

Kafka Streams Assignment

Description

You are a software engineer in an IoT company. There are thousands of IoT devices that send frequent data which are stored in a database. Your task is to create a Kafka stream processor which saves results to the database.

Requirements

a) Create a Kafka streams application

- reads from topics devices, coffees, coffeeMachines and customers
- produces information about coffeemachine serial - device serial- coffee dosage - coffee milligrams - timestamp - group id - customer name and push it into a topic with name coffeeInformation
- calculates single vs double (1 dosage vs 2 dosages) per customer and push it into a topic with name singleVsDouble
- calculates group usage percentage per coffee machine (group is a metal permanent attachment that brings water out of the machine and into the filter basket - where the coffee is brewed) and push it into a topic with name usagePerGroup
- All serialisation formats should be AVRO based and belong to iot schema

device		
Name	Type	Validation
id	string	
serial	string	
longitude	double	-180 - 180
latitude	double	-90 - 90
coffeeMachineld	string	

coffeeMachine		
Name	Type	Validation
id	string	
serial	string	
customerId	string	
groupCount	int	1-4

coffee		
Name	Type	Validation
Id	string	
timestamp	datetime	timestamp
groupId	int	1-4
coffeeMachineId	string	
customerId	string	
milligrams	int	
dosage	int	1-3

customer		
Name	Type	Validation
id	string	
name	string	
longitude	double	-180 - 180
latitude	double	-90 - 90

- b) Use Kafka connect to save produced topics on DB (preferably PostgreSQL).
- c) A jar application is provided that will stream fixed data on devices, coffeeMachines and customers and stream random data on coffees every 20 seconds every time it runs. As arguments expect schema registry URL and bootstrap servers. Run it by using this command

```
java -jar kafka-stream-test.jar "http://localhost:8081" "localhost:9092" ( for local settings)
```
- d) Create a repository in <https://bitbucket.org>, push your solution with a short README.md giving any details you deem important about your implementation as well as details on how to start your project
- e) Bonus points will be given for the following optional steps
 - Create a “notification” stream that produce an event when a coffee machine is out of the expected range (no more than 10 meters)
 - Create docker/docker-compose files for your implementation

