite3": "^5.0.8",
  "uuid": "^8.3.2"
},
"devDependencies": {
  "@types/bcrypt": "^5.0.0",
  "@types/cors": "^2.8.12",
  "@types/express": "^4.17.13",
  "@types/express-session": "^1.17.4",
  "@types/flat": "^5.0.2",
  "@types/ini": "^1.3.31",
  "@types/node": "^17.0.41",
  "@types/passport": "^1.0.8",
  "@types/passport-jwt": "^3.0.6",
  "@types/uuid": "^8.3.4",
  "@types/validator": "^13.7.3",
  "@typescript-eslint/eslint-plugin": "^5.27.1",
  "@typescript-eslint/parser": "^5.9.1",
  "eslint": "^8.17.0",
  "nodemon": "^2.0.16",
  "prettier": "2.6.2",
  "sequelize-cli": "^6.4.1",
  "ts-node": "^10.4.0",
  "typescript": "4.6.4"
}
```

Исследование зависимостей показало, что библиотека `adminjs` перестала работать на территории России, а попытка найти аналоги не удалась успехом, поэтому было решено удалить админ-панель вовсе.

Также был удален `TsLint`, который считается deprecated. Для контроля стиля кода был дополнительно установлен `prettier` со стандартным файлом конфигурации, взятым из официальной документации:

```
{
  "trailingComma": "es5",
  "tabWidth": 4,
  "semi": false,
  "singleQuote": true
}
```

И последнее – модель пользователя была изменена, чтобы удовлетворять дальнейшему выбранному варианту (№3, бронирование отелей). Добавлено разделение на имя и фамилию, а также паспортные данные

```
@Table
class User extends Model {
    @AllowNull( allowNull: false)
    @Column
    firstName: string

    @AllowNull( allowNull: false)
    @Column
    lastName: string

    @AllowNull( allowNull: false)
    @Unique
    @Column
    email: string

    @AllowNull( allowNull: false)
    @Column
    password: string

    @AllowNull( allowNull: false)
    @Column
    passportSeries: string

    @AllowNull( allowNull: false)
    @Column
    passportNumber: string

    @Column
    bio: string
}
```

Вывод:

В ходе работы было реализовано бекэнд-приложение на node.js, написаны контроллеры для end поинтов и реализовали паттерн «репозиторий» для работы с БД и описанными моделями.