

План научной работы на 2 года

💡 Суть: Разработка и масштабирование микросервисной архитектуры на Java с использованием фреймворка Spring

Общий план технологий и структуры:

1. Курсовая работа (первый этап):

- Изучение основ микросервисной архитектуры и принципов ее организации
- Исследование фреймворка Spring и его роли в разработке микросервисов
- Разработка базового микросервиса с использованием Spring Boot и Spring Cloud
- Работа с микросервисами, обеспечивающими функциональность кэширования и работу с базой данных

2. Вторая курсовая работа (второй этап):

2. Вторая курсовая работа (второй этап):

- Исследование и выбор оптимальных технологий для кэширования данных в микросервисной архитектуре (например, Redis)
 - Разработка отдельных микросервисов, обеспечивающих функцию кэширования
 - Расширение микросервисной архитектуры с использованием принципов горизонтального масштабирования
 - Тестирование и оптимизация производительности разработанных микросервисов
-

3. Дипломная работа (третий этап):

- Исследование и интеграция фронт-энд фреймворка React в микросервисную архитектуру на Java
 - Разработка микросервисов, предоставляющих RESTful API для взаимодействия с фронт-эндом на React
 - Использование инструментов для мониторинга и логирования в разработанной архитектуре
 - Анализ и оптимизация производительности взаимодействия микросервисов и фронт-энда
-

Примерный системный дизайн такого концептуального проекта:

- Микросервисы разделены на отдельные модули, каждый из которых отвечает за конкретную функциональность
 - Взаимодействие между микросервисами осуществляется через RESTful API
 - Для кэширования данных используется Redis
 - В качестве базы данных может быть использована PostgreSQL, MongoDB или другая подходящая технология
 - Фронт-энд разработан на React и обращается к микросервисам через API
-

💡 Обратите внимание, что это лишь примерная формулировка и требуется дальнейшее исследование, конкретизация и разработка каждого этапа работы.

Перейдём к конкретизации:

Ниже представлены возможные темы с листами, в которых они разобраны подробнее.

- 1) Приложение для обмена блоками текста - 📄 [Pastebin app](#)
- 2) Приложение для уведомлений пользователей о чрезвычайных ситуациях - 📄 [Emergency Notification System](#)
- 3) Приложение Web - 📄 [Web Crawler](#)
- 4) Приложения для удобного обмена валютой в игре монополия, и хранения рейтинга игроков - [бэк](#), подключение к [бд](#), авторизация, статистика проведённых игр, перевод денег между игроками, [админский](#) интерфейс для контроля игры, интерфейс игрока, интерфейс общей информации процесса игры.

Рассмотрим план для варианта 2) Приложение для уведомлений пользователей о чрезвычайных ситуациях - скорее всего выберу его.

+ :: Первый этап: Курсовая работа

1. Разработка архитектуры и дизайна приложения.
2. Создание основных модулей и классов для обработки уведомлений.
3. Разработка интерфейса пользователя для создания паттерна рассылки уведомлений.
4. Реализация микросервиса для отправки уведомлений через Телеграм.
5. Реализация микросервиса для отправки уведомлений по электронной почте.
6. Интеграция разработанных микросервисов с основным приложением.

Первый этап: Курсовая работа

1. Разработка архитектуры и дизайна приложения.
2. Создание основных модулей и классов для обработки уведомлений.
3. Разработка интерфейса пользователя для создания паттерна рассылки уведомлений.
4. Реализация микросервиса для отправки уведомлений через Телеграм.
5. Реализация микросервиса для отправки уведомлений по электронной почте.
6. Интеграция разработанных микросервисов с основным приложением.
7. Тестирование приложения и проведение отладки ошибок.

Второй этап: Вторая курсовая работа

1. Добавление кэширования данных для улучшения производительности приложения.
2. Внедрение механизмов безопасности, включая авторизацию и аутентификацию пользователей.
3. Исследование и реализация механизмов контроля доступа для организаций и их подписчиков.
4. Улучшение интерфейса пользователя и его функциональности.
5. Расширение возможностей микросервисов, добавление новых методов отправки уведомлений.
6. Оптимизация кода и проведение рефакторинга.

