



## Softwarepraktikum im Grundstudium im Wintersemester 2002/2003

### 5. Übung, 28. November 2002

In dieser Übung soll die Webanwendung um eine Benutzerverwaltung ergänzt werden. Die Servlet-Engine Tomcat wird so konfiguriert werden, dass die Aufgabenverwaltung nur autorisierten Benutzern zugänglich ist.

#### 1. Aufgabe (Benutzerverwaltung)

In Ihren Datenbanken stehen Ihnen nun drei weitere Tabellen zur Verfügung:

`users` (Verwaltet die einzelnen Benutzer):

<code>username</code>	<code>VARCHAR(50) (GetString)</code>
<code>passwd</code>	<code>VARCHAR(20) (GetString)</code>
<code>email</code>	<code>VARCHAR(50) (GetString)</code>

`user_roles` (Verwaltet die Rollen der Benutzern):

<code>username</code>	<code>VARCHAR(50) (GetString)</code>
<code>role</code>	<code>VARCHAR(50) (GetString)</code>

`roles` (Enthält alle Rollen, die vergeben werden können):

<code>role</code>	<code>VARCHAR(50) (GetString)</code>
-------------------	--------------------------------------

Damit Sie Ihr System besser testen können, wurden diese Tabellen bereits mit einigen Daten gefüllt. Die Tabelle `roles` enthält zwei vordefinierte Rollen:

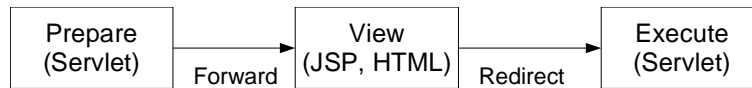
- `manager`: Verwaltet Benutzer
- `aufgabe`: Erlaubt Zugriff auf Webanwendung

Zusätzlich wurde ein Benutzer `test` mit dem Password `test` angelegt. Diesem Benutzer wurden die Rollen `manager` und `aufgabe` zugeordnet.

Implementieren Sie die Benutzerverwaltung als Teil Ihrer Webapplikation. Beachten Sie dabei folgende Hinweise:

- In `index.html` muss ein Link zur Benutzerliste eingetragen werden.
- Analog zu den Klassen `Task` und `Tasks` müssen die Klassen `User` und `Users` angelegt werden.
- `StorageAccess` muss um die Zugriffsfunktionen auf Datensätze aus den drei neuen Tabellen erweitert werden. Beachten Sie, dass keine speziellen Zugriffsfunktionen für die Tabellen `roles` und `user_roles` implementiert werden müssen, da diese beim Anlegen und Ändern von Benutzern manipuliert werden.
- Ähnlich wie für Aufgaben soll das Anlegen, Anzeigen, Ändern und Löschen von Benutzern implementiert werden. Beachten Sie hierbei, dass ein Benutzer mehrere Rollen haben kann. Beim Anlegen und Ändern von Benutzern sollten daher die Rollen in einer Auswahlliste mit Mehrfachselektion gewählt werden können. Für das Anzeigen von Benutzern reicht eine durch Kommata getrennte Liste der Rollen.
- Das Ändern eines Benutzers ist wesentlich einfacher, wenn zuerst alle seine Rollen gelöscht und anschließend neu angelegt werden.
- Es wurden Trigger definiert, die die referentielle Integrität der Datenbank sicherstellen. Diese sorgen beispielsweise dafür, dass automatisch die Rollen eines Benutzers gelöscht werden, wenn der Benutzer gelöscht wird. Dasselbe gilt auch für die Rollen. Achten Sie daher auf die Reihenfolge Ihrer SQL-Anweisungen!
- Die Tabelle `roles` wird nicht von `StorageAccess` geändert. Neue Rollen werden ausschliesslich direkt in der Datenbank eingetragen.

Die Beziehung zwischen den verschiedenen Servlets, JSP- und HTML-Seiten ist dabei immer wie folgt (vgl. Übung 2):



Für diese Aufgabe ergeben sich somit die folgenden Bezeichner:

- Anlegen von Benutzern: *NewUserPrepare*, *NewUserView*, *NewUserExecute*
- Anzeigen von Benutzern: *ShowUsersPrepare*, *ShowUsersView*
- Ändern von Benutzern: *ModifyUserPrepare*, *ModifyUserView*, *ModifyUserExecute*
- Löschen von Benutzern: *DeleteUsersPrepare*, *DeleteUsersView*, *DeleteUsersExecute*

## 2. Aufgabe (Konfiguration des Tomcat, Aktivieren der Benutzerüberprüfung für die Webapplikation)

Dem Tomcat muss mitgeteilt werden, dass die Benutzerberechtigungen aus der Datenbank gelesen werden müssen. Dies geschieht durch folgenden Eintrag in der Datei

~/i3\_spws/tomcat/conf/server.xml :

```
<Realm className="org.apache.catalina.realm.JDBCRealm"
  debug="99" driverName="org.postgresql.Driver"
  connectionURL="jdbc:postgresql://busoni/spws01_GROUP?user=USER;
  password=PASS"
  userTable="users"
  userNameCol="username"
  userCredCol="passwd"
  userRoleTable="user_roles"
  roleNameCol="role" />
```

Die Ursprüngliche Definition des Realms (`<Realm className="org.apache.catalina.realm.MemoryRealm"/>`) muss mit `<!--` und `-->` auskommentiert werden. Diese Änderungen finden im Element `<Engine name="Standalone"...` hinter dem Element `DefaultContext` statt. `GROUP`, `USER` und `PASS` muss dabei durch die Gruppennummer (zweistellig) und die Daten des Datenbankzugriffs ersetzt werden.

Jetzt muss noch die Benutzeranmeldung für die Webapplikation aktiviert werden. Dazu muss die Konfigurationsdatei der Webapplikation (`web.xml`) geändert werden. Fügen Sie folgende Einträge am Ende des Elements `web-app` ein:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Aufgabenverwaltung</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>aufgabe</role-name>
  </auth-constraint>
</security-constraint>

<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Aufgabenverwaltung</realm-name>
</login-config>
```

Der obere Eintrag definiert alle Seiten der Aufgabenverwaltung als eine Sicherheitszone, die nur für Benutzer mit der Rolle `aufgabe` zugänglich ist. Der untere Eintrag wählt für diese Anwendung die Methode `BASIC` als Authentifizierungsmechanismus aus, was bedeutet, dass der Browser den Benutzer in einer Dialogbox um die Anmeldung bittet.

## Testat

Donnerstag, 12.12.2002 von 10.00 - 11.45 Uhr