

"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"

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

ОТЧЕТ
по лабораторной работе 2
по дисциплине «Бэк-энд разработка»

Выполнила: Матрохина А.С., К33402

Проверил: Добряков Д.И.

Санкт-Петербург
2022

Цель работы:

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

Вариант:

Платформа для поиска и бронирования номера в отеле/квартире/хостеле (<https://airbnb.com>)

- Вход
- Регистрация
- Страница бронирований пользователя
- Страница для поиска номера с возможностью выбора города, времени заселения, количеству гостей

Результаты:

Был выбран вариант №3. Платформа для поиска и бронирования номера в отеле (функционал, похожий на Airbnb.com)

В ходе работы были реализованы следующие модели БД:

- Пользователь

```
@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
}
```

- Отель

```
@Table
class Hotel extends Model {
  @AllowNull( allowNull: false)
  @Column
  name: string

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

  @AllowNull( allowNull: false)
  @Column
  stars: number

  @HasMany( associatedClassGetter: () => Room)
  rooms: Room[]
}
```

- Комната в отеле

```
@Table
class Room extends Model {
  @AllowNull(allowNull: false)
  @Column
  number: string

  @AllowNull(allowNull: false)
  @Column
  floor: number

  @AllowNull(allowNull: false)
  @Column
  capacity: number

  @AllowNull(allowNull: false)
  @Column
  price: number

  @Column
  description: string

  @ForeignKey(relatedClassGetter: () => Hotel)
  @Column
  hotelId: number

  @BelongsTo(associatedClassGetter: () => Hotel)
  hotel: Hotel

  @HasMany(associatedClassGetter: () => Booking)
  bookings: Booking[]
}
```

- Бронирование

```
@Table
class Booking extends Model {
  @AllowNull(allowNull: false)
  @Column
  dateFrom: Date

  @AllowNull(allowNull: false)
  @Column
  dateTo: Date

  @ForeignKey(relatedClassGetter: () => User)
  @Column
  userId: number

  @BelongsTo(associatedClassGetter: () => User)
  user: User

  @ForeignKey(relatedClassGetter: () => Room)
  @Column
  roomId: number

  @BelongsTo(associatedClassGetter: () => Room)
  room: Room
}
```

Далее были реализованы необходимые методы для работы с сервисом. При помощи Postman была проверена работоспособность системы, и корректная обработка ошибок

```
POST http://localhost:8000/v1/users
POST http://localhost:8000/v1/users/login
GET http://localhost:8000/v1/users/profile/1
GET http://localhost:8000/v1/users/profile
POST http://localhost:8000/v1/hotels
GET http://localhost:8000/v1/hotels
GET http://localhost:8000/v1/hotels/1
POST http://localhost:8000/v1/hotels/1/rooms
GET http://localhost:8000/v1/hotels/1/rooms
POST http://localhost:8000/v1/bookings
GET http://localhost:8000/v1/hotels/city/Saint Petersburg
GET http://localhost:8000/v1/hotels/name/FiveStars SPB
```

Вывод:

В ходе работы было реализовано RESTful API приложение для поиска и бронирования номера в отеле/квартире/хостеле, используя ранее написанный boilerplate.