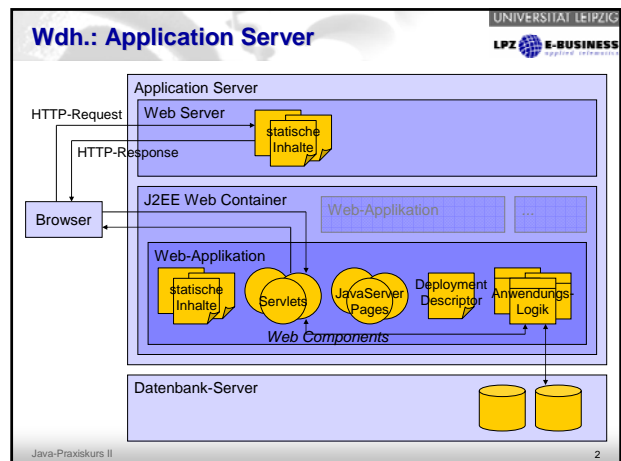


UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS
applied telematics

Übungen zum Java-Praxiskurs für Fortgeschrittene

Matthias Book
Malte Hülder

Java-Praxiskurs II WS 2005/2006



UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS

Wdh.: Servlet-Aufruf

- HTML-Formular (Beispiel):


```

<form action="/timeofday/login" method="get">
  Name: <input type="text" name="username"><br />
  Passwort: <input type="password" name="passwd"><br />
  <input type="submit" name="login" value="anmelden">
</form>
            
```
- erzeugte URL:


```

http://localhost:8080/timeofday/login?username=mbook
&passwd=javakurs&login=anmelden
            
```
- wird vom Web Container an das Servlet mit dem Alias `login` in der Web-Applikation `timeofday` übergeben

Context Root Servlet Alias ggf. Parameter

Java-Praxiskurs II 3

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS

Wdh.: Time of Day-Servlet

- Servlet zur Ausgabe der aktuellen Uhrzeit in einer HTML-Seite

```

import javax.servlet.*; import javax.servlet.http.*;
import java.util.*; import java.io.*;

public class TimeOfDayServlet extends HttpServlet {
    protected void doGet(
        HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body>");
        out.println("<p>Uhrzeit: " + new Date() + "</p>");
        out.println("</body>");
        out.println("</html>");
        out.close();
    }
}
            
```

Methoden für Initialisierung etc. erben
GET-Anfragen verarbeiten
MIME-Typ der folgenden Ausgabe setzen
HTML-Code in Ausgabestream schreiben

Java-Praxiskurs II 4

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS

Ü7.1 Deployment des Servlets

- Starten Sie den J2EE Default Server und das Deployment Tool. Erzeugen Sie eine neue Web Component. Fügen Sie dem `.war`-Archiv das Time-of-Day-Servlet hinzu, geben Sie den Alias an und deployen Sie das Servlet im J2EE-Server.
- Rufen Sie das Servlet im Browser auf und kontrollieren Sie die Ausgabe.

➤ Lösung: Deployment live

Java-Praxiskurs II 5

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS

Wdh.: Sessions und States

- HTTP ist ein zustandsloses Übertragungsprotokoll:
 - Der Server hat an sich keine Möglichkeit, festzustellen, ob zwei Requests vom gleichen Benutzer abgeschickt wurden.
- Problem: Oft muss die Anwendung aber wissen, welcher Benutzer welchen Request abgeschickt hat (**Session**) und mit welchen individuellen Datenstrukturen dieser Benutzer gerade arbeitet (**State**).
 - Beispiel: Nach dem Login weiß die Bibliotheksanwendung auch später beim Ausleihen noch, wer der Benutzer ist, und wonach er zuletzt gesucht hat.
- Lösungsansatz: Der Benutzer muss sich in jedem Request gegenüber der Anwendung identifizieren.
 - Wir müssen also dafür sorgen, dass bei jedem Request eine eindeutige Benutzererkennung mitgeschickt wird.
 - manuelle Implementierung: hoher technischer Aufwand, Fehlerquelle

Java-Praxiskurs II 6

Wdh.: Session Management in Servlets

- Anlegen bzw. Holen einer Session für einen Benutzer mit Methoden von `HttpServletRequest`:


```
public HttpSession getSession()
public HttpSession getSession(boolean create)
```

 - weist dem Benutzer eine neue Session zu (intern über eine Session-ID identifiziert), wenn er noch keine hat oder `create==true` ist
 - holt die `HttpSession`-Instanz, in der alle mit diesem Benutzer assoziierten Daten gespeichert werden
- Alle Servlets haben auf dieses Session-Objekt Zugriff und können darüber benutzerspezifische Daten austauschen.
- Das Session-Objekt existiert so lange, bis
 - es über seine `invalidate`-Methode explizit zerstört wird oder
 - der Benutzer eine bestimmte Zeit lang keine Requests mehr sendet

