Дипломна робота

Розробка серверного функціоналу веб-застосунків на базі мікросервісної архітектури

Запорізький інститут економіки
та інформаційних технологій

Виконав
ст. гр. ІПЗ – 217 **Бєлоусов Д.В.**

Науковий керівник
доцент **Жеребцов О.А.**

Дипломна робота

Розробка серверного функціоналу веб-застосунків на базі мікросервісної архітектури

Запорожский институт экономики и информационных технологий

Мета роботи: розробка серверного фунціоналу на базі мікросервісної архітектури, дослідження переваг та недоліків її використання та порівняння з іншою архітектурою.

Дипломна робота

Розробка серверного функціоналу веб-застосунків на базі мікросервісної архітектури

Запорожский институт экономики и информационных технологий

Актуальність теми:

- активний розвиток web-технологій у наш час;
- розвиток та популярність фреймворку NestJS;
- коммерційна успішність подібних проектів;
- розвиток хмарних технологій;

Дипломна робота

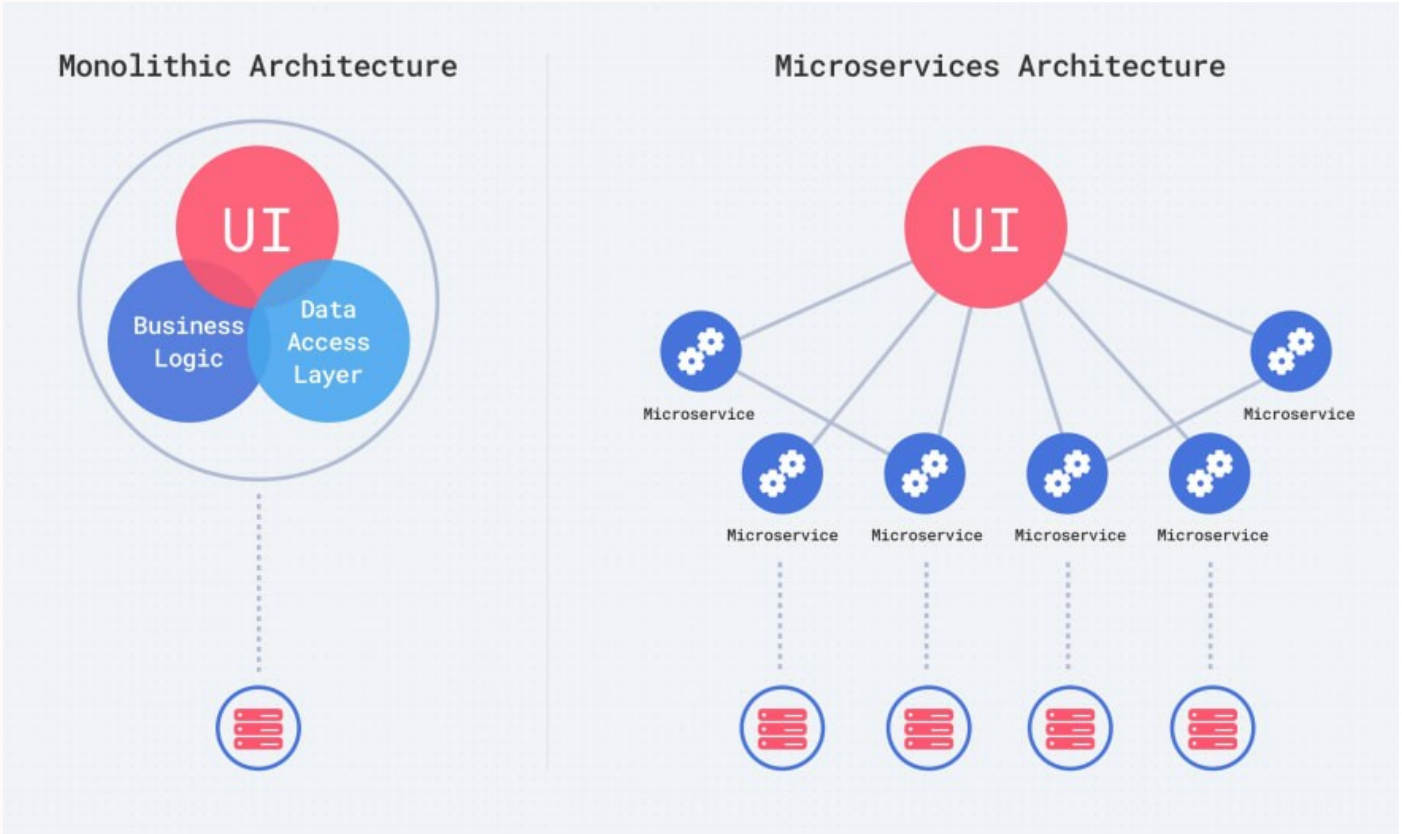
Розробка серверного функціоналу веб-застосунків на базі мікросервісної архітектури

Запорожский институт экономики и информационных технологий

Інструментарій для розробки:

- WebStorm;
- NestJS;
- TypeScript;
- Swagger;
- MongoDB.

