

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бэк-энд разработка

Отчет

Лабораторная работа №3

Выполнила:

Мищенко Виолетта

Группа К33402

Проверил:

Добряков Д.И.

Санкт-Петербург

2022 г.

Цель работы: необходимо реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций приложения.

Отдельный микросервис Hotels, отвечающий за создание, просмотр списка отелей и поиска отдельного отеля по определенному id.

```
import express from 'express'
import hotelRoutes from './hotels/Hotel'

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

router.use('/hotels', hotelRoutes)
export default router
```

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

  @AllowNull(false)
  @Column
  rating: number

  @AllowNull(false)
  @Column
  location: string

  @Column
  averagePrice: number
}

export default HotelModel
```

```
list = async (request: any, response: any) => {
  const { location } = request.query

  try {
    const hotels = await this.hotelService.list(location)

    return response.send(hotels)
  } catch (error: any) {
    response.status(404).send({ 'error': error.message })
  }
}
```

```

create = async (request: any, response: any) => {
  const { body } = request

  try {
    const hotel: HotelModel | HotelError = await this.hotelService.create(body)

    response.status(201).send(hotel)
  } catch (error: any) {
    response.status(400).send({ 'error': error.message })
  }
}

```

```

item = async (request: any, response: any) => {
  try {
    const hotel = await this.hotelService.item(request.params.id)
    return response.send(hotel)
  } catch (error: any) {
    response.status(404).send({ 'error': error.message })
  }
}

```

```

class HotelService {
  async list(location?: string) {
    if (location) {
      return await HotelModel.findAll({ where: { location } })
    } else {
      return await HotelModel.findAll()
    }
  }

  async create(data: object) {
    try {
      const hotel = await HotelModel.create(data)
      return hotel.toJSON()
    } catch (e: any) {
      const errors = e.errors.map((error: any) => error.message)
      throw new HotelError(errors)
    }
  }
}

```

```
    async item(hotelId: number) {  
      try {  
        return await HotelModel.findByPk(hotelId)  
      } catch (e: any) {  
        const errors = e.errors.map((error: any) => error.message)  
        throw new HotelError(errors)  
      }  
    }  
  }  
}
```

Вывод: в ходе лабораторной работы был реализован отдельный микросервис Hotels, выполняющий такие содержательные функции как создание, просмотр списка и получение отеля по номеру id.