Java-Praxiskurs II 7

Ü7.2 LoginServlet

- Implementieren Sie das `LoginServlet` und deployen Sie es so, dass es unter dem Alias `/login` angesprochen werden kann. Das `LoginServlet` soll die übergebenen Daten mit den Benutzern aus der Datenbank überprüfen. Wenn Benutzernummer und Passwort nicht zueinander passen, soll das Servlet eine Fehlermeldung und einen Link auf das Login-Formular ausgeben. Im positiven Fall soll das Servlet den Benutzer in der Session speichern und einen Link auf das Hauptmenü ausgeben.
- Quellcode: siehe nächste Folie
- Deployment anschließend live

Java-Praxiskurs II 8

Ü7.2 LoginServlet

```
public class LoginServlet extends HttpServlet {
protected void doPost(...) throws ServletException {
// ... ContentType setzen, PrintWriter holen, HTML-Kopf ausgeben ...
HttpSession session = request.getSession();
String nr = request.getParameter("nr");
String pass = request.getParameter("password");
try {
int nummer = Integer.parseInt(nr);
Benutzer b = Benutzer.findById(nummer);
if ((b!=null)&&b.checkPassword(pass)) {
session.setAttribute("Benutzer", b);
out.println("<p>Sie haben sich erfolgreich angemeldet.</p>");
out.println("<p><a href='\""+
response.encodeURL(\"menu\")+\"\">Weiter zum Hauptmenü</a></p>");
} else {
out.println("<p>Sie konnten nicht angemeldet werden.</p>");
out.println("<p>Bitte überprüfen Sie Benutzernummer und Passwort.</p>");
out.println("<p><a href='\""+
response.encodeURL(\"index.html\")+\"\">Zurück zum Login</a></p>");
}
} catch (NumberFormatException e) {
out.println("<p>Bitte melden Sie sich mit Ihrer Benutzer-<b>nummer</b> an!</p>");
out.println("<p><a href='\""+
response.encodeURL(\"index.html\")+\"\">Zurück zum Login</a></p>");
}
// ... HTML-Code abschließen, PrintWriter schließen ...
}
}
```

POST-Anfragen verarbeiten
Session besorgen
Parameter auslesen und prüfen
Authentifiziertes Benutzer-Objekt in Session ablegen

Java-Praxiskurs II 9

Ü7.2 MenuServlet

- Implementieren Sie ein `MenuServlet` und deployen Sie es so, dass es unter dem Link erreichbar ist, den das `LoginServlet` nach positiver Authentifizierung ausgibt. Im Menü sollen vier Menüpunkte stehen:
 - Bestand durchsuchen
 - Eigene Ausleihen einsehen
 - Eigene Vormerkungen einsehen
 - Abmelden
- Prüfen Sie vor der Ausgabe des Menüs, ob der Benutzer auch wirklich angemeldet ist und geben Sie im negativen Fall einen Hinweis und einen Link auf die Login-Seite aus.
- Quellcode: siehe nächste Folie
- Deployment anschließend live

Java-Praxiskurs II 10

Ü7.2 MenuServlet

```
public class MenuServlet extends HttpServlet {
protected void doGet(...) throws ServletException {
// ... ContentType setzen, PrintWriter holen,
// HTML-Kopf ausgeben ...
HttpSession session = request.getSession();
Benutzer benutzer = (Benutzer)session.getAttribute("Benutzer");
if (benutzer!=null) {
out.println("<ul>");
out.println("<li>Bestand durchsuchen</li>");
out.println("<li>Eigene Ausleihen einsehen</li>");
out.println("<li>Eigene Vormerkungen einsehen</li>");
out.println("<li><a href='\""+response.encodeURL(\"logout\")+
\"\">Abmelden</a></li>");
out.println("</ul>");
} else {
out.println("<p>Sie müssen sich zunächst anmelden!</p>");
out.println("<p><a href='\""+response.encodeURL(\"index.html\")+
\"\">Zurück zum Login</a></p>");
}
// ... HTML-Code abschließen, PrintWriter schließen ...
protected void doPost(...) throws ServletException {
doGet(request, response);
}
}
```

GET-Anfragen verarbeiten
Benutzer-Objekt aus Session holen
POST-Anfragen verarbeiten: Weiterleiten an doGet()

Java-Praxiskurs II 11

Ü7.2 LogoutServlet

- Implementieren Sie ein `LogoutServlet` und verlinken und deployen Sie es so, dass es unter dem Menüpunkt „Abmelden“ des Hauptmenüs zu erreichen ist. Invalidieren Sie die Session und geben Sie dem Benutzer den Hinweis, dass er nun abgemeldet ist.
- Quellcode: siehe nächste Folie
- Deployment anschließend live

