



Протоколы передачи данных, очереди и REST

HTTP, SOAP, AMQP, gRPC



АКАДЕМИЯ

01.

HTTP и SOAP

02.

AMQP

03.

gRPC

04.

RabbitMQ и ActiveMQ



АКАДЕМИЯ



HTTP и SOAP

Протоколы корпоративного предприятия



АКАДЕМИЯ

HTTP и SOAP

SOAP

- Протокол обмена данными с стиле RPC
- Базируется на XML
- Является развитием протокола XML-RPC
- Последняя версия протокола вышла в 2007 как рекомендация

WSDL

- Язык описания веб-сервисов и доступа к ним
- Базируется на XML
- Последняя версия спецификации вышла в 2007

Причины использовать и альтернативы

- Нет
- Приемлемые альтернативы: Swagger(OpenAPI), gRPC, Thrift, JSON RPC

Связка технологий SOAP-WSDL безнадежно устарела,

на старых проектах по возможности
переходить на более современные
технологии, на новых сразу
использовать современные



gRPC

Элегантная простота



АКАДЕМИЯ

gRPC

gRPC

- Современный протокол, первоначально разработанный Google в 2015
- Использует HTTP/2 в качестве транспорта
- Использует для описания сервисов IDL Protocol Buffers

Protocol Buffers

- Технология сериализации структурированных данных
- Включает в себя IDL для описания структур и методов

Причины использовать и альтернативы

- Удобнее традиционных Swagger API тем, что имеет более строгое описание
- Работает гораздо быстрее SOAP, в том числе из-за ускорения операций сериализации
- Приемлемые альтернативы: Apache Thrift, JSON RPC

gRPC – современный и высокопроизводительный протокол,

использование gRPC жизненно необходимо в системах с микросервисной архитектурой, и любых других, где сетевые задержки имеют большое влияние



AMQP

События, сообщения,
независимость и свобода



АКАДЕМИЯ

AMQP

Стандарт обмена сообщениями

- Реализует архитектурный паттерн Publisher-Subscriber
- Отделяет источник сообщений от потребителя
- Независимость частей системы друг от друга позволяет улучшить расширяемость и масштабируемость

AMQP

- Определяет формат сообщений и правила маршрутизации
- Предоставляет множество вариантов маршрутизации:
 - Прямая отсылка(Direct)
 - Веерная рассылка(Fanout)
 - Топики(Topic)
- Поддерживается транзакционность(внутри сервера)
- Поддерживается очередность доставки

AMQP

Основные понятия

- **Сообщение(Message)** – пакет передаваемых данных, с телом произвольного формата
- **Продюсер(Producer)** – компонент, генерирующий и рассылающий события
- **Потребитель или Получатель(Consumer)** – компонент, получающий события
- **Брокер(Broker)** – система, доставляющая сообщения от продюсера к получателю
- **Точка обмена (Exchange)** – объект, в который отправляются сообщения продюсером
- **Очередь (Queue)** — объект, который хранит входящие сообщения, до момента их получения потребителем

AMQP – наиболее общепринятый протокол для систем, основанных на сообщениях,

использование AMQP позволяет строить системы с элементами Event-Driven Architecture, позволяет расширять и масштабировать систему, но вместе с тем усложняет взаимодействие из-за природы асинхронности



RabbitMQ и ActiveMQ

Маленький шаг для программиста,
огромный скачок в
Event-Driven Architecture



АКАДЕМИЯ

1

ActiveMQ

- Написан на Java
- Поддерживает JMS 1.1
- Может работать как с выделенным брокером, так и точка-точка
- Прост в использовании
- Поддерживает кластеризацию

2

RabbitMQ

- Написан на Erlang
- Работает с выделенным брокером
- Очень прост в установке и использовании
- Высокая производительность и надежность

3

ZeroMQ

- Написан на C++
- Очень высокопроизводительный
- Работает без выделенного брокера
- Сложно использовать в сколь-нибудь реальных сценариях

Всем спасибо!

Лига –
**лучший
старт**
карьеры!

Мы в Лиге!

Умножай
знания –
верь в мечту!

Каждый
день – новый
челлендж!



Андрей Лабазин

Разработчик



АКАДЕМИЯ