

## Lernaufgabe

### LE 5 – Verteilte Architekturen und Messaging-basierte Kommunikation

#### Aufgabe 2: Messaging mit JMS

##### Lernziel

Sie sind in der Lage, die Funktionsweise von Messaging-Applikationen und von Messaging-oriented Middle-ware (MOM) zu erläutern sowie deren Einsatzzweck zu illustrieren.

##### Aufgabe

Die auf Moodle abgelegten Programme demonstrieren eine einfache Point-to-Point (P2P) Messaging-Anwendung in JMS (s. LE 5, Aufgabe 2). Ein Producer erzeugt Test-Nachrichten, die von einem Consumer empfangen und angezeigt werden.

Probieren Sie die Programme aus und beantworten Sie folgende Fragen:

- Was passiert, wenn ein Consumer gestartet wurde, aber keine Message in der Queue ist bzw. kein Producer läuft? Wo wird dieses Verhalten eingestellt?
- Was passiert mit einer Nachricht, wenn nie ein Consumer gestartet wird, nachdem ein Producer eine Nachricht erstellt bzw. gesendet hat? Wo wird dieses Verhalten eingestellt?
- Welche Art von Queue – transiente oder persistente – verwenden die Programme (s. Delivery-Modus in der Theorie)? Wo wird dieser Delivery-Modus eingestellt und was ist der Default?

##### Vorgehen

1. Entpacken Sie das ZIP-File in einem Arbeitsverzeichnis und importieren Sie das Projekt in Eclipse (Import->Existing Project into Workspace) oder einer anderen IDE.
2. Sie können den verwendeten Message-Broker (ActiveMQ) in einer Gruppe mit nur einem installierten Message-Broker oder lokal auf Ihrem Rechner verwenden (s. Abschnitt Hinweise, Tipps). Falls Sie keine eigene Installation verwenden wollen, können Sie direkt mit Schritt 6 weiterfahren (s. dazu noch den Abschnitt Hinweise, Tipps).
3. Installieren Sie ActiveMQ auf Ihren Rechner, falls Sie dies noch nicht gemacht haben:  
<http://activemq.apache.org/download.html> (ZIP-File, das auf dem Rechner entpackt werden muss. Achtung: Auf Windows darf es keine Leerzeichen im Verzeichnis haben!).
4. Öffnen Sie für den Message Broker (ActiveMQ) ein Command- bzw. Shell-Prompt und wechseln Sie ins Verzeichnis der Installation und in den Ordner `/bin`.
5. Starten des Message Brokers: `activemq start` (MacOS, Linux: `./activemq console`)
6. Öffnen Sie für den Producer und den Consumer je ein Command- bzw. Shell-Prompt und wechseln Sie ins das Arbeitsverzeichnis des entpackten ZIP-Files.
7. Der Producer wird aufgerufen mit: `java -cp "bin;lib/*" Producer Hallo` (Linux: „:“ anstatt „;“) oder einfach mit dem beigelegten Skript `producer Hallo` (Windows) bzw. `./producer.sh Hallo` (MacOS, Linux).
8. Der Consumer wird aufgerufen mit: `java -cp "bin;lib/*" Consumer1 30000` (Linux: „:“ anstatt „;“) oder einfach mit dem beigelegten Skript `consumer1 30000` (Windows) bzw. `./consumer.sh 30000` (MacOS, Linux).
9. Probieren Sie auch den zweiten `consumer2` aus und vergleichen Sie die Art, wie Messages konsumiert werden mit dem `consumer1`.

##### Hinweise, Tipps

Die Aktivitäten des Message Brokers sowie Nachrichtenziele (Queues, Topics) können mit Hilfe einer Web Console (Browser) überwacht werden (URL: <http://localhost:8161/admin>). Diese Anwendung ist geschützt mit Benutzernamen „admin“ und Passwort „admin“. Mit der der Web Console können auch Nachrichten erstellt und gelöscht werden (Reiter „Send“ in der Web Console).



Die Verbindungsdaten für den Zugriff auf die laufende ActiveMQ-Instanz und eine Queue bzw. Topic sind in der Datei `jndi.properties` im Verzeichnis `/bin` konfiguriert. Arbeit man mit verschiedenen Rechnern, so kann natürlich statt der vorkonfigurierten IP-Adresse `localhost` die IP-Adresse, auf dem ActiveMQ läuft, angegeben werden.

Mit dem Programm `QueueInfo` kann die Queue abgefragt werden, ohne sie zu löschen. Das Programm wird aufgerufen mit: `java -cp "bin;lib/*" QueueInfo` (Linux: „:“ anstatt „;“) oder einfach mit dem beigelegten Skript `queueinfo` für (Windows) bzw. `./queueinfo.sh` (MacOS, Linux)..

ActiveMQ implementiert auch ein RESTful API, welches erlaubt mittels HTTP(S) Messages zu publizieren und zu konsumieren. Weitere Informationen dazu finden Sie unter <http://activemq.apache.org/rest.html>.

### Ergebnis

Fassen Sie die Erkenntnisse zur den Fragen zusammen und diskutieren Sie sie in Murmelgruppen.

**Zeit:** 30'