Третий этап: Дипломный проект

1. Разработка мобильного приложения для уведомлений пользователей.
2. Интеграция мобильного приложения с основным приложением и базой данных.
3. Реализация пуш-уведомлений и интеграция с мобильными операционными системами.
4. Усиление безопасности мобильного приложения и защита данных пользователей.
5. Тестирование и отладка мобильного приложения.

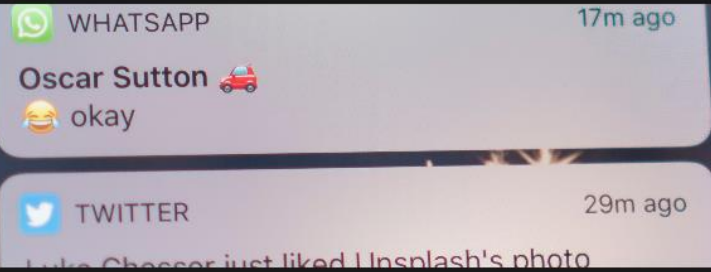
Второй этап: Вторая курсовая работа

1. Добавление кэширования данных для улучшения производительности приложения.
2. Внедрение механизмов безопасности, включая авторизацию и аутентификацию пользователей.
3. Исследование и реализация механизмов контроля доступа для организаций и их подписчиков.
4. Улучшение интерфейса пользователя и его функциональности.
5. Расширение возможностей микросервисов, добавление новых методов отправки уведомлений.
- + :: 6. Оптимизация кода и проведение рефакторинга.

Третий этап: Дипломный проект

1. Разработка мобильного приложения для уведомлений пользователей.
2. Интеграция мобильного приложения с основным приложением и базой данных.
3. Реализация пуш-уведомлений и интеграция с мобильными операционными системами.
4. Усиление безопасности мобильного приложения и защита данных пользователей.
5. Тестирование и отладка мобильного приложения.
6. Подготовка документации, написание отчета о дипломном проекте.

💡 Этот план обеспечивает постепенное развитие проекта от базовой функциональности до более сложных возможностей. Требуемые шаги и реализация могут отличаться в зависимости от конкретных потребностей и ресурсов.



Emergency Notification System

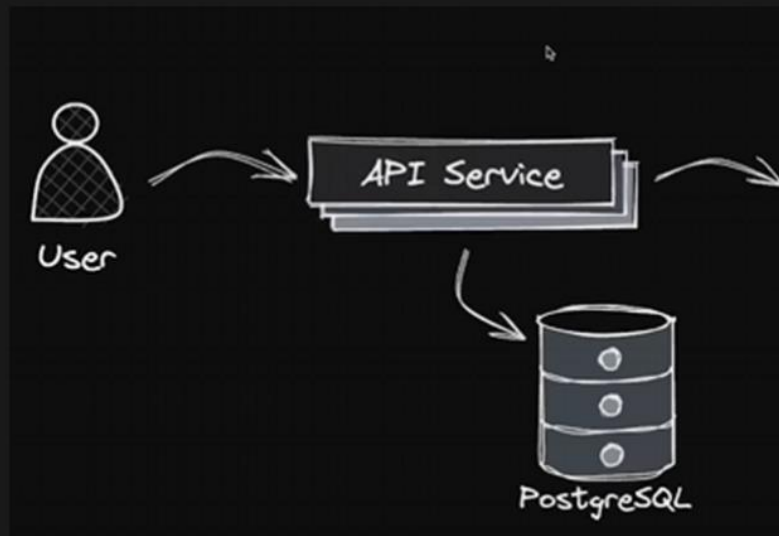
1 backlink

Упрощённый вариант - Вторая курсовая

ENS Фичи

- Загрузка пользователей с контактами через .csv или .xls файл
- Создание шаблонов уведомлений и конфигурация нужных получателей
- Отправка уведомлений на любой девайс получателя автоматически при отправке уведомления всей группе получателей

Press 'space' for AI, '/' for commands...



