

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №3: Микросервисная архитектура

Выполнила:

Кулагина Светлана

Группа К33412

Проверил:

Добряков Д. И.

**Санкт-Петербург
2023 г.**

Цель: необходимо реализовать отдельный микросервис, выполняющий какую-либо содержательную функцию

Ход работы:

Я разделила свое приложение на два микросервиса и добавила общую точку gateway, которая будет распределять запросы по нашим микросервисам.

Структура приложения:

```
> auth
> dogs_of_company
> gateway
```

Микросервис **auth** содержит всё, что связано с user и аторизацией

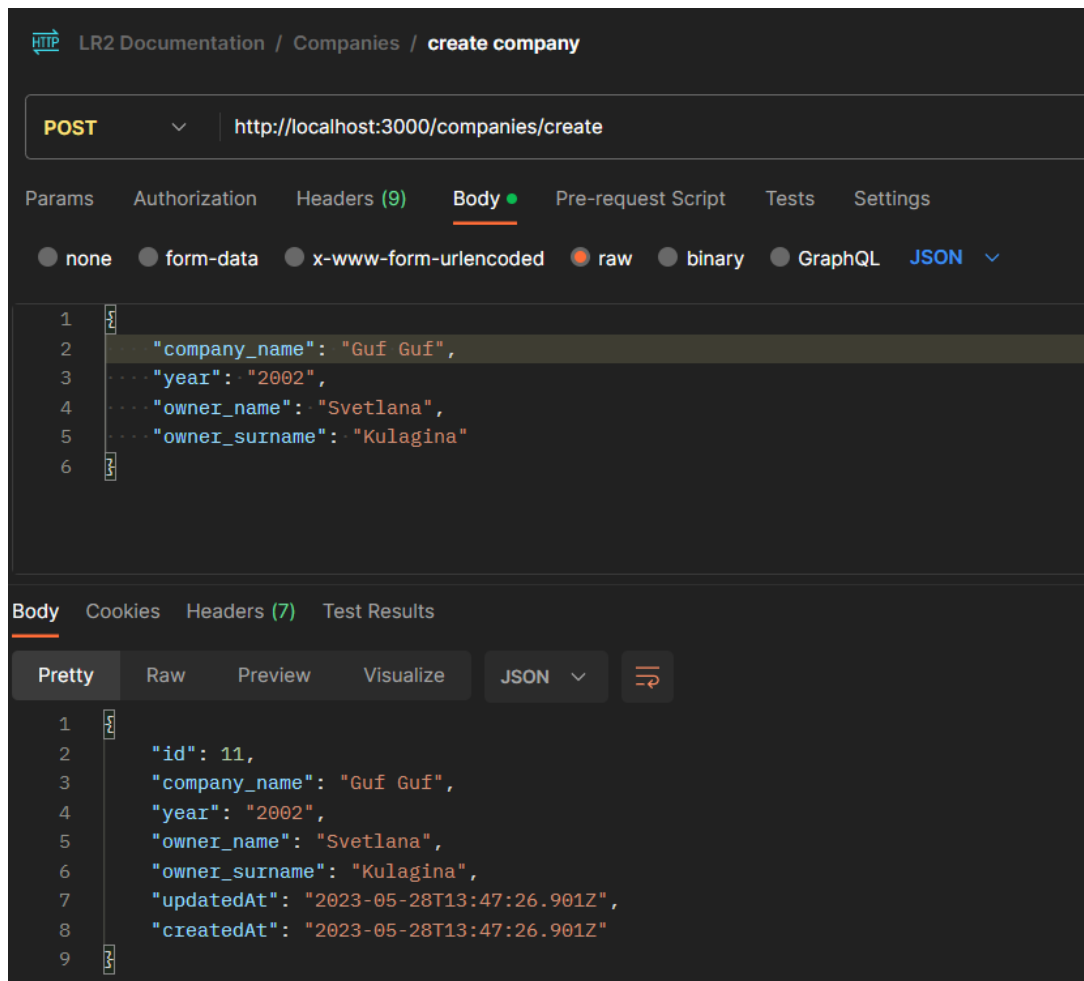
Микросервис **dogs_of_companies** содержит информацию о собаках, компаниях и связи между ними

Следующим шагом было прописать порты для каждого микросервиса и прописать файл, с помощью которого gateway будет перенаправлять запросы на них.

