

Beschreibung

Das Auth Service unterstützt Realms auf einfache Art und Weise. In der Tabelle für die Authentifizierung/Authorisierung wird pro Benutzer ein durch Kommata getrenntes Feld „auth.login.realmfield“ gehalten. Es gibt die Realms an, in denen der Benutzer aktiv sein kann.

Der Inhalt von Seiten kann auf Basis des Realms (sicherlich meistgenutzte Möglichkeit) und/oder des Benutzernamens reglementiert werden.

Durch den Einsatz von Bricks in Kombination von Realms können „Portlet-ähnliche“ Seiten implementiert werden. Es kann innerhalb einer Page Content für eingeloggte Benutzer sowie für nicht-eingeloggte Benutzer hinterlegt werden. Je nach Status des Benutzers wird ein Teil der Seite angezeigt oder nicht. Dabei kann auch Content für verschiedene Realms oder Benutzer hinterlegt werden.

Beispiele

Beispiel eines HTML-Formulars, dass Benutzername und Passwort abfragt:

```
<%@ include file="/init.jsp" %>

<html>
<body>

<form method="post" action="/login.do">

<input type="text" name="qUsername" length="20">
<input type="password" name="qPassword" length="20">
<input type="button" name="Login">

</form>

</body>
</html>
```

Beispiel für einen Link, der ein Login durchführt:

```
/login.do?qUsername=abc&qPassword=abc
```

Beispiel einer(!) dmerce JSP, welche die qauth-Tags nutzt:

```
<%@ include file="/init.jsp" %>

<html>
<body>

<!-- 1. Teil: Content für normale Benutzer und Admins -->

<qauth:logged-in realm="user">
... hier steht content für normale benutzer einer website ...
</qauth:logged-in>
<qauth:logged-in realm="admin">
... darunter kann gleichzeitig content für admins angezeigt werden ... wie z.b. link zur administration
```

des darüber angezeigten contents für normale benutzer ...
</qauth:logged-in>

<!-- 2. Teil: Content für nicht angemeldete Benutzer -->

<qauth:not-logged-in>
... hier steht content für benutzer, die der website nicht bekannt sind ...
</qauth:not-logged-in>

<body>
</html>

Konfiguration des Auth Services

Innerhalb von etc/applications.xml müssen folgende Properties pro Applikation gesetzt werden.
„appl“ ist hier eine Beispiel-Applikation:

```
<property name="appl.auth.table" value="auth"/>  
<property name="appl.auth.usernamefield" value="username"/>  
<property name="appl.auth.passwordfield" value="password"/>  
<property name="appl.auth.realmfield" value="realm"/>  
<property name="appl.auth.datafields" value="firstname,lastname"/>  
<property name="appl.auth.login.successfullpage" value="/loginok.jsp"/>  
<property name="appl.auth.login.unsuccessfullpage" value="/loginnotok.jsp"/>  
<property name="appl.auth.logout.page" value=""/>
```

Abfrage des Status des Benutzers innerhalb einer JSP

<qauth:logged-in realm="<Name des Realms aus auth.table.realmfield, durch Kommata getrennt
auch mehrere Realms>" username="<Benutzername aus auth.table.usernamefield>">
... wird nur angezeigt, wenn man eingeloggt ist ...
</qauth:logged-in>

<qauth:not-logged-in>
... wird nur angezeigt, wenn man NICHT eingeloggt ist ...
... eingeloggte Benutzer sehen dies NICHT ...
</qauth:not-logged-in>

Abfrage der Datenfelder eines Benutzers

<qauth:datafield name="<feld aus auth.table, dass in auth.datafields angegeben wurde>">

Allgemein

Werden die Attribute „username“ und „realm“ bei den Tags nicht angegeben, wird nur nach dem Status des Benutzers entschieden (eingeloggt oder nicht eingeloggt).

Ausloggen per URL

/logout.do

Loggt den Benutzer der aktuellen Session aus und leitet auf die URL appl.auth.logout.page weiter.

Die Session selbst bleibt erhalten.