



OTUS

# Онлайн образование

[otus.ru](http://otus.ru)

• REC

Проверить, идет ли запись

Меня хорошо видно  
&& слышно?



Тема вебинара

# Confluent REST API



**Чащина Александра**

Big Data Engineer



# Правила вебинара



Активно  
участвуем



Off-topic обсуждаем  
в Телеграм



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое  
на активность



Пишем в чат



Говорим голосом



Документ

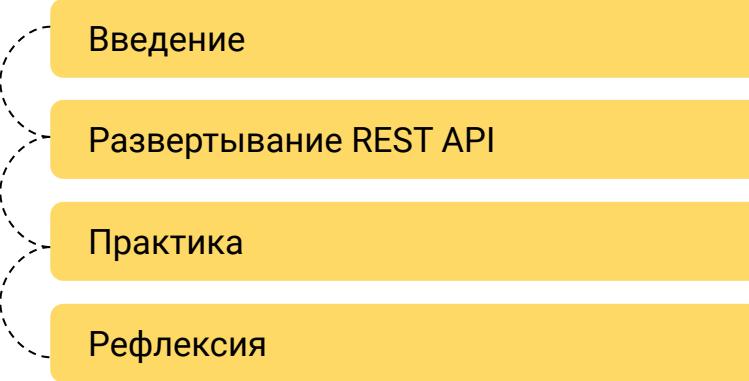


Ответьте себе или  
задайте вопрос

# Карта курса



# Маршрут вебинара



# Цели вебинара

После занятия вы сможете

1. Разворачивать Confluent REST API локально
  2. Использовать Confluent REST API
- 
-

# Смысл

Зачем вам это уметь

- 
1. Confluent REST API - удобный способ пользоваться и управлять кластером Kafka

# Введение

# Confluent REST API

- RESTful интерфейс кластера Kafka:

# Confluent REST API

- RESTful интерфейс кластера Kafka:
  - Запись и чтение сообщений

# Confluent REST API

- RESTful интерфейс кластера Kafka:
  - Запись и чтение сообщений
  - Мониторинг кластера

# Confluent REST API

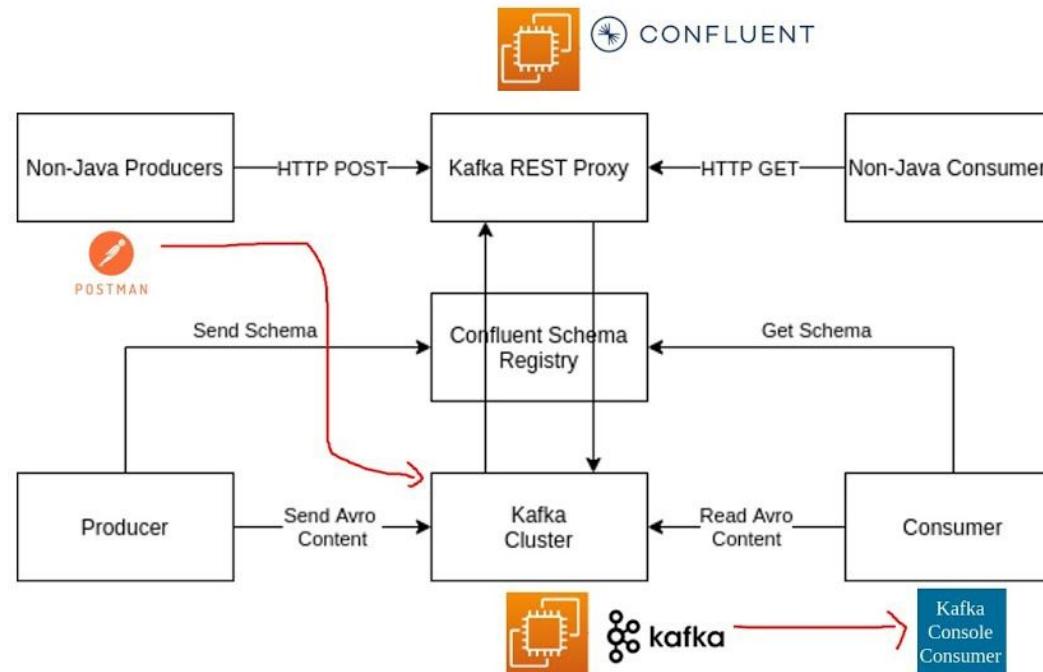
- RESTful интерфейс кластера Kafka:
  - Запись и чтение сообщений
  - Мониторинг кластера
  - Администрирование кластера

