

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №3: Микросервисы

Выполнил:

Попов Ньургун

К33412

Проверил:

Добряков Д. И.

Санкт-Петербург

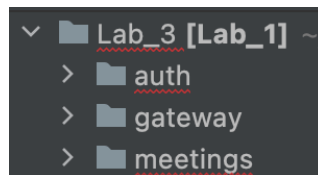
2023 г.

Задача

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

Ход работы

1. Структура проекта (сервисы):



2. Gateway:

```
import express from "express"
import proxy from "express-http-proxy"

const app = express();

app.use('/auth', proxy('http://localhost:8001'))
app.use('/meetings', proxy('http://localhost:8002'))

app.listen(8000, () => {
  console.log(`Running server on port 8000`)
})
```

3. Auth:

```
import express from "express"
import { UserController } from "../../controllers/auth"
import passport from "../../middlewares/passport"

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

const controller: UserController = new UserController()

routes.route('/signup')
  .post(controller.signup)

routes.route('/login')
  .post(controller.login)

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

routes.route('/accessToken')
  .post(controller.accessToken)

routes.route('/:id')
  .get(controller.get)

export default routes
```

4. Meetings:

Event:

```
import express from "express"
import { EventController } from "../controllers/Event"

const routes: express.Router = express.Router()
const controller: EventController = new EventController()

routes.route('/filter')
  .get(controller.filter)

routes.route('/')
  .get(controller.calendar)

routes.route('/author/:id')
  .get(controller.author)

routes.route('/:id')
  .get(controller.get)

export default routes
```

UserEvent:

```
import express from "express"
import { UserEventController } from "../controllers/UserEvent"
import passport from "../middlewares/passport";

const routes: express.Router = express.Router()
const controller: UserEventController = new UserEventController()

routes.route('/createEvent')
  .post(passport.authenticate('bearer', { session: false })),
  controller.createEvent)

routes.route('/updateEvent/:id')
  .patch(passport.authenticate('bearer', { session: false })),
  controller.updateEvent)

routes.route('/deleteEvent/:id')
  .delete(passport.authenticate('bearer', { session: false })),
  controller.deleteEvent)

routes.route('/joinTheEvent')
  .post(passport.authenticate('bearer', { session: false })),
  controller.joinTheEvent)

routes.route('/getEvent')
  .get(passport.authenticate('bearer', { session: false })),
  controller.getEvent)

export default routes
```

Вывод

В результате выполненной работы: бал реализован отдельный микросервис, выполняющий какую-либо содержательную функцию из всего арсенала функций вашего приложения.