

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-енд разработка

Отчет

Лабораторная работа № 3

Выполнил:

Легин Денис

Группа

К33402

Проверил:

Добряков Д.И.

Санкт-Петербург

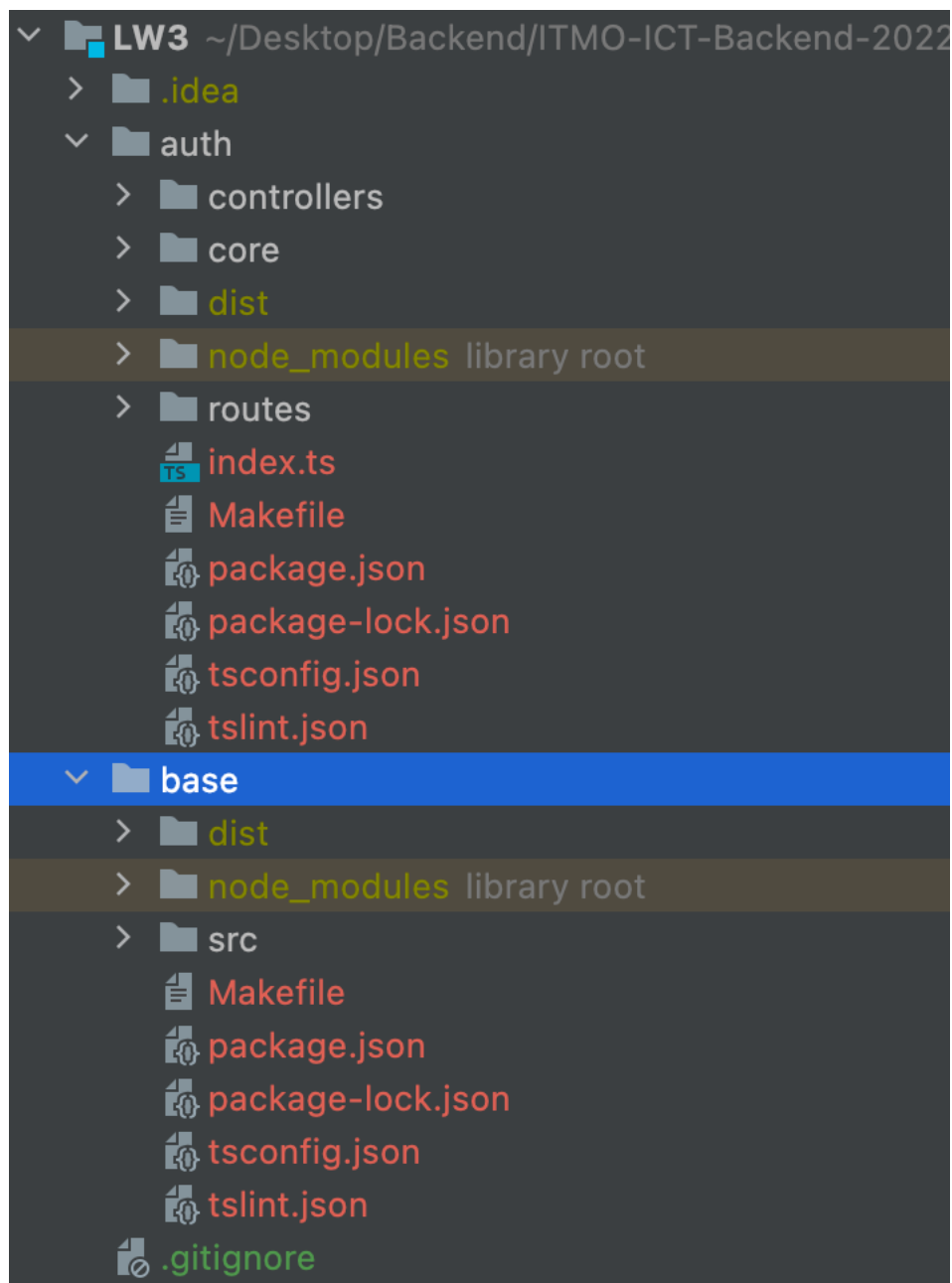
2022 г.

Задача

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

Ход работы

Структура приложения:



Контроллер микросервиса авторизации

```
export default class AuthController {

  post = async (request: any, response: any) => {
    const { email, password } = request.body
    const secretOrKey = 'test101'
    if (email && password) {
      const user = await axios.get( url: "http://localhost:8080/v1/user", config: {
        params: {
          email
        }
      })
      if (!user) {
        response.status(401).json({ msg: 'No such user found', user })
      }
      if (user?.data.password === password) {
        const payload = { id: user?.data.id }
        const jwt = require('jsonwebtoken')
        const token = jwt.sign(payload, secretOrKey)
        response.json({ msg: 'ok', token })
      } else {
        response.status(401).json({ msg: 'Password is incorrect' })
      }
    }
  }
}
```

Роутер авторизации:

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

const authController = new AuthController()

router
  .route( prefix: '/auth')
  .post(authController.post)

export default router
```

Авторизационный запрос, пользователь не зарегистрирован:

http://127.0.0.1:8180/v1/auth

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:8180/v1/auth
- Body Type:** raw (selected)
- Request Body:**

```
1 {  
2   "email": "mr.zooko@mail.ru",  
3   "password": "aboba"  
4 }
```
- Response Body:**

```
1 {  
2   "msg": "Password is incorrect"  
3 }
```

Авторизационный запрос, пользователь зарегистрирован

```
1 {  
2   "msg": "ok",  
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNjU1MjMwNTE0fQ.ZqEstuna-1UBzpGrLNRjq1ZatAhxAkErKXYvkUu72zM"  
4 }
```

Регистрация пользователя в микросервисе base

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:8080/v1/user
- Body Type:** raw (selected)
- Body Content:**

```
1 {  
2   "nickname": "aN00Bis666",  
3   "name": "Denis",  
4   "surname": "Login",  
5   "email": "mr.zooko@mail.ru",  
6   "password": "aboba"
```
- Response Tab:** Body (selected), showing:

```
1 Added 'aN00Bis666' user
```

Вывод:

В ходе работы было создано два микро сервиса, работающие друг с другом посредством http запросов. Первый микро сервис осуществляет аутентификацию пользователей, а второй – основную работу с приложением – бронирование отелей и прочие взаимодействия с ними.