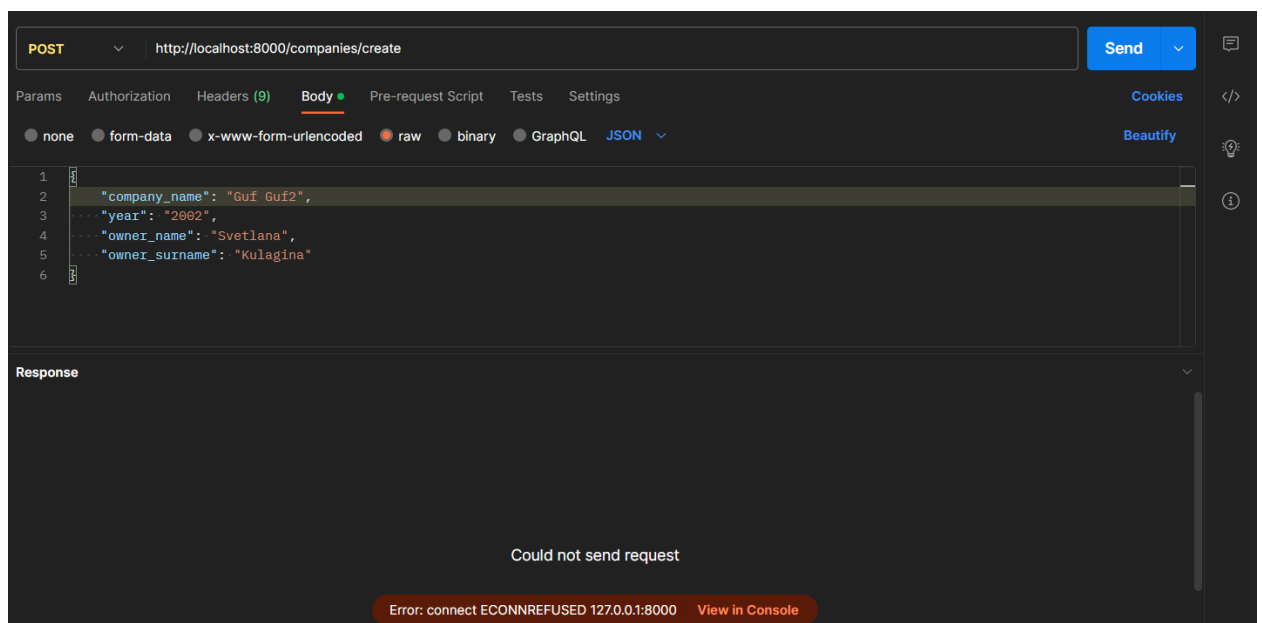
Пример:

```
app.all('/companies/*', async (req, res) => {
  const url = `http://localhost:9001${req.originalUrl}`;
  try {
    const response = await axios({
      method: req.method,
      url,
      data: req.body,
    });
    res.status(response.status).send(response.data);
  } catch (error) {
    if (error.response) {
      res.status(error.response.status).send(error.response.data);
    } else {
      res.status(500).send({'error': 'Internal Server Error'});
    }
  }
});
```

Сам gateway работает у меня на 3000, поэтому перейдем в Postman и отправим запрос на порт 3000, чтобы проверить, что у нас работает gateway



Теперь проверим, что наши микросервисы разделены, и при создании компании на порте для авторизации ничего не произойдет



ВЫВОД

В ходе лабораторной работы мной была создана микросервисная архитектура приложения с общей точкой gateway и протестирована