Мікросервіси та моноліт



Переваги використання мікросервісів

1. Модульність
2. Надійність
3. Змінюваність
4. Масштабованість
5. Незалежна робота команд
6. Більше уваги до якості

Недоліки використання мікросервісів

1. Зростання витрат на інфраструктуру
2. Додатковий рівень комунікації
3. Середовище розробки стає складнішим

Фреймворк NestJS

Модуль мікросервіса

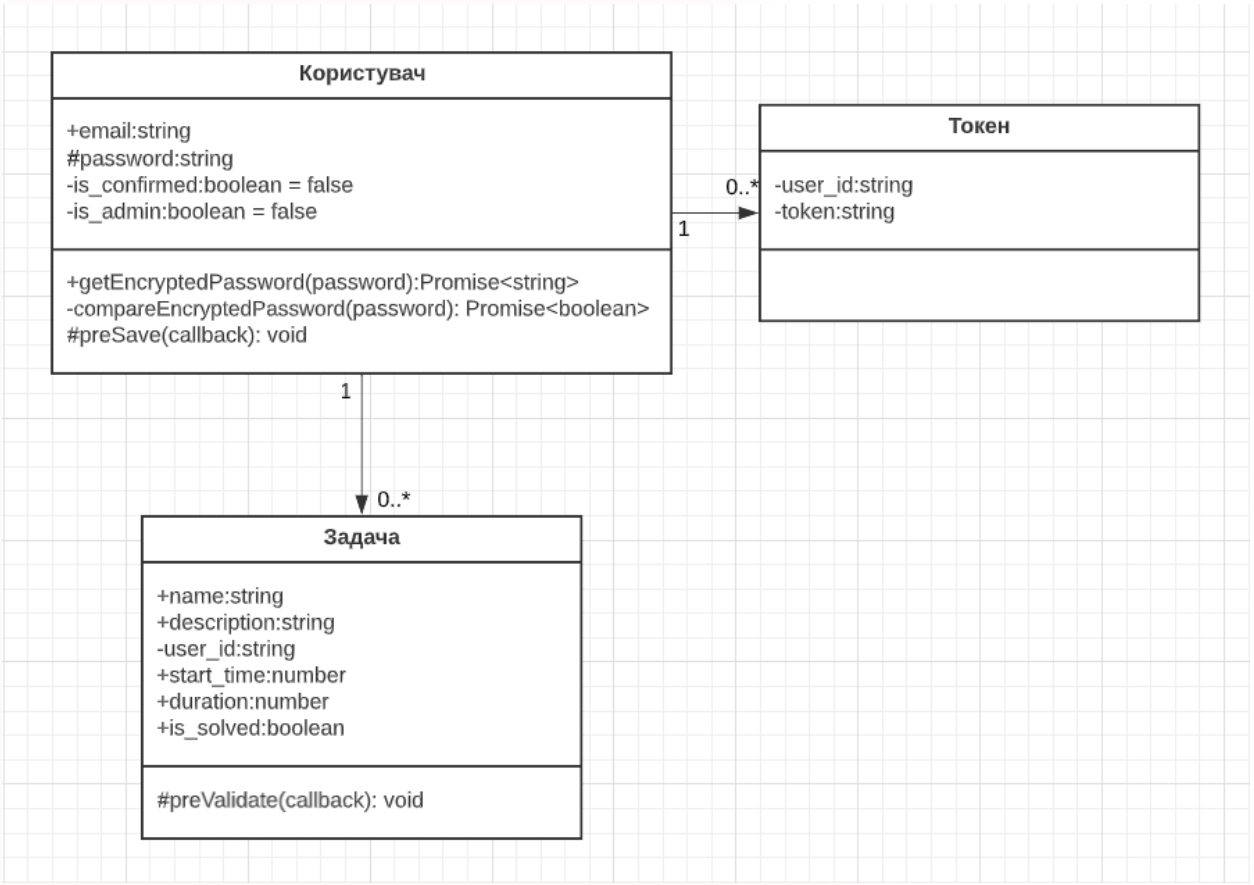
```
@Module( metadata: {
  imports: [
    MongooseModule.forRootAsync( options: {
      useClass: MongoConfigService,
    }),
    MongooseModule.forFeature( models: [
      {
        name: 'Task',
        schema: TaskSchema,
      },
    ]),
  ],
  controllers: [TaskController],
  providers: [TaskService],
})
export class TaskModule {}
```

