

RabbitMQ Client Library

quinta-feira, 16 de setembro de 2021 13:31

O QUE É

A biblioteca cliente RabbitMQ é uma implementação da biblioteca cliente AMQP.

IModel	<i>AMQP data channel and provides the AMQP operations</i>
IConnection	<i>Represents an AMQP connection</i>
ConnectionFactory	<i>Constructs IConnection instances</i>

ConnectionParameters	<i>Configures a ConnectionFactory</i>
QueuingBasicConsumer	<i>Receives messages delivered from the server</i>

- IModel: interface classes principais da API e isso representa o canal de dados AMQP e fornece a maioria das operações AMQP.
- Oconnection: representa uma conexão AMQP com o intermediário de mensagem, um connection factory, que constrói instâncias de conexão.
- ConnectionParameters e QueuingBasicConsumer: classes úteis que incluem os parâmetros de conexão, que são usados para configurar o connection factory e o consumidor básico de enfileiramento, que recebe mensagens entregues do servidor.

COMO USAR

Você pode usar o código a seguir para se conectar ao RabbitMQ. Os parâmetros para se conectar ao broker são:

- Host name: servidor RabbitMQ ao qual se conectar;
- Nome de usuário: nome de usuário ao qual se conectar. Abaixo temos a conta de convidado mas você pode usar suas próprias contas e;
- Senha: senha com a qual você se conecta.

Assim que a conexão for estabelecida com o RabbitMQ, você poderá abrir um canal para o RabbitMQ com o código abaixo. Este canal agora pode ser usado para enviar e receber mensagens.

```
ConnectionFactory factory
```

```
    = new ConnectionFactory { HostName = "localhost",  
                             UserName = "guest",  
                             Password = "guest" };
```

```
IConnection connection = factory.CreateConnection();
```

```
IModel _model = connection.CreateModel();
```

Os aplicativos cliente funcionam com trocas e filas. As trocas nas filas devem ser declaradas antes de

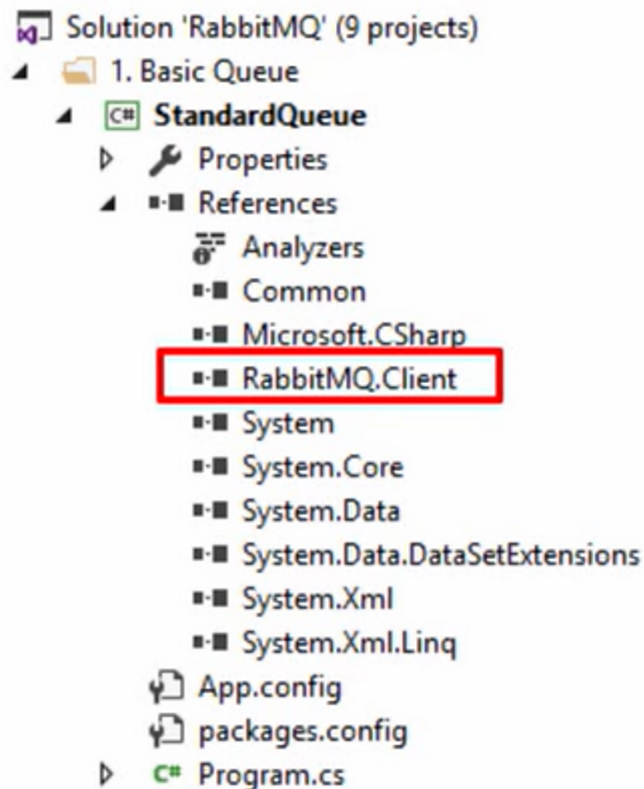
serem usadas. A declaração de qualquer um dos tipos de objetos garante a existência de um com aquele nome. Se não existir, é criado. O trecho de código abaixo declara uma troca e uma fila e as combina. Assim que a troca e as filas estiverem configuradas e vinculadas, você estará livre para começar a enviar e receber mensagens.

```
channel.ExchangeDeclare("MyExchange", "direct");  
  
channel.QueueDeclare("MyQueue");  
  
channel.QueueBind("MyQueue", ExchangeName, "");
```

