

Pub/Sub Queues

Implement a persistent (in memory) pub-sub queue mechanism with guaranteed delivery of every published message for all the subscribed consumers in that subscribed topic in the same order.

Features:

1. Expose topics
2. Publisher to push messages against a topic
3. Subscribe and unsubscribe from topic
4. Subscriber to consume from topic. A topic can have multiple subscribers
5. Maintain the state of consumption of each message in each topic for each consumer.
6. Maintain order of message consumption for each consumer.
7. Filtered message consumption in a topic by the consumer
8. Handle cases of consumers not being available. What to do with those messages?

--> create a topic

→ producer

→ consumer (subscribing)

→ push message to a topic!

→ consume it and log it!

TOPIC →

- LIST OF PRODUCERS,
- LIST OF CONSUMERS ,
- NAME
- LIST<MESSAGE_IDS>
- TOPIC ID

MESSAGE

- PRODUCER
- TOPIC_ID
- MESSAGE_ID

PRODUCER

- NAME
- ID
- LIST<TOPICS>

CONSUMER

- NAME
- ID
- LIST<TOPICS>
- OFFSET_ID

PULL MODEL → CONSUMER PULLS MESSAGES FROM TOPIC AND KEEPS TRACK OF OFFSET AGAINST A TOPIC

PUSH