Запуск мікросервіса

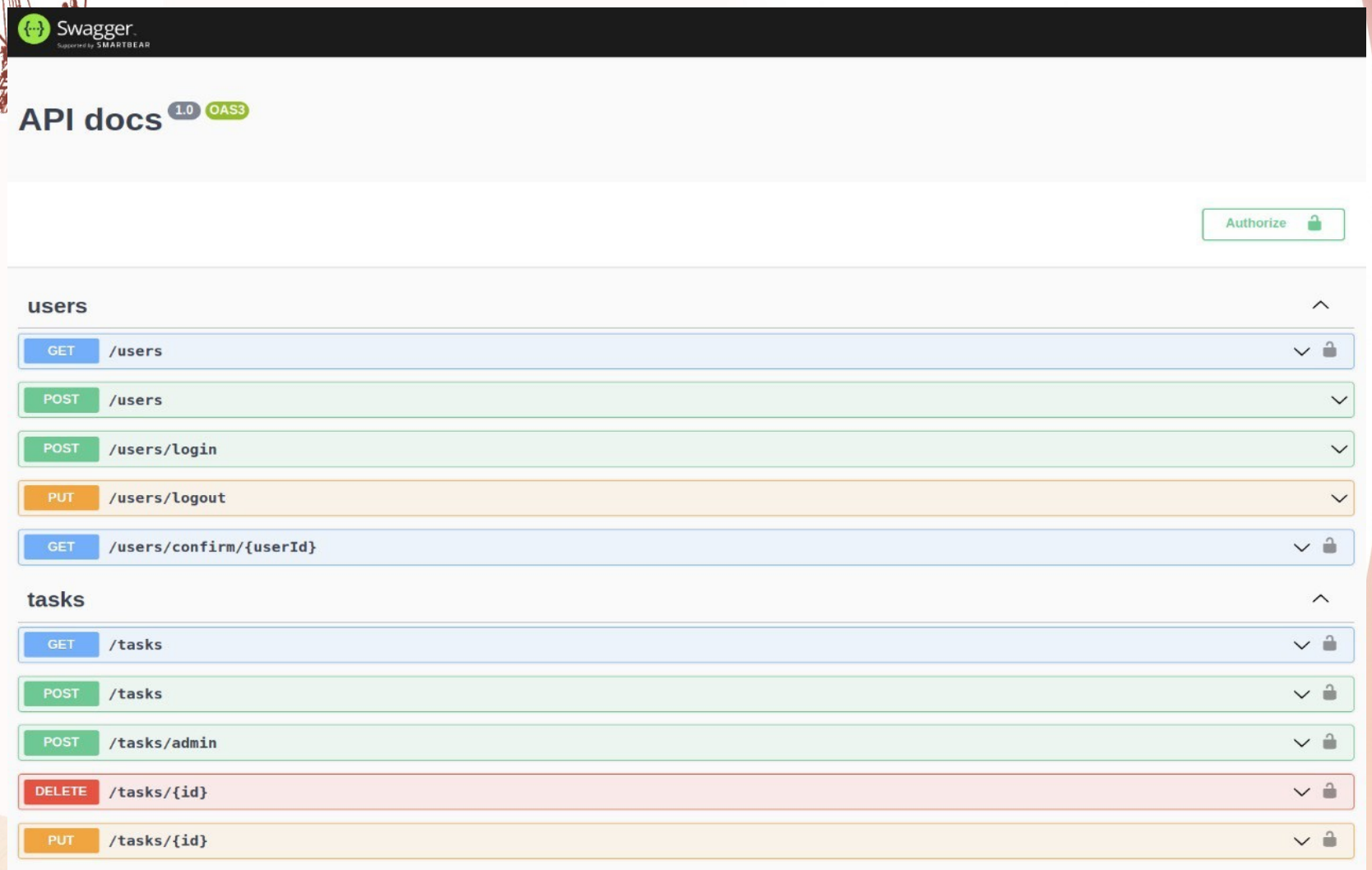
```
async function bootstrap() {
  const app = await NestFactory.createMicroservice(TaskModule, {
    transport: Transport.TCP,
    options: {
      host: 'localhost',
      port: 3000,
    },
  } as TcpOptions);

  await app.listen();
}
```

