**QUEUE:**

(used in email integration, use JmsTemplate (Java’s default class))

The Queue interface is present in java.util package and extends the Collection interface is used to hold the elements about to be processed in FIFO(First In First Out) order.

It is an ordered list of objects with its use limited to inserting elements at the end of the list and deleting elements from the start of the list, (i.e.),

it follows the FIFO or the First-In-First-Out principle.

(-use queue in email integration for consuming mail sending time.)

**REDIS: (create CacheConfig and CacheOpeartion)**

(used in customUserDetailsService)

Redis, which stands for Remote Dictionary Server, is a fast, open source, in-memory, key-value data store

**Redis Template: (we store key value(permissions) in redis )**

Redis Template is the central class to interact with the Redis data.

It performs automatic serialization and deserialization between the given objects and binary data stored in Redis. By default Redis Template uses Java serialization.

**Lettuce Connection Factory:**

it is scalable thread-safe redis client, and it provides synchronous ,asynchronous ,reactive api’s to interact with Redis.

**Stored Procedure:**

A stored procedure provides an important layer of security between the user interface and the database.

It supports security through data access controls because end users may enter or change data, but do not write procedures.

Advantages :

The main advantages of stored procedure are given below:

Better Performance –

The procedure calls are quick and efficient as stored procedures are compiled once and stored in executable form. Hence the response is quick. The executable code is automatically cached, hence lowers the memory requirements.

Higher Productivity –

Since the same piece of code is used again and again so, it results in higher productivity.

Ease of Use –

To create a stored procedure, one can use any Java Integrated Development Environment (IDE). Then, they can be deployed on any tier of network architecture.

Scalability –

Stored procedures increase scalability by isolating application processing on the server.

Maintainability –

Maintaining a procedure on a server is much easier then maintaining copies on various client machines, this is because scripts are in one location.

Security –

Access to the Oracle data can be restricted by allowing users to manipulate the data only through stored procedures that execute with their definer’s privileges.

**File Storage:**

Used for upload files and download files.

**Interceptors:**

(api logger)

Used for stored all the requests of api’s with body(which is passed In payload) .

**Payment Integration:**

Using Razor Pay, create Razor pay account, create api keys from razor pay and download it,

Create api for create Order and then

add dependency of razor pay in pom.xml file

**Swagger:**

Swagger is an open-source tool to develop APIs, interact with APIs, and mainly generate documentation for APIs.

Third-party developers who are using your API want to integrate as quickly as possible to move forward in their software development. So they should soon understand the value and usage of your API.

**[**

Failed to start bean 'documentationPluginsBootstrapper'; nested exception

**To solve this error:**

**Add** spring.mvc.pathmatch.matching-strategy=ant-path-matcher

**in application.properties file.**

**]**

**Path:** [**http://localhost:8089/swagger-ui/index.html**](http://localhost:8089/swagger-ui/index.html)

**Docket class -from this will we configure swagger**

**ApiKey()-Used for pass apikey,it is a class that extends securityScheme**

**Roles and Permissions:**

Roles are used to collect permissions that define a particular function within the portal, according to a particular scope

**Actuator:**

An actuator is an additional feature of Spring that helps you to monitor and manage your application when you push it to production.

These actuators include auditing, health, CPU usage, HTTP hits, and metric gathering, and many more that are automatically applied to your application.

Add dependency:

<dependency>

<groupId> org.springframework.boot </groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

management.security.enabled=false

management.endpoints.web.exposure.include=\*

Add in application.properties file

url: <http://localhost:8089/actuator/health>

**Bulk Upload:**

Upload excel file in dummy database table.

Then move dummy tables’s data into main table.

Add dependency in pom.xml file

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>4.1.2</version>

</dependency>

Add one file in helper package.

**Apache Kafka:**

-Apache Kafka is distributed publish-subscribe messaging system.

-It was originally developed at LinkedIn and later on become a part of apache project.

-Kafka is fast, scalable, durable, fault-tolerant and distributed by design.

(**if every consumer belongs to different consumer group, all the messages will be consumed in every client**; that's called a 'publish-subscribe' model.)

Mainly kafka is used for real-time streams of data & used to collect big data for real time analysis.

LindkedIn, Microsoft and Netflix process billions of messages per day with kafka.

**Kafka Terminologies:**

1.Topic:

Basically, Kafka maintains feeds of messages in categories. And, messages are stored as well as published in a category/feed name that is what we call a topic. In addition, all Kafka messages are generally organized into Kafka topics.

(a unique name for kafka stream).

2.Kafka Broker:

There are one or more servers available in Apache Kafka cluster, basically, these servers (each) are what we call a **broker**.

(kafka server).

The producer and consumer don’t interact directly they use kafka broker to exchanges there messages.

3.Kafka Partitions:

In each broker in Kafka, there is some number of partitions. These Kafka partitions in Kafka can be both a leader or a replica of a topic. So, on defining a Leader, it is responsible for all writes and reads to a topic whereas if somehow the leader fails, replica takes over as the new leader.

(part of a topic).

4.Kafka Producers:

In simple words, the processes which publish messages to Kafka is what we call **Producers**. In addition, it publishes data on the topics of their choice.

(an application that send messages(or data) to kafka)

5.kafka Consumers:

The processes that subscribe to topics and process as well as read the feed of published messages, is what we call **Consumers**.

(an application that receives data from kafka).

6.Kafka Messages:

In one line, Message in Kafka is an information which travels from the producer to a consumer through Apache Kafka

7.Cluster:

A group of computers sharing wokload for a common purpose.

8.Offset:

A sequence id given to messages as they arrive in the partition.

**For start kafka server:**

**To start zookeeper server:**

.\**bin\windows\zookeeper-server-start.bat config\zookeeper.properties**

**(New cmd)**

**To start kafka server: (new cmd)**

**.\bin\windows\kafka-server-start.bat .\config\server.properties**

**To create Topic: (new cmd)**

**.\bin\windows\kafka-topics.bat --create --topic topic-test --bootstrap-server localhost:9092**

To create producer:

.\bin\windows\kafka-console-producer.bat --topic topic-test --bootstrap-server localhost:9092

>Hello world

>demo

>demo1

>hi sankita

Ctrl +c >> type y n enter(to stop)

To create consumer:

.\bin\windows\kafka-console-consumer.bat --topic topic-test --from-beginning --bootstrap-server localhost:9092

**Response:**

Hello world

demo

demo1

hi sankita

**Connect kafka with spring boot:**

**Add spring kafka dependency:**

<!-- https://mvnrepository.com/artifact/org.springframework.kafka/spring-kafka -->

<dependency>

<groupId>org.springframework.kafka</groupId>

<artifactId>spring-kafka</artifactId>

<version>2.8.6</version>

</dependency>

**Date Format:**

@Temporal(TemporalType.DATE)

private Date startDate = new Date();

@Temporal(TemporalType.DATE)

private Date endDate = new Date();

Payload: "startDate":"2018-09-22", "endDate":"2018-09-30"

RequestParam: 2022/09/22

What do you know about the company? Print 2nd largest number in an array. Divide number without using division operator. Water jug problem.