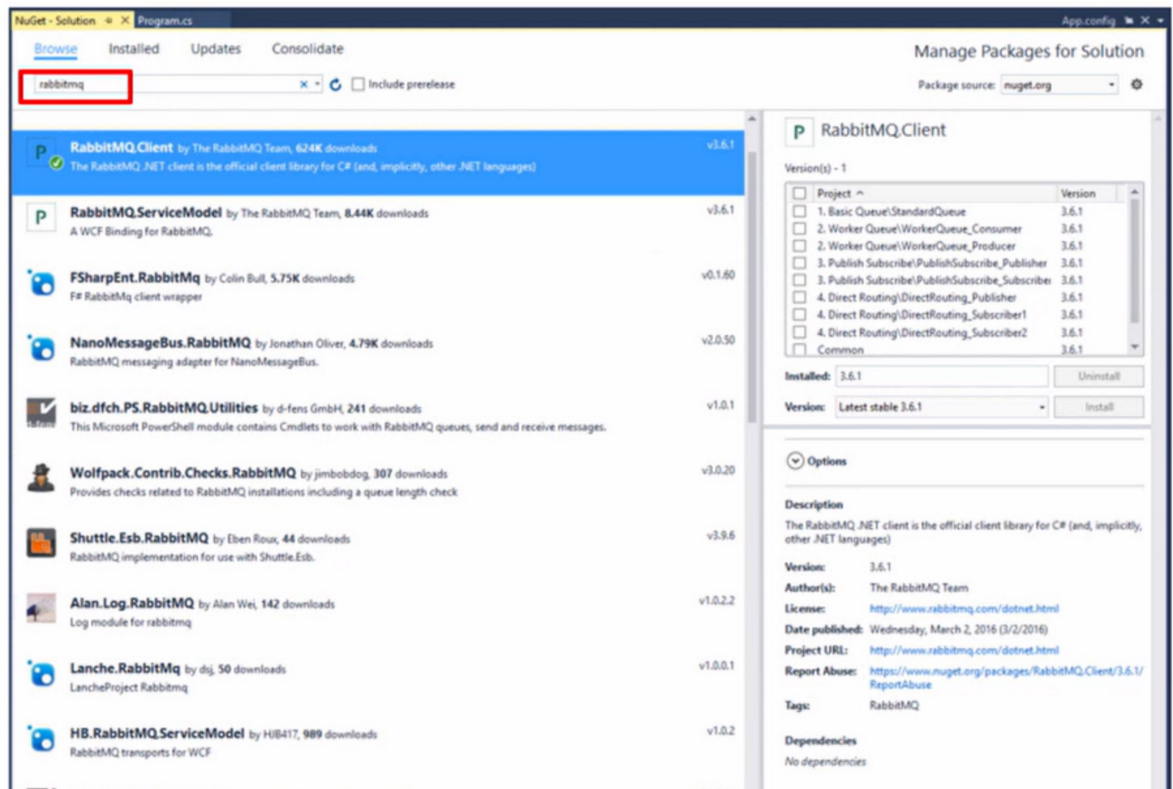
CONFIGURANDO A BIBLIOTECA RabbitMQ

Existem algumas maneiras de configurar a biblioteca cliente em seu aplicativo:

- Download: Ir para a página de downloads da biblioteca cliente RabbitMQ e baixar a biblioteca diretamente: <https://www.rabbitmq.com/clients.html>. Depois de fazer o download da biblioteca, você pode incluir uma referência ao RabbitMQ dot client dot dll em seu aplicativo, conforme mostrado abaixo.



- Visual Stúdio: Abre o gerenciador de pacotes no Visual studio no menu de ferramentas e, em seguida, no novo menu get package manager e, em seguida, seleciona manage e obtém os pacotes para a solução. Em seguida, na caixa de pesquisa, no canto superior direito da tela, você digita RabbitMQ. Isso mostrará a biblioteca do cliente RabbitMQ dot, na lista de bibliotecas na coluna do meio da tela. Clique duas vezes nesta biblioteca para instalá-la em seu projeto.



- Console de gerenciamento de pacotes: Para abri-lo, vá para o menu de ferramentas e selecione ferramentas, em seguida, selecione o novo menu obter pacote e, em seguida, selecione o console do gerenciador de pacotes. Quando a janela do console for exibida, você precisará digitar install dash package, RabbitMQ dot client, conforme mostrado abaixo (exemplo de saída). Agora você está pronto para começar a desenvolver código no RabbitMQ.

Install-Package RabbitMQ.Client