# Confluent REST API

- RESTful интерфейс кластера Kafka:
  - Запись и чтение сообщений
  - Мониторинг кластера
  - Администрирование кластера

 Нет необходимости использовать нативные протоколы или клиентов Kafka

# Confluent REST API - кейсы использования



# Confluent REST API - основные функции

- **Метаданные** - состояние брокера, топиков, партиций, конфигурации

# Confluent REST API - основные функции

- **Метаданные** - состояние брокера, топиков, партиций, конфигурации
- **Producers** - отправка сообщений через API
  - Объекты producer создаются на стороне кластера и используются совместно всеми запросами через API. Поэтому их нельзя тонко настраивать.

# Confluent REST API - основные функции

- **Метаданные** - состояние брокера, топиков, партиций, конфигурации
- **Producers** - отправка сообщений через API
  - Объекты producer создаются на стороне кластера и используются совместно всеми запросами через API. Поэтому их нельзя тонко настраивать.
- **Consumers** - чтение сообщений через API
  - Объект consumer создается под каждый запрос.



# Confluent REST API - основные функции

- **Метаданные** - состояние брокера, топиков, партиций, конфигурации
- **Producers** - отправка сообщений через API
  - Объекты producer создаются на стороне кластера и используются совместно всеми запросами через API. Поэтому их нельзя тонко настраивать.
- **Consumers** - чтение сообщений через API
  - Объект consumer создается под каждый запрос.
  - Используют ресурсы кластера



# Confluent REST API - основные функции

- **Метаданные** - состояние брокера, топиков, партиций, конфигурации
- **Producers** - отправка сообщений через API
  - Объекты producer создаются на стороне кластера и используются совместно всеми запросами через API. Поэтому их нельзя тонко настраивать.
- **Consumers** - чтение сообщений через API
  - Объект consumer создается на каждый инстанс Proxy
  - Используют ресурсы кластера
  - Ограниченнaя конфигурация



# Confluent REST API - основные функции

- **Метаданные** - состояние брокера, топиков, партиций, конфигурации
- **Producers** - отправка сообщений через API
  - Объекты producer создаются на стороне кластера и используются совместно всеми запросами через API. Поэтому их нельзя тонко настраивать.
- **Consumers** - чтение сообщений через API
  - Объект consumer создается на каждый инстанс Proxy
  - Используют ресурсы кластера
  - Ограниченнная конфигурация
- **Форматы данных:**
  - JSON

# Confluent REST API - основные функции

- **Метаданные** - состояние брокера, топиков, партиций, конфигурации
- **Producers** - отправка сообщений через API
  - Объекты producer создаются на стороне кластера и используются совместно всеми запросами через API. Поэтому их нельзя тонко настраивать.
- **Consumers** - чтение сообщений через API
  - Объект consumer создается на каждый инстанс Proxy
  - Используют ресурсы кластера
  - Ограниченнная конфигурация
- **Форматы данных:**
  - JSON
  - Бинарный формат (base64)



# Confluent REST API - основные функции

- **Метаданные** - состояние брокера, топиков, партиций, конфигурации
- **Producers** - отправка сообщений через API
  - Объекты producer создаются на стороне кластера и используются совместно всеми запросами через API. Поэтому их нельзя тонко настраивать.
- **Consumers** - чтение сообщений через API
  - Объект consumer создается на каждый инстанс Proxy
  - Используют ресурсы кластера
  - Ограниченнная конфигурация
- **Форматы данных:**
  - JSON
  - Бинарный формат (base64)
  - Данные со схемами в формате JSON, Avro или Protobuf (используется SchemaRegistry)



# Confluent REST API - ограничения

- Очень ограниченные настройки **Consumer** и **Producer** (конфигурируются на стороне сервера REST)

# Confluent REST API - ограничения

- Очень ограниченные настройки **Consumer** и **Producer** (конфигурируются на стороне сервера REST)
- В рамках одного запроса сообщения можно писать только в один топик

# Confluent REST API - ограничения

- Очень ограниченные настройки **Consumer** и **Producer** (конфигурируются на стороне сервера REST)
- В рамках одного запроса сообщения можно писать только в один топик
- Нельзя использовать разные сериализаторы для ключа и сообщения.

