

2014

JMS Chat



Inhaltsangabe

Inhalt

Aufgabenstellung.....	2
Designüberlegung.....	3
Aufwandschätzung	3
Eigentlicher Aufwand	4
Arbeitsdurchführung	4
Quellen	5

Aufgabenstellung

Implementieren Sie eine Chatapplikation mit Hilfe des Java Message Service. Verwenden Sie Apache ActiveMQ (<http://activemq.apache.org>) als Message Broker Ihrer Applikation. Das Programm soll folgende Funktionen beinhalten:

- Benutzer meldet sich mit einem Benutzernamen und dem Namen des Chatrooms an.

Beispiel für einen Aufruf:

```
vsdbchat <ip_message_broker> <benutzername> <chatroom>
```

- Der Benutzer kann in dem Chatroom (JMS Topic) Nachrichten an alle Teilnehmer eine Nachricht senden und empfangen.

Die Nachricht erscheint in folgendem Format:

```
<benutzername> [<ip_des_benutzers>]: <Nachricht>
```

- Zusätzlich zu dem Chatroom kann jedem Benutzer eine Nachricht in einem persönlichen Postfach (JMS Queue) hinterlassen werden. Der Name des Postfachs ist die IP Adresse des Benutzers (Eindeutigkeit).

Nachricht an das Postfach senden:

```
MAIL <ip_des_benutzers> <nachricht>
```

Eignes Postfach abfragen:

MAILBOX

- o Der Chatraum wird mit dem Schlüsselwort EXIT verlassen. Der Benutzer verlässt den Chatraum, die anderen Teilnehmer sind davon nicht betroffen.

Gruppenarbeit: Die Arbeit ist in einer 2er-Gruppe zu lösen und über das Netzwerk zu testen! Abnahmen, die nur auf localhost basieren sind unzulässig und werden mit 6 Minuspunkten benotet!

Software:

Apache ActiveMQ Installationspaket ist unter Resource verfügbar.

Designüberlegung

Wir brauchen einen Chatraum der sowohl senden und empfangen kann und Clients die sich zum Chatraum verbinden und auch senden und empfangen können. Beide sollen via Thread laufen und immer auf einer Chatnachricht warten. Weiteres brauchen müssen wir mit der JMS Queue das Postfach erstellen(keine Designüberlegung vorhanden). Zum Exit Befehl haben wir uns gedacht dass die Verbindung getrennt wird und die Threads gestoppt werden

Aufwandschätzung

Aufgabe

Schätzung in Stunden

Installation ActiveMQ	1/2
Benutzer melden sich in Chatr an	3
Benutzer können schreiben(Format)	2
Postfach(JQueue)	4
Mit Exit Befehl beenden	1.5
Gesamt	11

Eigentlicher Aufwand

Aufgabe

Schätzung in Stunden

Installation ActiveMQ	1
Benutzer melden sich in Chatr an	5
Benutzer können schreiben(Format)	3
Postfach(JQueue)	3+(unfertig)
Mit Exit Befehl beenden	1/2
Gesamt	12

Arbeitsdurchführung

Zuerst haben wir ActiveMq runtergeladen und installiert. Danach haben wir die Änderungen :

Für das Debian-Image muss eine Änderung durchgeführt werden, damit Apache Active MQ gestartet werden kann.

> Öffnen Sie die Datei bin/activemq mit einem Editor Ihrer Wahl

> Suchen Sie nach folgender Zeile

`ACTIVEMQ_OPTS_MEMORY="-Xms1G -Xmx1G"`

> Ersetzen Sie diese Zeile mit:

`ACTIVEMQ_OPTS_MEMORY="-Xms512M -Xmx512M"`

Durchgeführt

Wir haben das Jar von Elearning heruntergeladen und die Klassen mit ActiveMq zum Laufen gebracht. Danach haben wir versucht die Klassen so zu erweitern bzw neu schreiben das sie die Funktionen der Angabe erfüllen. Nach einigen Fehlschlägen und falsches Denken haben wir in den examples von ActiveMq eine bessere Vorlage gefunden. Diese konnten wir so erweitern dass wir man ein Chatraum aufmachen kann und Leute sich übers Netzwerk schreiben können. Danach wurde das Format verbessert und der Exit befehl hinzugefügt. Mit JMSQueue hatten wir einige Probleme.

Quellen

<http://activemq.apache.org/index.html>

<http://www.academictutorials.com/jms/jms-introduction.asp>

<http://docs.oracle.com/javaee/1.4/tutorial/doc/JMS.html#wp84181>

<http://www.openlogic.com/wazi/bid/188010/How-to-Get-Started-with-ActiveMQ>

<http://jmsexample.zcage.com/index2.html>

http://www.onjava.com/pub/a/onjava/excerpt/jms_ch2/index.html

<http://www.oracle.com/technetwork/systems/middleware/jms-basics-jsp-135286.html>

<http://java.sun.com/developer/technicalArticles/Ecommerce/jms>