

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа 1

Выполнил:
Комиссаров Александр
К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2022 г.

Задача

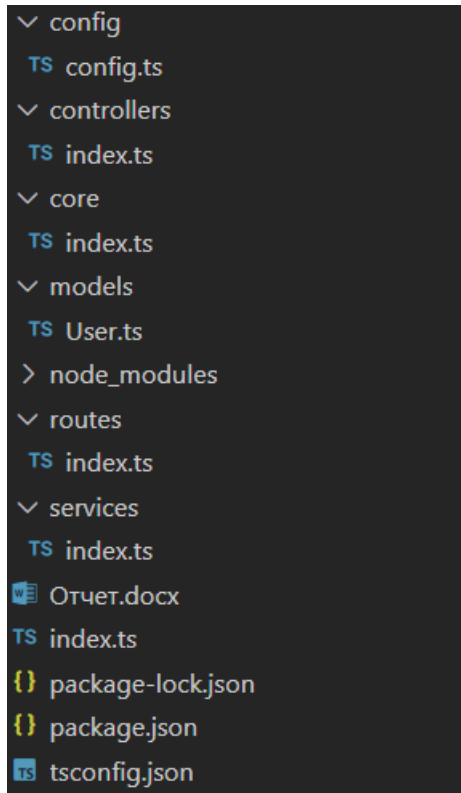
Нужно написать свой boilerplate на express + sequelize / TypeORM + typescript.

Должно быть явное разделение на:

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

Ход работы

Структура бойлерплейта:



Модель пользователя:

```
import { Table, Column, Model } from 'sequelize-typescript'

@Table
export default class User extends Model {
  @Column
  name: string

  @Column
  surname: string

  @Column
  email: string

  @Column
  address: string
}
```

Контроллер с методами post и get

```
import DefaultService from '../services/index'

export default class ExampleController {

  private service = new DefaultService()

  post = async (request: any, response: any) => {
    try {
      const user = request.body
      await this.service.add(user.name, user.surname, user.email, user.address)
      response.send('Successfully added')
    } catch (error: any) {
      response.status(400).send(error.message)
    }
  }

  get = async (request: any, response: any) => {
    try {
      const data = await this.service.get()
      response.send(data)
    } catch (error: any) {
      response.status(400).send(error.message)
    }
  }
}
```

Сервис, позволяющий добавить пользователя и получить всех пользователей

```
import User from '../models/User'
import { sequelize } from '../config/config'

export default class DefaultService {

  private repo = sequelize.getRepository(User)

  add(name: string, surname: string, email: string, address: string) {
    this.repo.create({ name: name, surname: surname, email: email, address: address })
  }

  get() {
    return this.repo.findAll()
  }
}
```

Роутер, позволяющий по маршруту / вызвать методы post и get

```
import express from "express"
import ExampleController from '../controllers/index'

const router: express.Router = express.Router()

const exampleController = new ExampleController()

router
  .route('/')
  .get(exampleController.get)
  .post(exampleController.post)

export default router
```

Вывод

В результате работы мы получили универсальный бойлерплейт с моделью пользователя и базовыми методами, который может быть использован в качестве основы для большинства бэкенд-приложений с необходимым функционалом. В работе использовались Express, Sequelize и Typescript.