

Задание 04

1. Для одного сервиса управления данными (созданного в предыдущих лабораторных работах) создайте долговременное хранилище данных в noSQL базе данных MongoDB (4.0 или 5.0);
2. Выберите любой сервис, не связанный с клиентскими данными (клиентский сервис остается в PostgreSQL). Например, данные о поездках, данные о планах, данные о сообщениях
3. Должен быть создан скрипт по наполнению СУБД тестовыми значениями. Он должен запускаться при первом запуске вашего сервиса;
4. Для сущности, должны быть созданы запросы к БД (CRUD) согласно ранее разработанной архитектуре
5. Должны быть созданы индексы, ускоряющие запросы
6. Должно применяться индексирование по полям, по которым будет производиться поиск
7. При необходимости актуализируйте модель архитектуры в Structurizr DSL
8. Ваши сервисы должны запускаться через docker-compose командой `docker-compose up` (создайте Docker файлы для каждого сервиса)

Рекомендации по C++

- Используйте фреймворк Poco <https://docs.pocoproject.org/current/>
- Используйте mongo 4.0 (пока нет стабильного драйвера под более новые версии)
- Пример по работе с Poco Web Servers и Mongo https://github.com/DVDEmon/arch_lecture_examples/tree/main/hl_mai_lab_02

Рекомендации по Python:

- Используйте FastAPI для построения интерфейсов
- Рекомендуется использовать Pedantic для валидации моделей
- Используйте pymongo для работы с СУБД MongoDB
- Простой пример применения pymongo https://github.com/DVDEmon/architecture_python/tree/main/05_mongo