



RabbitMQ

İçerik

01	RabbitMQ Nedir?	06	RabbitMQ Kullanım Örneği 1
02	RabbitMQ Özellikleri	07	RabbitMQ Kullanım Örneği 2
03	RabbitMQ Temel Kavramlar	08	RabbitMQ Artıları
04	RabbitMQ Çalışma Prensibi	09	RabbitMQ Kurulum
05	RabbitMQ Avantajları	10	Son

RabbitMQ Nedir?

- RabbitMQ, açık kaynaklı bir mesaj kuyruğu (message broker) yazılımıdır. Uygulamalar arasında mesajların gönderilmesi ve alınmasını kolaylaştırmak için kullanılır.
- Mesaj kuyruğu, bir uygulamanın diğer uygulamalara ya da servis parçalarına mesaj (bilgi) göndermesi için kullanılan bir sistemdir. RabbitMQ, bu mesajları sıraya alır ve güvenilir bir şekilde teslim eder.

RabbitMQ Özellikleri



Hafif ve Güvenilir: Yüksek performanslı ve dayanıklı bir mesajlaşma altyapısı sunar.



İletişim: Dağıtık sistemlerde haberleşmeyi sağlar.



Platform Bağımsız: Farklı programlama dilleri arasında iletişim sağlar.



Performans: Yüksek performanslıdır ve ölçeklendirilebilir.

RabbitMQ Temel Kavramlar



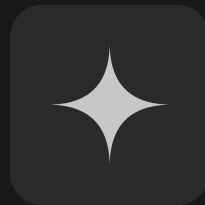
Queue (Kuyruk): Mesajların saklandığı yapıdır.



Exchange (Değişim): Mesajların hangi kuyruğa yönlendirileceğini belirler.



Routing Key (Yönlendirme Anahtarı): Mesajların doğru kuyruğa yönlendirilmesini sağlar.



Producer (Üretici): Mesajları gönderen uygulamadır.



Consumer (Tüketici): Mesajları alan uygulamadır.

RabbitMQ Çalışma Prensipleri

1

Üretici mesaj üretir

2

Mesaj exchange'e gönderilir

3

Exchange mesajı ilgili kuyruğa yönlendirir

4

Tüketici kuyruktan mesajı alır ve işler

RabbitMQ'nun Avantajları



Mesaj Sıralama

Mesajların sırasıyla iletilmesini sağlar.



Asenkron İşlem

Uygulamalar arasındaki bağlantıyı kopararak daha verimli bir iş akışı sunar.



İşlem Hızında Artış

Yük dengeleme ve paralel işlem desteği ile performansı artırır.

RabbitMQ Kullanım Örneđi 1

E–Ticaret Platformları

Büyük e-ticaret platformları, kullanıcı siparişlerini işlemek için RabbitMQ gibi mesajlaşma sistemlerini kullanır. Bir kullanıcı sipariş verdiğiinde, siparişin işlenmesi için ödeme, stok güncellemesi, kargo takibi gibi bir dizi işlem sırasıyla yapılır. RabbitMQ, bu işlemleri kuyruđa alarak asenkron ve ölçeklenebilir bir şekilde yönetir.

RabbitMQ Kullanım Örneği 2

Mikroservis Mimarisi ve Dağıtık Sistemler

Birçok modern uygulama mikroservis mimarisi kullanır. Netflix ve Uber gibi büyük şirketler, farklı mikroservislerin birbiriyle haberleşmesini sağlamak için RabbitMQ kullanır. RabbitMQ, mikroservislerin birbirinden bağımsız bir şekilde çalışmasını sağlarken, gerektiğinde birbirlerine mesaj iletmesini sağlar. Örneğin, bir Uber yolculuk rezervasyonu yapıldığında, ödeme, araç yönlendirme ve kullanıcı bildirim işlemleri mikroservisler tarafından bağımsız olarak yönetilir, ancak RabbitMQ sayesinde koordine edilir.

Bu projede RabbitMQ Kullanmanın Artıları

Ölçeklenebilirlik: Birden fazla tüketici (consumer) eklenebilir.

Güvenilirlik: Mesajlar kuyrukta güvenle saklanır.

Hata Takibi: Bir hata olursa mesaj kaybolmaz.

Bağımsızlık: Servisler bağımsız çalışır. Her servis kendi işini yapar.

Asenkron: Kullanıcı hemen cevap alır. İşlemler arkaplanda devam eder.

RabbitMQ Kurulum

1 Docker Desktop'u indir.

[Link](#)

2 Terminal'de bu komutu çalıştır:

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

http://localhost:15672 adresine giderek RabbitMQ paneline ulaşabilirsiniz.

3 Projenize RabbitMQ paketini yükleyin:

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

4 Üretici (Producer) ve Tüketici (Consumer) sınıflarını oluşturun.



Teşekkürler!