

Avaliação 02: Arquitetura Orientada a Eventos

Daiaarah Kalil e Lucas de Lima

Nosso trabalho é um sistema de notícias de saúde, finanças e esportes. Foram feitos 3 consumidores e 2 produtores Kafka, e para isso utilizamos o `Consumer` e o `Producer` da biblioteca `confluent_kafka`.

O dicionário `config` foi criado tanto no consumidor quanto no produtor, e ele armazena as configurações de endereço do servidor Kafka, ID do grupo de consumidores e o ponto de partida da leitura de mensagens (no nosso caso é a partir do início dos tópicos) no **consumidor**, e as configurações de endereço do servidor Kafka no **produtor**.

Os consumidores foram projetados para se inscrever em diferentes tópicos: o consumidor 1 consome os tópicos “saúde e finanças” do produtor 1, o consumidor 2 consome o tópico “esportes” do produtor 2, e o consumidor 3 consome o tópico “saúde” do produtor 1.

CONSUMIDORES:

Primeiramente criamos uma instância do consumidor Kafka que chamamos de `c`, e ela utiliza as informações que estão em `config`. Pelo método `subscribe()` do Kafka, cada consumidor é inscrito nos seus respectivos tópicos.

A função `main` contém um loop infinito para que fique chamando continuamente o método `poll()`. Esse é um método do consumidor Kafka que obtém novas mensagens, e o valor passado como parâmetro é o tempo limite, em segundos, para aguardar mensagens. Caso não receba mensagens neste período de tempo, o método `close()` encerra a conexão do consumidor.

A cada mensagem recebida é feita a decodificação pelo método `decode()` no padrão `UTF-8`, e logo após a mesma é impressa no terminal.

PRODUTORES:

Inicialmente foi criada uma instância do consumidor Kafka que chamamos de `p`, e ela utiliza as informações que estão em `config`.

Em seguida foram criados os respectivos tópicos, e suas listas contendo 3 manchetes cada.

A função `main` contém um loop infinito, e em cada iteração o método `produce()` é chamado para cada um dos tópicos. Os parâmetros deste método são dois: o nome do tópico, e a mensagem a ser enviada (que é escolhida aleatoriamente entre as manchetes da lista do respectivo tópico e codificada no padrão `UTF-8`).

Após a mensagem ser produzida chama-se o método `flush()` para garantir que esta seja enviada imediatamente para o Kafka. Utilizamos também o método `sleep()` para pausar o loop e facilitar a visualização das mensagens que estão sendo enviadas.

Referências

- <https://towardsdatascience.com/how-to-build-a-simple-kafka-producer-and-consumer-with-python-a967769c4742>
- <https://kafka-python.readthedocs.io/en/master/apidoc/KafkaConsumer.html>
- <https://kafka-python.readthedocs.io/en/master/apidoc/KafkaProducer.html>