

Wprowadzenie do systemów rozproszonych

Norbert Białas

Informatyka Stosowana

Ćwiczenie 4a

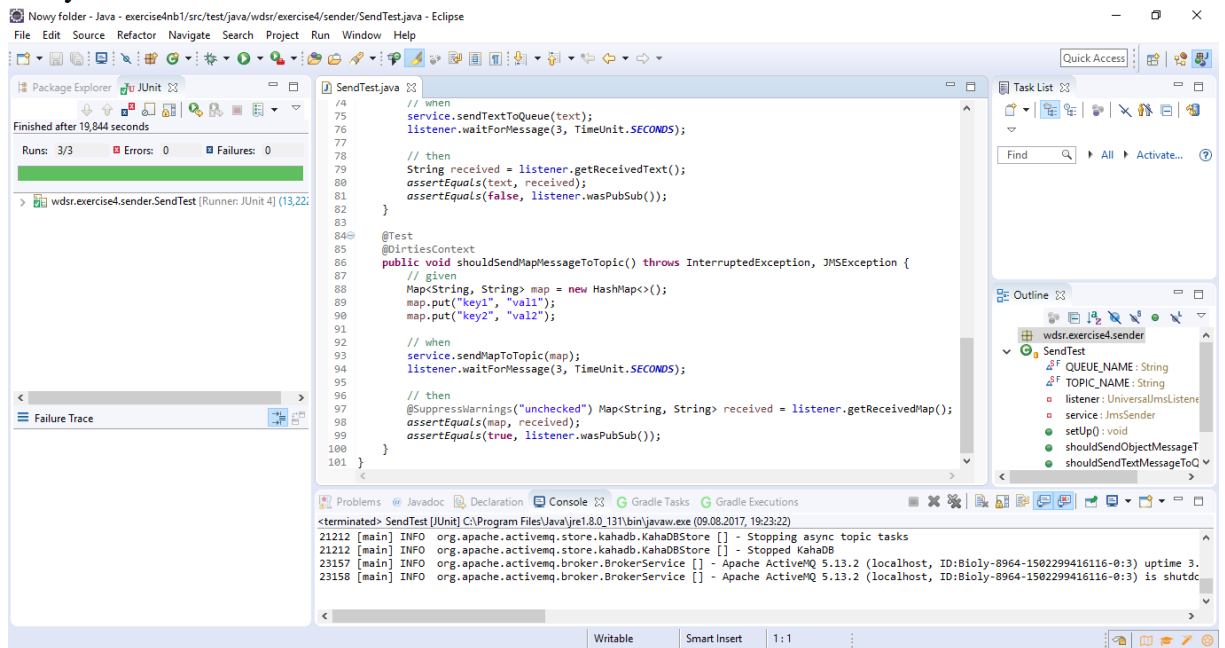
Celem ćwiczenia jest implementacja metod w klasie `JmsSender`: **`sendOrderToQueue`**, **`sendTextToQueue`**, **`sendMapToTopic`** oraz metod **`JmsQueueReceiver`**, **`registerCallback`** oraz **`shutdown`** w klasie `JmsQueueReceiver`.

Klasa *JmsSender* odpowiada za wysyłanie komunikatów do brokera JMS.

Klasa *JmsQueueReceiver* odpowiada za ich odbieranie.

W wyniku prawidłowej implementacji wszystkie testy jednostkowe powinny zakończyć się powodzeniem.

1. Testy :



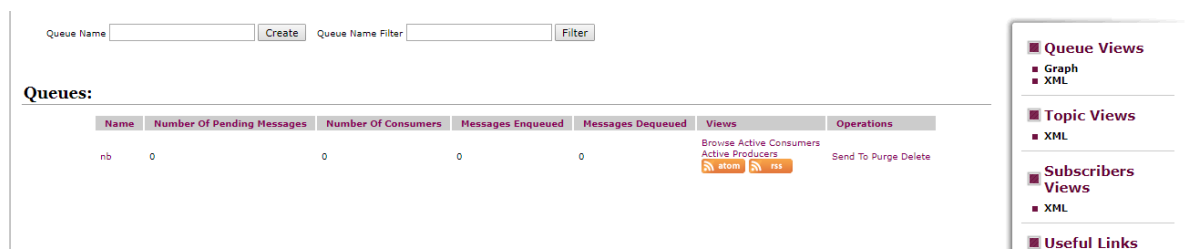
Ćwiczenie 4b

Cel ćwiczenia to napisanie programu który wyśle na kolejkę 20 tysięcy komunikatów podzielonych na 2 transze. Każdy ma wysłać 10 tysięcy.

Delivery mode w pierwszej części ma być ustawione na PERSISTENCE a druga część na NON_PERSISTENCE. Kolejną częścią jest napisanie programu który te komunikaty odbierze z kolejki.

Program działa prawidłowo jednak ze względu na możliwości sprzętowe screeny są robione w trakcie wysyłania i odbierania.

Uruchomienie ActiveMQ w przeglądarce w zakładce kolejki.



Dodawanie do kolejki:

Queues:

Name	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
BIOLY.QUEUE	17501	0	17501	0	Browse Active Consumers Active Producers atom rss	Send To Purge Delete

Odbieranie z kolejki:

BIOLY.QUEUE	0	1	18480	18480	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
-------------	---	---	-------	-------	---	--

Wysłanie na kolejkę 20 tysięcy komunikatów jest bardzo obciążające dla sprzętu na który testy są przeprowadzane ale wnioski są następujące :

Wysłanie PERSISTENT jest dłuższe od wysyłania NON_PERSISTENT.

Ćwiczenie 4c

Branch: persistence_topic_publisher.

Celem ćwiczenia jest napisanie dwóch programów. W tym wypadku nie będziemy używać kolejki a mechanizm publish – subscribe.

Uruchomiony program do wysyłania komunikatów :

Topic Name [Create](#)

Topics

Name ↑	Number Of Consumers	Messages Enqueued	Messages Dequeued	Operations
ActiveMQ.Advisory.Connection	0	2	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.Producer.Topic:BIOLY.TOPIC	0	2	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.Topic	0	1	0	Send To Active Subscribers Active Producers Delete
BIOLY.TOPIC	0	20000	0	Send To Active Subscribers Active Producers Delete

Branch: persistence_topic_subscriber

Uruchomiony program do odbierania komunikatów:

Topics

Name ↑	Number Of Consumers	Messages Enqueued	Messages Dequeued	Operations
ActiveMQ.Advisory.Connection	0	4	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.Consumer.Topic.BIOLY.TOPIC	0	1	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.Producer.Topic.BIOLY.TOPIC	0	3	0	Send To Active Subscribers Active Producers Delete
ActiveMQ.Advisory.Topic	0	1	0	Send To Active Subscribers Active Producers Delete
BIOLY.TOPIC	1	20000	0	Send To Active Subscribers Active Producers Delete