Java-Praxiskurs II 12

Ü7.2 LogoutServlet

```

public class LogoutServlet extends HttpServlet {
    protected void doGet(...) throws ... {
        // ... ContentType setzen, PrintWriter holen,
        // HTML-Kopf ausgeben ...
        HttpSession session = request.getSession();
        session.invalidate();
        out.println(
            "<p>Sie haben sich erfolgreich abgemeldet.</p>");
        out.println("<p><a href=\""+
            response.encodeURL("index.html")+
            "\">Erneut anmelden</a></p>");
        // ... HTML-Code abschließen, PrintWriter schließen ...
    }

    protected void doPost(...) throws ... {
        doGet(request, response);
    }
}

```

Session invalidieren

Java-Praxiskurs II 13

Ü7.3 DurchsuchenServlet

- ❖ Implementieren Sie ein **DurchsuchenServlet** und deployen und verlinken Sie es so, dass es im Hauptmenü unter „Bestand durchsuchen“ zu erreichen ist. Prüfen Sie, ob der Benutzer auch wirklich angemeldet ist und geben Sie im negativen Fall einen Hinweis und einen Link auf die Login-Seite aus.
- ❖ Prüfen Sie, ob der Submit-Button für die Suche geklickt wurde. Ist das nicht der Fall, soll das Suchformular angezeigt werden. Wurde das Formular bereits angezeigt, ausgefüllt und der Submit-Button geklickt, soll das Servlet den Buchbestand nach dem eingegebenen Stichwort durchsuchen. Das Ergebnis soll in tabellarischer Form ausgegeben werden.

➤ Lösung: siehe Quellcode

Java-Praxiskurs II 14

Ü7.3 DetailServlet

- ❖ Implementieren Sie ein **DetailServlet** und verlinken und deployen Sie es so, dass es aus der Tabelle der Suchergebnisse zu erreichen ist. Dabei soll es für jedes Suchergebnis einen Link geben, mit dem man die Details zu genau diesem Medium einsehen kann. Prüfen Sie, ob der Benutzer auch wirklich angemeldet ist und geben Sie im negativen Fall einen Hinweis und einen Link auf die Login-Seite aus.

➤ Lösung: siehe Quellcode

Java-Praxiskurs II 15

Ü7.4 ProzessServlet

- ❖ Ergänzen Sie das **DetailServlet** derart, dass abhängig vom Leihstatus eines Mediums die möglichen Aktionen (Ausleihen, Verlängern, Zurückgeben und Vormerken) angezeigt werden.
- ❖ Implementieren Sie dann ein **ProzessServlet** und deployen und verlinken Sie es so, dass es über die angezeigten möglichen Aktionen erreicht werden kann.
- ❖ Je nachdem, welche Aktion ausgewählt wurde, soll das **ProzessServlet** die geeignete(n) Methode(n) auf den Medienobjekten aufrufen und den Benutzer über Erfolg oder Misserfolg der Durchführung informieren.

➤ Lösung: siehe Quellcode

Java-Praxiskurs II 16

Ü7.5 AusleihenEinsehenServlet

- ❖ Implementieren Sie ein **AusleihenEinsehenServlet** und deployen und verlinken Sie es so, dass es im Hauptmenü unter „Eigene Ausleihen einsehen“ erreichbar ist. Dieses Servlet soll alle vom angemeldeten Benutzer entliehenen Medien in tabellarischer Form auflisten. Dabei sind Ausleih- und Rückgabedatum mit auszugeben.
- ❖ Außerdem soll jedes Medium so mit dem **DetailServlet** verlinkt werden, dass man sich Details zu genau diesem Medium ansehen kann. Prüfen Sie, ob der Benutzer auch wirklich angemeldet ist und geben Sie im negativen Fall einen Hinweis und einen Link auf die Login-Seite aus.

➤ Lösung: siehe Quellcode

Java-Praxiskurs II 17

Ü7.6 VormerkungenEinsehenServlet

- ❖ Implementieren Sie ein **VormerkungenEinsehenServlet** und deployen und verlinken Sie es so, dass es im Hauptmenü unter „Eigene Vormerkungen einsehen“ erreichbar ist. Dieses Servlet soll alle vom angemeldeten Benutzer vorgemerkten Medien in tabellarischer Form auflisten. Dabei ist das Datum der Vormerkung mit auszugeben.
- ❖ Außerdem soll jedes Medium so mit dem **DetailServlet** verlinkt werden, dass man sich Details zu genau diesem Medium ansehen kann. Prüfen Sie, ob der Benutzer auch wirklich angemeldet ist und geben Sie im negativen Fall einen Hinweis und einen Link auf die Login-Seite aus.

➤ Lösung: siehe Quellcode

Java-Praxiskurs II 18