На этом месте в отличие от следующих вариантов будет не реализация пуш-уведомлений через телефон, а два микросервиса, для отправки уведомлений на почту и в телеграмм. И куда именно тебе будут приходить сообщения выбирается опционно в своём аккаунте.

+ В следующих реализациях я добавлю андроид приложение, как клиентский интерфейс, и через него уже будут приходить пуш-уведомления на телефон.

ENS Фичи



Загрузка пользователей с контактами через .csv или .xls файл

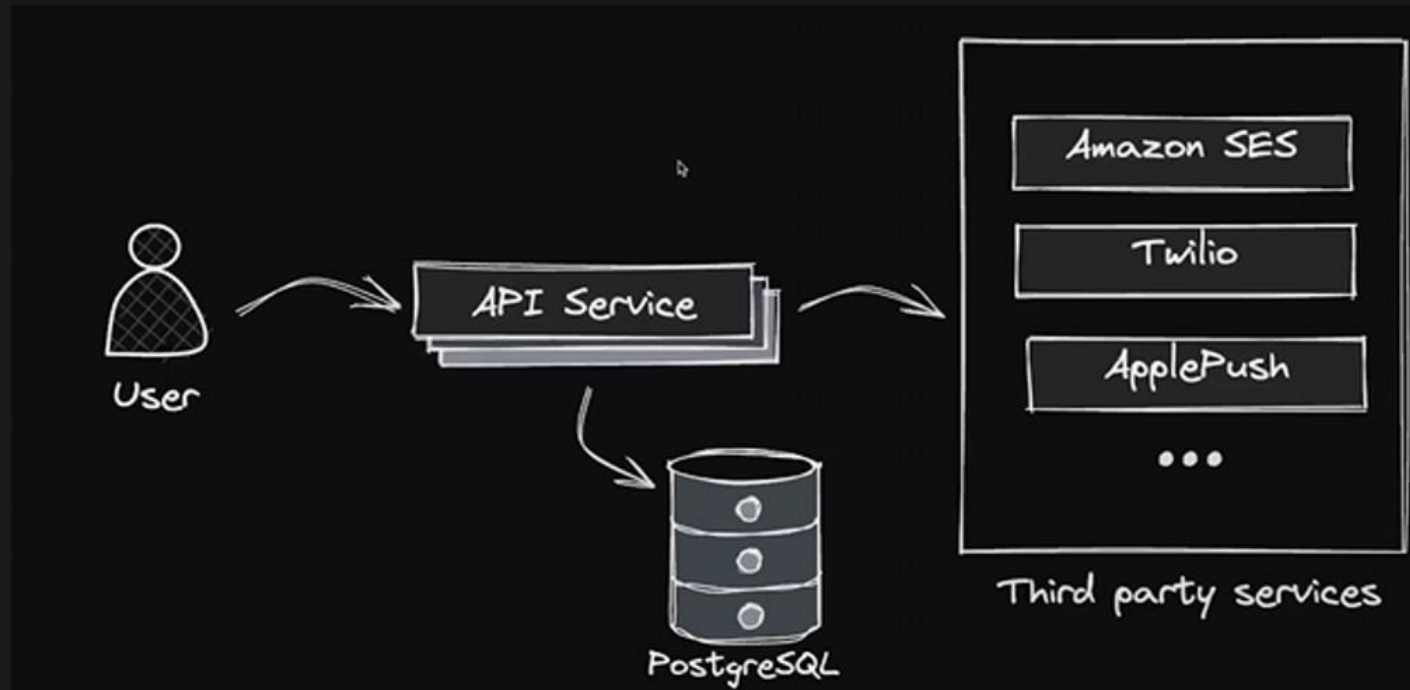


Создание шаблонов уведомлений и конфигурация нужных получателей



Отправка уведомлений на любой девайс получателя автоматически при отправке уведомления всей группе получателей

+ ::



Секретные требования



Нотификации могут отправляться на миллион человек



Сервер может упасть в процессе отправки нотификации



Все пользователи обязаны получить сообщение

+ ::