Схема бази даних

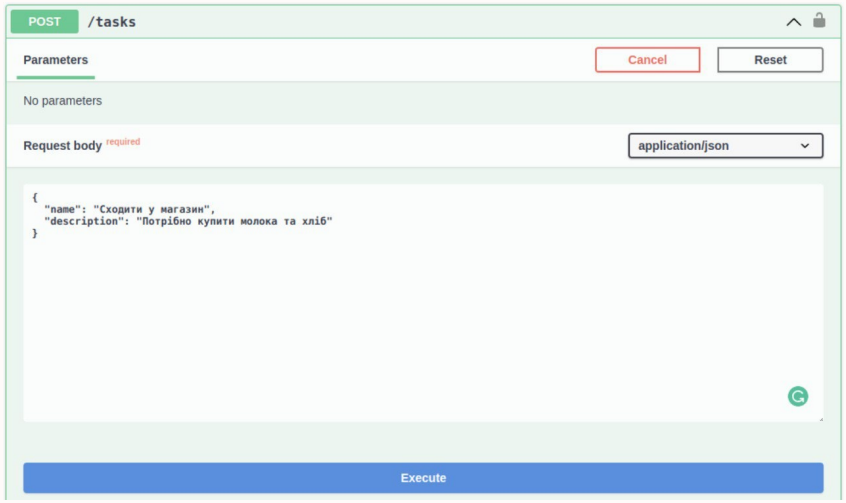


Swagger endpoints



Приклади запитів

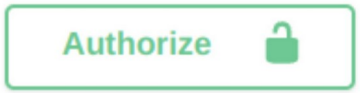
Створення задачі



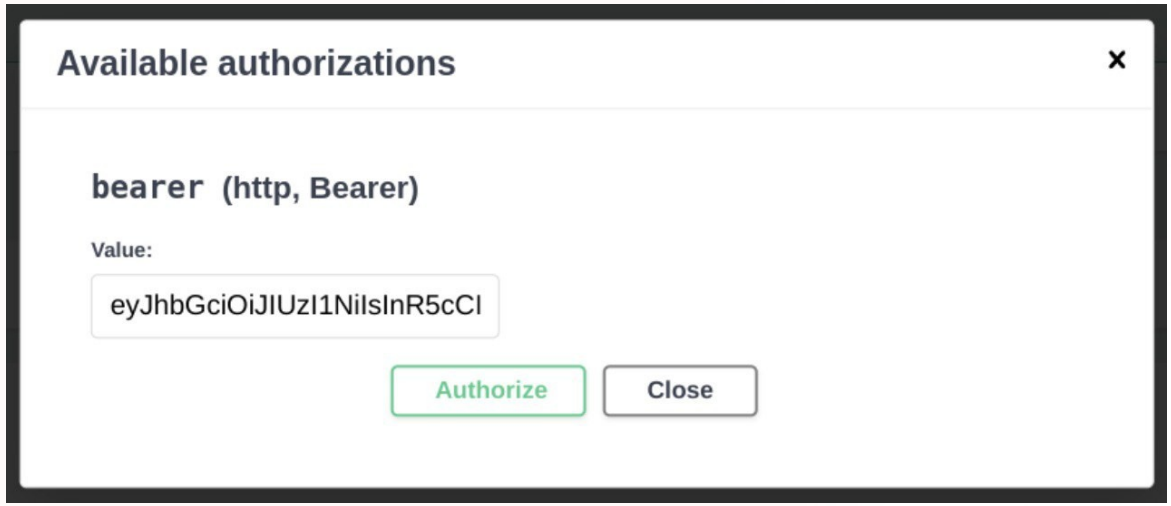
Конвертований запит Login

Авторизація

Кнопка Authorize



Вікно для вводу токена



Зв’язок між мікросервісами

Запит

Сервіс Swagger

Сервіс токенів

Сервіс користувачів

Сервіс доступів

Сервіс задач

Висновки

В час коли створення невеликих хмарних сервісів набирає популярність, веб-сервер на базі мікросервісів є вдалим архітектурним рішенням. Саму тому темою моєї роботи став веб-сервер на базі мікросервісної архітектури, який має функціонал:

- Створення та авторизація користувачів.
- Керування правами користувачів.
- Створення, редагування та видалення задач.
- Підтвердження користувачів адміністратором.

При розробці системи було використано фреймворк NestJS на мові програмування TypeScript для створення та під’єднання мікросервісів. Front-end сайту був створений за допомогою Swagger.

В ході роботи було розроблено чотири мікросервіси:

- Сервіс користувачів.
- Сервіс задач.
- Сервіс доступів.
- Сервіс токенів.

Дякую за увагу!