
Документация сервиса Kafka2Rest

Выпуск

ROBIN RPA Team

дек. 25, 2020

Содержание:

1	Kafka2Rest	2
1.1	Оглавление	2



Проксирующий сервис между Apache Kafka, позволяющий выполнять REST-запросы для получения/выгрузки информации из Кафки.

1.1 Оглавление

- *Installation*
- *Launch*
- *Configuration*
- *Logging*
- *Authorization*
- *Requests*
- *Notes*

1.1.1 Установка

Для использования сервиса на линуксовой машине, необходимо установить следующие зависимости:

```
sudo apt install librdkafka-dev
```

На линуксовой машине выполняем следующую команду.

```
pip install aiohttp pyyaml confluent-kafka
```

Для Windows придётся собрать и подгрузить `librdkafka`.

После подготовки всех зависимостей и библиотек, клонируем репозиторий

```
git clone https://git-server/research/kafka-2-rest.git -b confluent
```

И перемещаемся в каталог с репозиторием

```
cd kafka-2-rest
```

1.1.2 Запуск

Сервис можно запустить как просто

```
sudo python3 server.py
```

Так и в фоновом режиме

```
sudo python3 server.py &
```

1.1.3 Конфигурация

В каталоге с сервером находится файл `config.yml` со следующим содержимым:

```
server:
  host: 0.0.0.0      # Адрес, на котором будет запущен сервис
  port: 5443        # Порт, на котором будет запущен сервис
kafka:
  default_group_id: 12345 # Group Id, использующийся по умолчанию. Может быть любым, однако ↪
  ↪ не рекомендуется использовать в его названии пробелы и спецсимволы. Системный параметр.
  bootstrap_servers:
    - localhost:9092 # Здесь может быть несколько ваших серверов с уже запущенной кафкой
    - 172.28.1.119:9292
  consumer_options:
    auto_offset_reset: earliest
    consumer_timeout_ms: 1000
    max_poll_records: 1
  # enable_auto_commit: true
  producer_options:
    retries: 5
```

1.1.4 Логгирование

В сервисе предусмотрено логгирование двух видов - в консоль и в файл с логами, который располагается в той же директории и называется `kafka2rest.log`

1.1.5 Авторизация

Для использования сервиса необходимо авторизоваться.

В Postman во вкладке **Authorization** выбираем тип авторизации **Basic Auth** и прописываем ваш логин и пароль для доступа к сервису.

Можно сгенерировать токен **Basic Auth** следующим образом:

Пару `mylogin:mypassword` кодируем в `base64`

И при всех дальнейших запросах используем следующий заголовок

`Authorization: Basic закодированная_пара_в_base64`

1.1.6 Запросы

В сервисе предусмотрены следующие запросы:

- **GET** `https://193.232.107.172/status/test_topic/`
Получает все непрочитанные сообщения из топика `test_topic`.
- **POST** `https://193.232.107.172/execute/`

Body:

```
{
  "topic": "test_topic",
  "data": {
    // Здесь могут быть любые json-объекты, которые вам нужно передать
    "hello": "world",
    "answer": 42,
    "my_awesome_list": [],
    "my_awesome_inner_object": {}
  }
}
```

Отправляет сообщение в топик `test_topic`.

1.1.7 Условия

В POST-запросе тело сообщения **должно быть строго в формате JSON** и содержать в себе атрибут `topic`, с заданным топиком для отправки.