

Aufgabenblatt 4: EJB

(Zu bearbeiten bis 13.01.2020)

Client/Server-Programmierung

Wintersemester 2019/20

Aufgabe 1: EJB Hello World

Laden Sie sich zunächst den Code zum Aufgabenblatt von der CSP-Webseite herunter, entpacken Sie das Archiv und wechseln Sie ins Verzeichnis EJB/HELLO_WORLD.

Der Code zeigt die Verwendung einer stateless Session-Bean mit OpenEJB als Container, wie in der Vorlesung skizziert. Der Aufruf erfolgt über eine clientseitige Java-Applikation, welche auf den OpenEJB-Container zugreift. Machen Sie sich mit der Bedeutung und dem Aufbau der Dateien vertraut. Führen Sie eine Kompilierung, Paketierung und ein Deployment nach OpenEJB durch. Eine Anleitung dazu finden Sie bei den Materialien zum Übungsblatt in Form eines EJB-Tutorials, sowie in den Vorlesungsfolien.

Aufgabe 2: Börsenanwendung mit EJB

Realisieren Sie Ihre Börsenanwendung mit Hilfe von zwei Session Beans. Die Klasse `StockExchange` wird dabei zu einer *stateless* Session Bean, `Depot` sollten Sie als *stateful* Session Bean realisieren. Da EJB die explizite Erzeugung einer Bean-Instanz nicht vorsieht, muß die Methode `StockExchange.openDepot()` entfallen. Stattdessen erhält der Client eine Referenz auf sein sitzungslokales Depot über einen JNDI-Lookup der *stateful* Session Bean.

Testen Sie die EJB mit gleichzeitig geöffneten Client-Sitzungen.

Zusatz (für Motivierte):

Schreiben Sie die Methoden Ihrer Anwendung so um, dass sie nicht mehr direkt über JDBC auf die MySQL-Datenbank zugreifen, sondern stattdessen Entities verwenden. Dazu müssen Sie die OpenEJB-Konfigurationsdatei `$OPENEJB_HOME/conf/openobj.xml` um die folgenden Einträge erweitern, um auf die MySQL-Datenbank zugreifen zu können:

```
<Resource id="My SQL DB" type="DataSource">
  JdbcDriver com.mysql.jdbc.Driver
  JdbcUrl jdbc:mysql://bslabserver1.lab.bvs/cspdb
  UserName
  Password
  JtaManaged true
</Resource>
<Resource id="My Unmanaged SQL DB" type="DataSource">
  JdbcDriver com.mysql.jdbc.Driver
  JdbcUrl jdbc:mysql://bslabserver1.lab.bvs/cspdb
  UserName
  Password
  JtaManaged false
</Resource>
```

Um in den Entities die Java Namenskonventionen einhalten zu können und die Beziehung zwischen den Entities zu realisieren, müssen Sie ein passendes OR-Mapping mit Hilfe von Annotationen spezifizieren (siehe Kapitel 5.4.9 der Vorlesung und das Beispiel „Verwendung von Entities mit Beziehungen“ auf der Vorlesungs-Homepage).

Aufgabe 3: Börsenanwendung mit Entities (für Motivierte)

Realisieren Sie das Depot der Börsenanwendung mit Hilfe von Entities und der in OpenEJB eingebauten Datenbank HSQLDB. Sie müssen das Depot dazu analog wie in dem Beispiel in Kap. 5.4.9 des Skripts (siehe das Code-Beispiel „Verwendung von Entities mit Beziehungen“ auf der Vorlesungs-Homepage) mit Hilfe von zwei Datenbank-Tabellen (und damit zwei Entity-Klassen) realisieren: Die Tabelle `depot` enthält Nutzernamen und Depot-Nummer, die Tabelle `buchung` die Buchungen für alle Depots. Die Verbindung wird durch eine *One-to-many*-Assoziation über eine eindeutige Buchungs-ID hergestellt. Den bei der Authentifizierung angegebenen Benutzernamen können Sie, wie in Kap. 5.4.8 des Skripts (Beispiel zur *Dependency Injection*) angeben, mit Hilfe der Methode `getCallerPrincipal()` der Klasse `SessionContext` erfragen.