# Развертывание REST API

# Рекомендации к оборудованию

- **Оперативная память.**
  - Запись и администрирование - стабильный объем памяти (~1GB)

# Рекомендации к оборудованию

- **Оперативная память.**
  - Запись и администрирование - стабильный объем памяти (~1GB)
  - Чтение является stateful-операций, поэтому объем используемой памяти пропорционален предполагаемому количеству запросов чтения (~16 MB на Consumer)



# Рекомендации к оборудованию

- **Оперативная память.**
  - Запись и администрирование - стабильный объем памяти (~1GB)
  - Чтение является stateful-операций, поэтому объем используемой памяти пропорционален предполагаемому количеству запросов чтения (~16 MB на Consumer)
- **Процессор.** Затратные операции - сжатие и сериализация



# Рекомендации к оборудованию

- **Оперативная память.**
  - Запись и администрирование - стабильный объем памяти (~1GB)
  - Чтение является stateful-операций, поэтому объем используемой памяти пропорционален предполагаемому количеству запросов чтения (~16 MB на Consumer)
- **Процессор.** Затратные операции - сжатие и сериализация
- **Диск** не используется



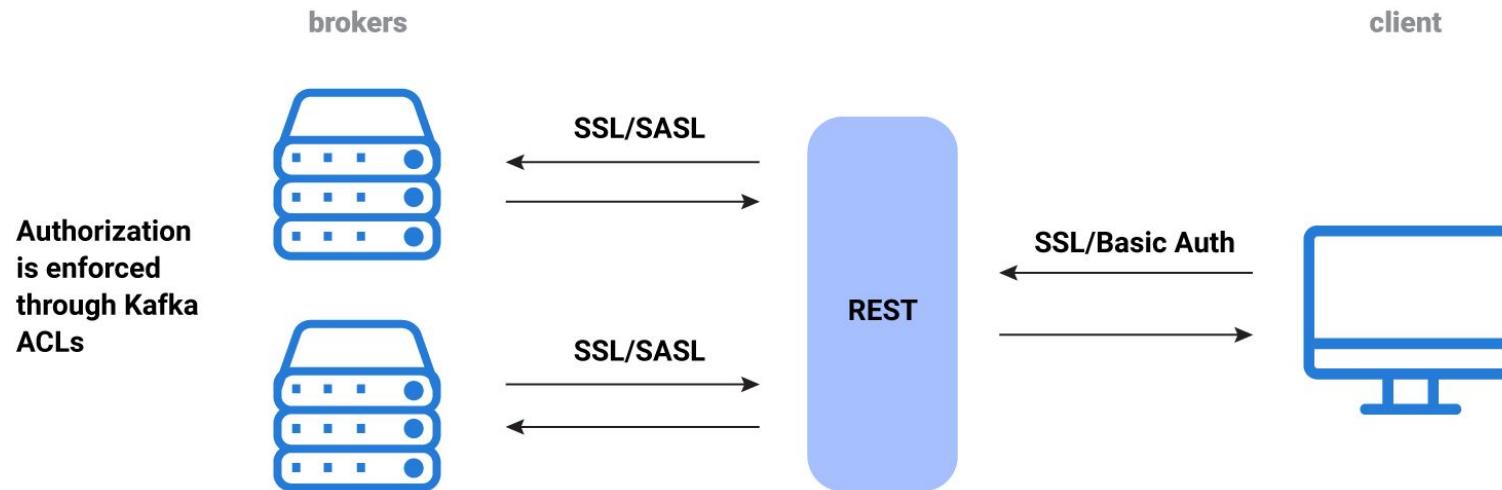
# Рекомендации к оборудованию

- **Оперативная память.**
  - Запись и администрирование - стабильный объем памяти (~1GB)
  - Чтение является stateful-операций, поэтому объем используемой памяти пропорционален предполагаемому количеству запросов чтения (~16 MB на Consumer)
- **Процессор.** Затратные операции - сжатие и сериализация
- **Диск** не используется
- **Сеть.** Лучше размещать в том же дата центре, что и сам кластер Kafka

