

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №3

Выполнил:

Коровин Александр

Группа:

К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2022 г.

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

Ход работы:

ValidateSchema

```

1  import Joi, { ObjectSchema } from "joi";
2  import { NextFunction, Request, Response } from "express";
3  import { IUser } from "../models/User";
4  import { IBooking } from "../models/Booking";
5  import Logging from "../library/Logging";
6
7  export const ValidateJoi = (schema: ObjectSchema) => {
8    return async (req: Request, res: Response, next: NextFunction) => {
9      try {
10        await schema.validateAsync(req.body);
11
12        next();
13      } catch (error) {
14        Logging.error(error);
15
16        return res.status(422).json({ error });
17      }
18    };
19  };
20
21  export const Schemas = {
22    user: {
23      create: Joi.object<IUser>({
24        firstName: Joi.string().required(),
25        lastName: Joi.string().required(),
26        email: Joi.string().email({
27          minDomainSegments: 2,
28          tlds: { allow: ["com", "net"] },
29        }),
30        password: Joi.string().regex(/^[a-zA-Z0-9]{3,30}$/),
31      }),
32      update: Joi.object<IUser>({
33        firstName: Joi.string().required(),
34        lastName: Joi.string().required(),
35        email: Joi.string().email({
36          minDomainSegments: 2,
37          tlds: { allow: ["com", "net"] },
38        }),
39        password: Joi.string().regex(/^[a-zA-Z0-9]{3,30}$/),
40      }),
41      login: Joi.object<IUser>({
42        email: Joi.string().email({
43          minDomainSegments: 2,
44          tlds: { allow: ["com", "net"] },
45        }),
46        password: Joi.string().regex(/^[a-zA-Z0-9]{3,30}$/),
47      }),
48    },
49    booking: {
50      create: Joi.object<IBooking>({
51        checkIn: Joi.date().greater("now").required(),
52        checkOut: Joi.date().greater(Joi.ref("from")).required(),
53      }),
54      update: Joi.object<IBooking>({
55        checkIn: Joi.date().greater("now").required(),
56        checkOut: Joi.date().greater(Joi.ref("from")).required(),
57      }),

```

Файл валидации схем, в частности валидация пароля, емэйла и дат въезда и отъезда.

Utils

hashPassword

```
1 import bcrypt from "bcrypt";
2
3 export default (password: string): string =>
4 |   bcrypt.hashSync(password, bcrypt.genSaltSync(8));
5
```

Хеширование пароля

checkPassword

```
1 import bcrypt from "bcrypt";
2
3 export default (user: any, password: string) => {
4 |   return bcrypt.compareSync(password, user.password);
5 | };
6
```

Проверка пароля

generateAccessToken

```
1 import { config } from "../config/config";
2 import { sign } from "jsonwebtoken";
3
4 export default (email: string) => {
5 |   return sign(email, config.jwt.key);
6 | };
7
```

Генерация токена доступа

Вывод: реализован отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций.