Sprawozdanie

# Rabbit MQ

Paweł Ambroziak 125 NCI B

Rabbit jest Message Broker, który służy do przekazywania wiadomości między systemami. Przekształca wiadomości z formalnego protokołu wiadomości wysyłającego na oficjalny protokół wiadomości odbiorcy.

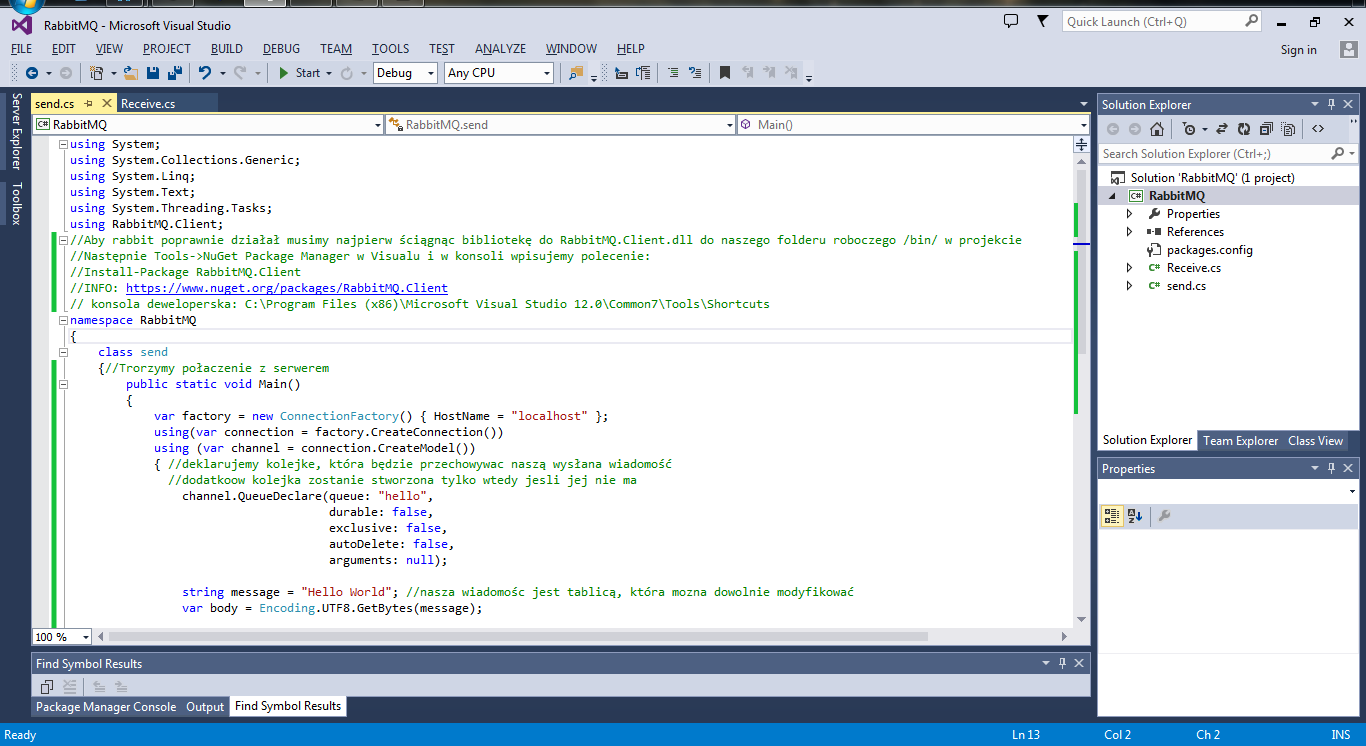
**Jaka jest rola i zadania protokołu AMQP?**

* **AMQP** (*Advanced Message Queuing Protocol*) to otwarty standard, który definiuje protokół dla systemów, które chcą wymieniać wiadomości.

