

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Бек-энд разработка

Отчет

Лабораторная работа 3

Выполнил:

Дао Куанг Ань

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

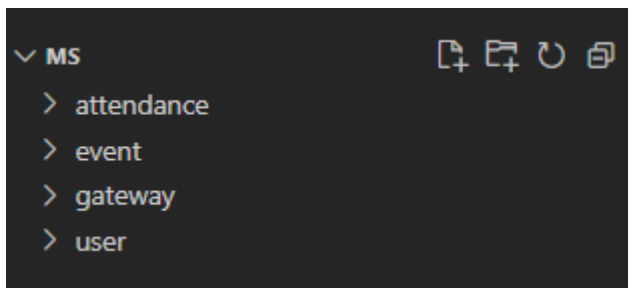
2022 г.

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

Вариант: Платформа для поиска профессиональных мероприятий

## Ход работы

### 1. Дерево файла



### 2. Эндпоут для сообщения между серверами

```
const AttendanceController = require('../controllers/attendance');

const { PublishUserEvent } = require("../utils");

const UserAuth = require('../middlewares/auth');

module.exports = (app) => {

    const service = new AttendanceController();

    app.post('/attendance/create', UserAuth, async (req, res, next) => {

        const { _id } = req.user;

        const { eventId } = req.body;

        try {

            const { data } = await service.Participation({_id, eventId});

            const payload = await service.GetAttendancePayload(_id, data, 'ATTENDANCE');
```

```

        PublishUserEvent(payload);

        return res.status(200).json(data);

    } catch (err) {
        next(err)
    }

});

app.get('/attendance/list',UserAuth, async (req,res,next) => {

    const { _id } = req.user;

    try {
        const { data } = await service.GetParticipation(_id);
        return res.status(200).json(data);
    } catch (err) {
        next(err);
    }

});

}

```

### 3. Controller где отправляем мероприятию к user серверу.

```

async Participation(userInput){

```

```

const { _id, eventId } = userInput

try {
  const attendRes = await this.repository.NewParticipation(_id, eventId);

  return FormateData(attendRes);
} catch (err) {
  throw new APIError('Data Not found', err)
}
}

async GetAttendancePayload(userId,attendance,event){

  if(attendance){

    const payload = {

      event: event,

      data: { userId, attendance }

    };

    return payload

  }else{

    return FormateData({error: 'No event participations'});

  }

}

```

## 4. Реализация

```

module.exports.FormateData = (data) => {

  if(data){

    return { data }

  }else{

    throw new Error('Data Not found!')

  }

}

```

```

module.exports.PublishUserEvent = async(payload) => {

    axios.post('http://localhost:8000/user/app-events', {

        payload

    })

}

```

```

async NewParticipation(userId, eventId) {

    try {

        const attendId = uuidv4();

        const request = new AttendanceModel({

            attendId,

            userId,

            eventId,

            status: 'received',

        })

        const attendRes = await request.save();

        return attendRes;

    } catch(e) {

        throw new Error(e)

    }

}

}

```

## Вывод

- Реализован отдельный микросервис, выполняющий функцию участия пользователя в мероприятии.