RabbitMQ

Ata Ekren

04.12.2024

Table of Contents

What is RabbitMQ?

- RabbitMQ is an open-source message queue software used to facilitate the sending and receiving of messages between applications.

- A message queue is a system that allows one application to send messages (information) to other applications or service components. RabbitMQ queues these messages and ensures their reliable delivery.

Features of RabbitMQ

- **Lightweight and Reliable:** Provides a high-performance and durable messaging infrastructure.
- **Communication:** Enables communication in distributed systems.
- **Platform Independent:** Facilitates communication between different programming languages.
- **Performance:** Highly performant and scalable.

Basic Concepts of RabbitMQ

- **Queue:** The structure where messages are stored.
- **Exchange:** Determines which queue the messages are routed to.
- **Routing Key:** Ensures messages are routed to the correct queue.
- **Producer:** The application that sends messages.
- **Consumer:** The application that receives messages.

Principles of RabbitMQ

1. The producer generates a message.
2. The message is sent to the exchange.
3. The exchange routes the message to the appropriate queue.
4. Consumer receives and processes the message from the queue.

Advantages of RabbitMQ

Message Queuing

Ensures messages are delivered in the correct order.

Asynchronous Processing

Decouples applications, providing a more efficient workflow.

Processing Speed

Enhances performance through load balancing and parallel processing support.

RabbitMQ Usage Example 1

E-Commerce Platforms

Large e-commerce platforms use messaging systems like RabbitMQ to process user orders. When a user places an order, a series of tasks such as payment processing, stock updates, and shipment tracking are carried out in sequence. RabbitMQ queues these tasks and manages them asynchronously and in a scalable manner.

RabbitMQ Usage Example 2

Microservices Architecture and Distributed Systems

Many modern applications use microservices architecture. Large companies like Netflix and Uber use RabbitMQ to enable communication between different microservices. RabbitMQ allows microservices to operate independently while facilitating message transmission between them when needed. For example, when an Uber ride is booked, tasks like payment processing, vehicle dispatching, and user notifications are managed independently by microservices, but they are coordinated through RabbitMQ.

Benefits of Using RabbitMQ in This Project

Scalability: Multiple consumers can be added.

Reliability: Messages are safely stored in the queue.

Error Tracking: In case of an error, messages are not lost.

Independence: Services operate independently. Each service handles its own tasks.

Asynchronous: User receives an immediate response, continue in the background.

RabbitMQ Installation

1 Download Docker Desktop. Link

2 Run this command in the terminal:

```
docker run -d --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:management
```

Access the RabbitMQ panel by navigating to: http://localhost:15672

3 Add the RabbitMQ package to your project:

```
dotnet add package RabbitMQ.Client
```

4 Create the Producer and Consumer classes.

Thank you!

github.com/ataekren/MQProject Link

Ata Ekren