[Документация](#)

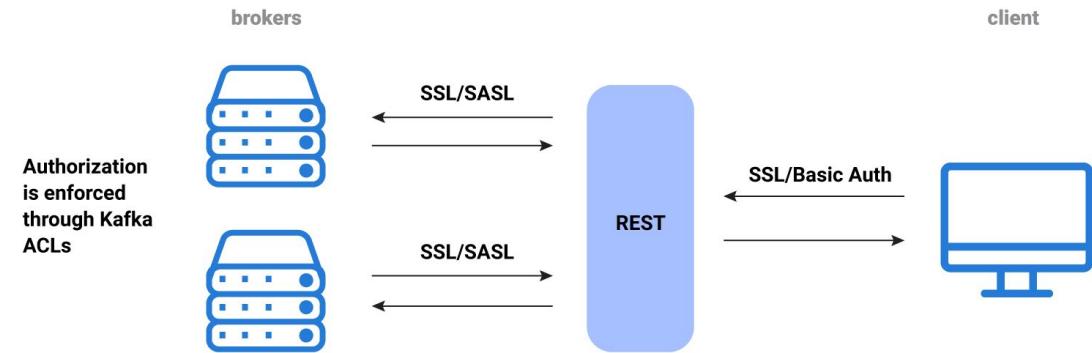


# Авторизация



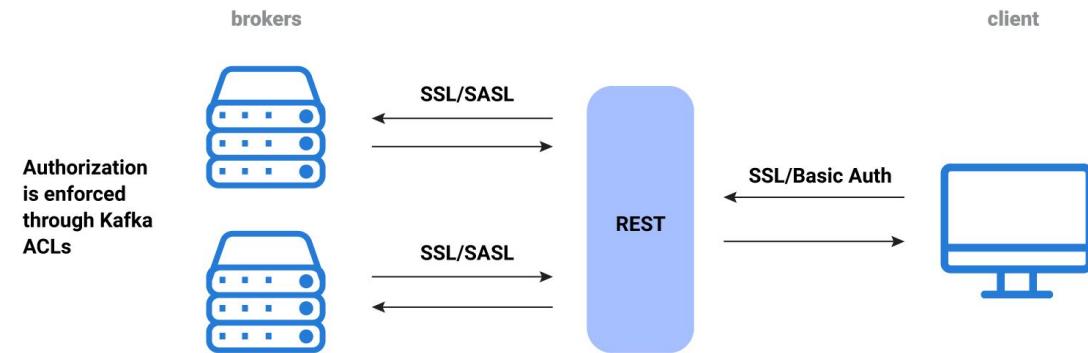
# Авторизация

- **HTTP Basic Authentication** между клиентом и REST, настраивается в конфигурационных файлах REST



# Авторизация

- **HTTP Basic Authentication** между клиентом и REST, настраивается в конфигурационных файлах REST
- **SASL Authentication** между REST и брокером (как стандартный Java-клиент)



# Вопросы?



Ставим “+”,  
если вопросы есть



Ставим “-”,  
если вопросов нет

# Практика

# Тезисы

## Подведем итоги

1. Confluent REST API позволяет пользоваться и управлять кластером Kafka без специальных клиентов
2. Остается ограниченным решением по сравнению со стандартными клиентами



# Рефлексия

# Цели вебинара

После занятия вы сможете

1. Разворачивать Confluent REST API локально
  2. Использовать Confluent REST API
- 
-

# Рефлексия



С какими основными мыслями  
и инсайтами уходите с вебинара?



Как будете применять на практике то,  
что узнали на вебинаре?

# Следующий вебинар



**ksqlDB**

✓ Ссылка на вебинар  
будет в ЛК за 15 минут

✓ Материалы  
к занятию в ЛК –  
можно изучать

bookmark-red ✓ Обязательный материал  
обозначен красной  
лентой

**Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате**

Спасибо за внимание!

# Приходите на следующие вебинары



**Чащина Александра**

Big Data Engineer

