Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

docker run --rm -it --hostname my-rabbit -p 15672:15672 -p 5672:5672 rabbitmq:3-management

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Utenza guest pass guest

Il codice della classe RoleModification aiuterà a effettuare le modifiche a un ruolo. Ora useremo queste 2 classi per aggiungere e rimuovere utenti da un Identity Role. Per questo aggiungeremo un'azione Update al Role Controller.

Immagine che contiene testo, Carattere, linea, schermata

Descrizione generata automaticamente

**Cos'è un Message Broker** Si può pensare a un message broker come a un ufficio postale. La sua responsabilità principale è quella di mediare i messaggi tra publisher e subscriber. Una volta che un message broker riceve un messaggio da un producer, lo indirizza a un subscriber. Il pattern del message broker è uno dei pattern più utili quando si tratta di disaccoppiare i microservizi.

**Cos'è uno Exchange** Gli Exchange sono esattamente ciò che suggerisce il nome. Sono exchange per messaggio. Proprio come abbiamo stop exchange, dove le persone scambiano azioni, un venditore vende azioni a un acquirente.

Exchange funge da router delle azioni. Allo stesso modo, Exchange in RabbitMQ instrada i messaggi da un produttore a uno o più consumatori. Un Exchange utilizza attributi di intestazione, chiavi di routing e binding per instradare i messaggi. (Ne parleremo in dettaglio più avanti) In RabbitMQ, infatti, i messaggi non vengono mai pubblicati in una coda, passano sempre attraverso un Exchange. Anche quando inviamo un messaggio a una coda, utilizza l'impostazione predefinita

**Type of Exchanges**

Direct: lo scambio diretto usa la chiave di routing nell'intestazione per identificare la coda a cui deve essere inviato il messaggio. La chiave di routing è un valore di intestazione impostato dal produttore. E il consumatore usa la chiave di routing per associarsi alla coda. Lo scambio esegue la corrispondenza esatta dei valori della chiave di routing.

**Topic**: anche lo scambio di argomenti utilizza la chiave di routing, ma non esegue una corrispondenza esatta sulla chiave di routing, ma esegue una corrispondenza di pattern basata sul pattern.

**Header**: lo scambio di header indirizza i messaggi in base ai valori di header ed è molto simile allo scambio di argomenti.

Fanout: come suggerisce il nome, lo scambio di fanout indirizza i messaggi a tutti gli

**RabbitMQ Topic Exchange, Headers Exchange and Fanout Exchange (Part 3, RabbitMQ in .NET Core)**

**Topic**: anche lo scambio di Topic utilizza la chiave di routing, ma non esegue una corrispondenza esatta sulla chiave di routing, ma esegue una corrispondenza di pattern basata sul pattern. Intestazione: lo scambio di header indirizza i messaggi in base ai valori di header ed è molto simile allo scambio di argomenti. Fanout: come suggerisce il nome, lo scambio di fanout indirizza i messaggi a tutti gli

**Building Reactive Microservice with RabbitMQ and ASP.NET Core**

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente