

## 在.net中集成RabbitMQ实现消息队列功能，实例解析

作者：微信公众号：【架构师老卢】

11-9 8:2

161



**概述：**使用 RabbitMQ 是一种常见的消息队列解决方案，用于构建可扩展的分布式应用程序。下面我将详细讲解如何在 .NET 中使用 RabbitMQ，包括常用功能和示例源代码。### 1、安装 RabbitMQ 首先，你需要安装 RabbitMQ 服务器。你可以从官方网站下载 RabbitMQ，并按照官方文档进行安装和配置。安装完成后，确保 RabbitMQ 服务器正在运行。### 2、使用 RabbitMQ 基本功能#### 2.1. 发布和订阅消息 RabbitMQ 的基本工作方式是生产者将消息发布到交换机，然后由消费者从队列中订阅消息。以下是一个示例： ``pla

使用 RabbitMQ 是一种常见的消息队列解决方案，用于构建可扩展的分布式应用程序。下面我将详细讲解如何在 .NET 中使用 RabbitMQ，包括常用功能和示例源代码。

### 1. 安装 RabbitMQ

首先，你需要安装 RabbitMQ 服务器。你可以从官方网站下载 RabbitMQ，并按照官方文档进行安装和配置。安装完成后，确保 RabbitMQ 服务器正在运行。

### 2. 使用 RabbitMQ 基本功能

#### 2.1. 发布和订阅消息

RabbitMQ 的基本工作方式是生产者将消息发布到交换机，然后由消费者从队列中订阅消息。以下是一个示例：

```
1 // 生产者
2 var factory = new ConnectionFactory() { HostName = "localhost" };
3 using (var connection = factory.CreateConnection())
4 using (var channel = connection.CreateModel())
5 {
6     channel.ExchangeDeclare(exchange: "logs", type: ExchangeType.Fanout);
7     var message = "Hello, RabbitMQ!";
8     var body = Encoding.UTF8.GetBytes(message);
9     channel.BasicPublish(exchange: "logs", routingKey: "", basicProperties: null, body: body);
10    Console.WriteLine("Sent: {0}", message);
11 }
12
13 // 消费者
14 var factory = new ConnectionFactory() { HostName = "localhost" };
15 using (var connection = factory.CreateConnection())
16 using (var channel = connection.CreateModel())
17 {
18     channel.ExchangeDeclare(exchange: "logs", type: ExchangeType.Fanout);
19     var queueName = channel.QueueDeclare().QueueName;
20     channel.QueueBind(queue: queueName, exchange: "logs", routingKey: "");
21
22     var consumer = new EventingBasicConsumer(channel);
23     consumer.Received += (model, ea) =>
24     {
25         var body = ea.Body.ToArray();
26         var message = Encoding.UTF8.GetString(body);
27         Console.WriteLine("Received: {0}", message);
28     };
29     channel.BasicConsume(queue: queueName, autoAck: true, consumer: consumer);
30 }
```

上述代码中，生产者将消息发布到名为 "logs" 的交换机，而消费者从该交换机创建一个队列并订阅消息。

#### 2.2. 路由消息

RabbitMQ 允许你使用路由键将消息路由到特定队列。以下是一个示例：

```
1 // 生产者
2 var factory = new ConnectionFactory() { HostName = "localhost" };
3 using (var connection = factory.CreateConnection())
4 using (var channel = connection.CreateModel())
5 {
6     channel.ExchangeDeclare(exchange: "direct_logs", type: ExchangeType.Direct);
7     var severity = "info";
8     var message = "Informational message.";
9     var body = Encoding.UTF8.GetBytes(message);
10    channel.BasicPublish(exchange: "direct_logs", routingKey: severity, basicProperties: null, body: body);
11    Console.WriteLine("Sent: [{0}] {1}", severity, message);
12 }
13
14 // 消费者
15 var factory = new ConnectionFactory() { HostName = "localhost" };
16 using (var connection = factory.CreateConnection())
17 using (var channel = connection.CreateModel())
18 {
19     channel.ExchangeDeclare(exchange: "direct_logs", type: ExchangeType.Direct);
20     var queueName = channel.QueueDeclare().QueueName;
21     var severity = "info";
22     channel.QueueBind(queue: queueName, exchange: "direct_logs", routingKey: severity);
23
24     var consumer = new EventingBasicConsumer(channel);
25     consumer.Received += (model, ea) =>
26     {
27         var body = ea.Body.ToArray();
28         var message = Encoding.UTF8.GetString(body);
29         Console.WriteLine("Received: [{0}] {1}", severity, message);
30     };
31     channel.BasicConsume(queue: queueName, autoAck: true, consumer: consumer);
32 }
```

在此示例中，消息被路由到具有特定路由键 "info" 的队列。

#### 2.3. 主题交换机

RabbitMQ 还支持主题交换机，允许你根据模式匹配消息的路由键进行订阅。以下是一个示例：

```
1 // 生产者
2 var factory = new ConnectionFactory() { HostName = "localhost" };
3 using (var connection = factory.CreateConnection())
4 using (var channel = connection.CreateModel())
5 {
6     channel.ExchangeDeclare(exchange: "topic_logs", type: ExchangeType.Topic);
7     var routingKey = "kern.critical";
8     var message = "Critical kernel error.";
9     var body = Encoding.UTF8.GetBytes(message);
10    channel.BasicPublish(exchange: "topic_logs", routingKey: routingKey, basicProperties: null, body: body);
11    Console.WriteLine("Sent: [{0}] {1}", routingKey, message);
12 }
13
14 // 消费者
15 var factory = new ConnectionFactory() { HostName = "localhost" };
16 using (var connection = factory.CreateConnection())
17 using (var channel = connection.CreateModel())
18 {
19     channel.ExchangeDeclare(exchange: "topic_logs", type: ExchangeType.Topic);
20     var queueName = channel.QueueDeclare().QueueName;
21     var routingKey = "kern.*";
22     channel.QueueBind(queue: queueName, exchange
23
24 : "topic_logs", routingKey: routingKey);
25
26     var consumer = new EventingBasicConsumer(channel);
27     consumer.Received += (model, ea) =>
28     {
29         var body = ea.Body.ToArray();
30         var message = Encoding.UTF8.GetString(body);
31         Console.WriteLine("Received: [{0}] {1}", routingKey, message);
32     };
33     channel.BasicConsume(queue: queueName, autoAck: true, consumer: consumer);
34 }
```

在此示例中，消息被路由到匹配模式 "kern.\*" 的队列。

### 3. 其他功能

RabbitMQ 还支持消息持久化、RPC（远程过程调用）、集群和安全等功能。你可以根据项目需求探索这些功能，并使用 RabbitMQ 的官方文档和 .NET 客户端库来实现。

上述示例涵盖了 RabbitMQ 的一些常见用例，希望可以帮助你入门 RabbitMQ 并在 .NET 应用程序中使用它。