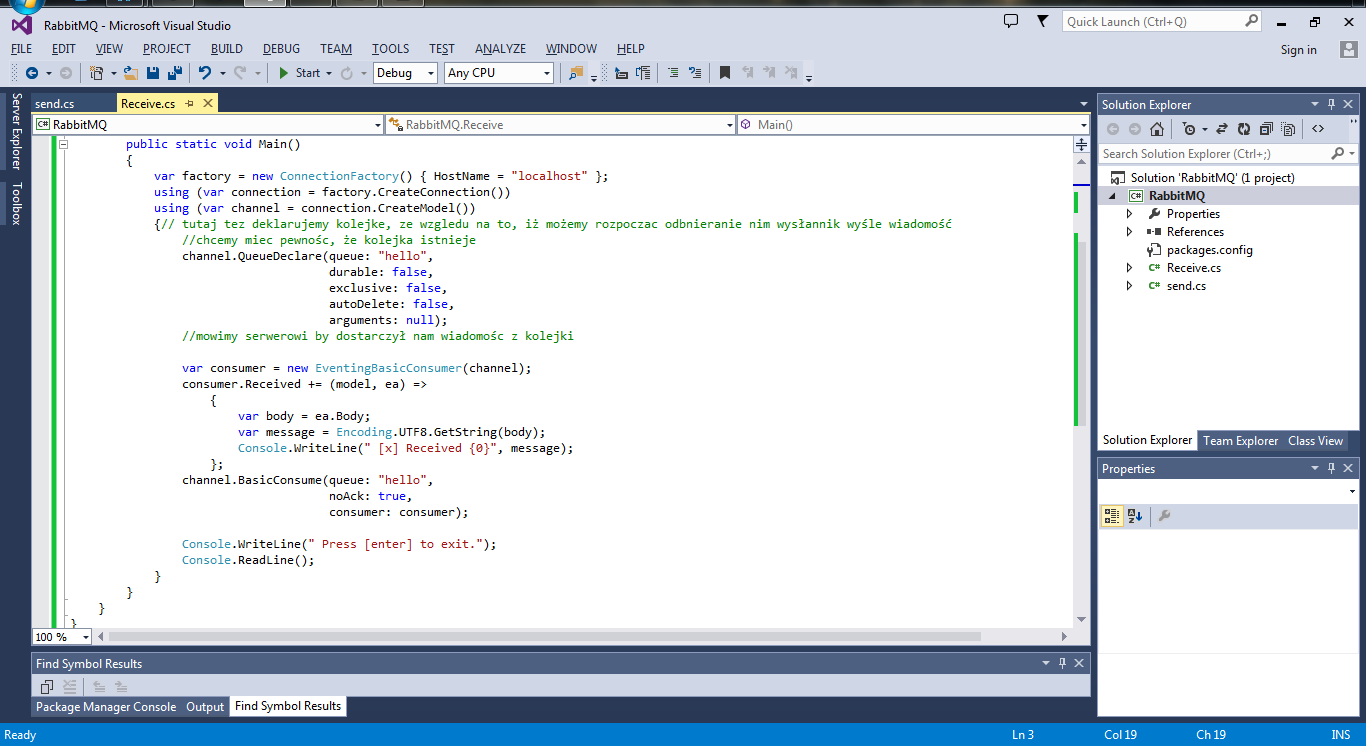
**Wyjaśnij ideę systemu powiadamiania (*messaging system*).**

* ***Messaging system*** zapewnia walidację, przechowywanie, routing i transformację wiadomości, które są przesyłane między różnymi systemami. Te systemy odpowiadają za przygotowanie wiadomości, które będą przesyłane poprzez *messaging system* (zwany także **messaging broker** lub **message broker**) do innych systemów połączonych z *message brokerem*.

Aby rabbit poprawnie działał musimy najpierw ściągnąc bibliotekę RabbitMQ.Client.dll do naszego folderu roboczego /bin/ w projekcie. Następnie Tools->NuGet Package Manager w Visualu i w konsoli wpisujemy polecenie: *Install-Package RabbitMQ.Client*



Na screenie przedstawiony jest proces połączenia z serwerem Rabbita oraz deklaracja kolejki, dzięki której wiadomość może oczekiwać na reakcje odbiorcy, który ja odbierze. Nasza wiadomość wysyłana do kolejki jest tablica, którą można dowolnie modyfikować. Program ten wykona się raz wysyłając wiadomość i przestanie działać.



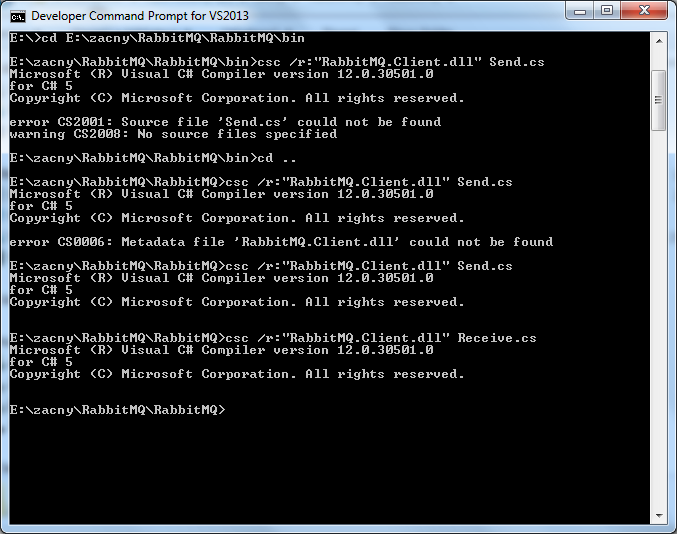
Program otrzymujący również tworzy kolejkę, gdyż chcemy by ona istniała nim wyślemy wiadomość. Program ten działa cały czas i dostarcza nam z kolejki wiadomość.

Aby móc stworzyc programy jako exe, należy uruchomic konsole deweloperską oraz wpisac polecenie:

*csc /r:"RabbitMQ.Client.dll" Send.cs*

*csc /r:"RabbitMQ.Client.dll" Receive.cs*

Konsola Deweloperska:



Repozytorium: https://github.com/PowerAmbroz/RabbitMQ.git