

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

**Отчет**

**Лабораторная работа №1: Boilerplate**

**Выполнила:**

**Кулагина Светлана**

**Группа К33412**

**Проверил:**

**Добряков Д. И.**

**Санкт-Петербург  
2023 г.**

**Цель:** написать свой boilerplate на express + sequelize / TypeORM typescript

**Задачи:**

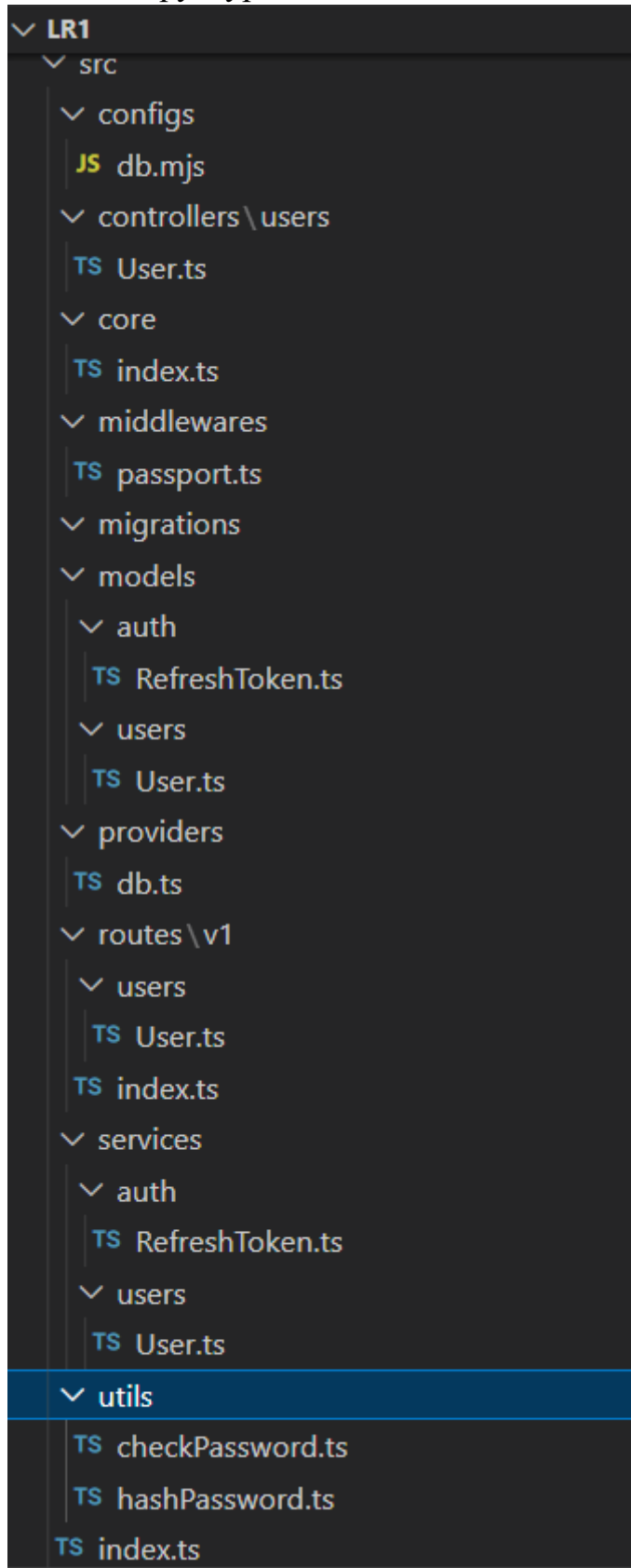
Создать явное разделение на следующие папки:

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

**Ход работы:**

1. Для успешного запуска проекта, мне необходимо было создать файл Makefile, для удобства работы, где я реализовала следующие команды:
  - **Make init**
  - **Make install\_d**
  - **Make start**
  - **Make build**
  - **Make migrate**
  - **Make lint**
2. В дальнейшем надо было инициализировать проект, установить все зависимости, создать файл .env, и сделать make start

В итоге структура папки src выглядит следующим образом:



Также мной были созданы модели:

- Модель пользователя

```

import { AllowNull, BeforeCreate, BeforeUpdate, Column, Model, Table, Unique } from 'sequelize-typescript'
import hashPassword from '../utils/hashPassword'

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

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

  ...@Unique
  ...@Column
  ...email: string

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

  ...@BeforeCreate
  ...@BeforeUpdate
  ...static generatePasswordHash(instance: User) {
  ...  const { password } = instance

  ...  if (instance.changed('password')) {
  ...    instance.password = hashPassword(password)
  ...  }
  ...}
}

```

## - Модель RefreshToken'a

```

import { Table, Column, Model, Unique, AllowNull, ForeignKey } from 'sequelize-typescript'
import User from '../users/User'

@Table
class RefreshToken extends Model {
  ...@Unique
  ...@AllowNull(false)
  ...@Column
  ...token: string

  ...@ForeignKey(() => User)
  ...@Column
  ...userId: number
}

export default RefreshToken

```

Маршруты для различных get и post запросов можно найти по следующим routes:

```
router.post('/', controller.create)

router.get('/profile', passport.authenticate('jwt', { session: false }), controller.me)

router.get('/:id', controller.get)

router.post('/login', controller.auth)

router.post('/refresh', controller.refreshToken)
```

**Вывод:** В ходе лабораторной работы мной были получены практические навыки по созданию boilerplate на express + sequelize / TypeORM typescript, поняла важность и полезность boilerplate на практике.