Теория

“Исследование методов межсервисного взаимодействия для передачи данных большого объема.”

# АННОТАЦИЯ

**- Выпускная квалификационная работа содержит \_ страниц, \_ рисунков, \_ таблиц и \_ источников.**

**- Ключевые слова:** межсервисная передача данных, протоколы передачи данных, сервисная архитектура, архитектуры передачи данных, скорость передачи, данные большого объема.

**- Объектом исследования является**

**- Предметом исследования является** алгоритм сравнения передачи данных большого объема в сервисно-ориентированной архитектуре.

**- Целью работы является** исследовать и сравнить методы передачи данных большого объема в сервисно-ориентированной архитектуре. (практически выяснить положительные и отрицательные стороны различных подходов передачи данных большого объема)

**- Результатом работы является** практически доказанное сравнение передачи данных большого объема в сервисно-ориентированной архитектуре, обоснование возможности или невозможности, а также удобства использования каждого метода. (получить наилучший способ передачи, понять негативные стороны остальных методов)

# ОГЛАВЛЕНИЕ

**- Список определений и сокращений**

**- Введение**

**- Постановка задачи**

- Глава 1. Исследование способов передачи данных

- Что такое сервисы, виды архитектур?

- Про протоколы http 1, http 2

- Про FastApi

- Синхронная передача

- Асинхронная передача

- Почему не использую Uploadfile

- Про потоки (concurrent.futures)

- gRPC

- Веб сокеты

- Реализация

**- Результаты. Сравнение**

**- Вывод**

# СПИСОК ОПРЕДЕЛЕНИЙ И СОКРАЩЕНИЙ

В таблице 1 приведены определения и сокращения, используемые в данной работе.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# ВВЕДЕНИЕ

**- Актуальность**

**- Про подобные исследования. Сравнение с другими работами.**

**- В данной работе объектом исследования является**

**- Предметом исследования является**

**- Целью работы является**

**- Для достижения поставленной цели в данной работе решался ряд задач:**

1. **Задача 1**
2. **Задача 2**
3. **Задача 3**

**- При проведении исследований использовались следующие методы:**

1. **метод 1**
2. **метод 2**
3. **метод 3**

**- Научной новизной обладают следующие работы:**

1. **новизна 1**
2. **новизна 2**

**- Практическая значимость работы**

**- Основная часть магистерской диссертации состоит из \_ разделов:**

1. **раздел 1**
2. **раздел 2**
3. **раздел 3**

# ПОСТАНОВКА ЗАДАЧИ

- В данной работе необходимо

- В рамках выполнения данной работы были поставлены следующие задачи:

1. Изучение существующих подходов
2. Разработка
3. Проведение исследований

# ЗАКЛЮЧЕНИЕ

- Целью данной работы являлось

- Для достижения поставленной цели в работе был проведен

- Результаты исследований показывают, что

- Представленные результаты показывают, что все цели поставленные в данной работе были достигнуты.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

—-------------------

## Что такое Сервисы?

Различные приложения на различных узлах сети функционируют на разных аппаратно-программных платформах, и используют различные технологии и языки.

Чтобы связать все это и предоставить возможность одним приложениям обмениваться данными с другими, и были придуманы веб-сервисы.

По сути, веб-сервисы — это реализация абсолютно четких интерфейсов обмена данными между различными приложениями, которые написаны не только на разных языках, но и распределены на разных узлах сети.

Именно с появлением веб-сервисов развилась идея SOA — сервис-ориентированной архитектуры веб-приложений (Service Oriented Architecture).

### Преимущества связывания сервисов

https://pkf-alena.ru/kak-svyazat-servisy-mezdu-soboi/

Модульность: Связывание сервисов позволяет разделять сложные системы на отдельные модули, каждый из которых отвечает за конкретную функциональность. Это упрощает понимание и поддержку кода, позволяет повторно использовать уже написанный функционал и легко вносить изменения, не затрагивая другие модули.

Расширяемость: Связывание сервисов позволяет легко добавлять новые сервисы и компоненты в систему. Новые модули могут быть добавлены и интегрированы существующей системой без необходимости внесения значительных изменений в код.

Гибкость: Связывание сервисов позволяет конструировать системы с помощью различных комбинаций сервисов, что позволяет настраивать систему под конкретные требования и условия эксплуатации. Можно использовать только те сервисы, которые нужны, и не тратить ресурсы на ненужный функционал.

Упрощение тестирования: Связывание сервисов позволяет проводить модульное тестирование отдельных компонентов системы независимо от других. Это делает процесс тестирования более прозрачным и уменьшает вероятность появления ошибок из-за взаимодействия модулей.

Масштабируемость: Связывание сервисов позволяет легко масштабировать систему по мере роста нагрузки. Каждый сервис может быть масштабирован независимо от других, что позволяет эффективно использовать ресурсы и обеспечивать высокую производительность системы.

Использование правильных методов и инструментов для связывания сервисов может существенно улучшить качество и эффективность разработки ПО, а также обеспечить стабильную и масштабируемую работу системы в долгосрочной перспективе.