package com.example.demo.service;

import io.confluent.kafka.schemaregistry.client.CachedSchemaRegistryClient;

import io.confluent.kafka.schemaregistry.client.SchemaRegistryClient;

import io.confluent.kafka.schemaregistry.client.rest.exceptions.RestClientException;

import org.apache.avro.Schema;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.stereotype.Service;

import java.io.IOException;

@Service

public class SchemaService {

private final SchemaRegistryClient schemaRegistryClient;

public SchemaService(@Value("${schema.registry.url}") String schemaRegistryUrl) {

this.schemaRegistryClient = new CachedSchemaRegistryClient(schemaRegistryUrl, 100);

}

public String getSchema(String subject) {

try {

Schema schema = schemaRegistryClient.getLatestSchemaMetadata(subject).getSchema();

return schema.toString(true);

} catch (IOException | RestClientException e) {

e.printStackTrace();

return null;

}

}

}

package com.example.demo.controller;

import com.example.demo.service.SchemaService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class SchemaController {

private final SchemaService schemaService;

@Autowired

public SchemaController(SchemaService schemaService) {

this.schemaService = schemaService;

}

@GetMapping("/getSchema")

public String getSchema(@RequestParam String subject) {

return schemaService.getSchema(subject);

}

}