10. Aufgabenblatt

Java Servlets

Java Servlets

- serverseitig dynamisch generierte Webseiten
- Klasse mit bestimmte Schnittstelle implementieren und dem (Web-/Application-)Server bekannt machen
- bei Aufruf einer bestimmten URL wird die `service`-Methode des dafür vorgesehen Servlets ausgeführt
- Parameter werden über die URL (GET) oder als Daten (POST) übertragen und können abgefragt werden
  http://bla.mi.fu-berlin.de/myservlet?name=max
- Servlet "antwortet" mit HTML-Code

Java Servlets

- Interaktions-Schema
  - Link führt zu bestimmten Servlet (ein Servlet pro Aktion) oder
  - Parameter steuert auszuführende Aktion
    (Servlet für verschiedene Anfragen)
- Lebenszyklus eines Servlets
  - Objekterzeugung (Servlet) bei Serverstart oder bei Bedarf (init)
  - "lebt" unbestimmte Zeit, wird für ein oder mehrere Anfragen verwendet (service)
  - wird "irgendwann" abgeräumt (destroy)
  - Listener einsetzbar
- Servlet-Container bietet u.a. Session-Management, um Aufrufer zu identifizieren

API

- javax.servlet.Servlet
  - public void service(ServletRequest req, ServletResponse res) { ... }
- javax.servlet.http.HttpServlet implements [...] Servlet
  - public void doGet(HttpServletRequest request,
    HttpServletResponse response) { ... }
- HttpServletRequest bietet unter anderem getSession()

Demo-Servlet

```
package test;

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

public class MyServlet extends HttpServlet {

    int start;

    /** initialisieren */
    public void init () throws ServletException {
        String param = getInitParameter("start");
        if (param != null)
            start = Integer.parseInt(param);
    }
```

Anfrage bearbeiten: Zustand/Sessionmanagement

```
/** GET behanden */
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{

    // Zustand/Session (vor dem PrintWriter!)
    HttpSession session = request.getSession(false);
    if (session == null) {
        session = request.getSession(true);
        session.setMaxInactiveInterval(10); // sec
    }
    MyState state;
    Object o = session.getAttribute("state");
    if (o != null) {
        state = (MyState) o;
    } else {
        state = new MyState(start);
        session.setAttribute("state", state);
    }
```

```
public class MyState {
    public int val;
    public MyState(int x) {
        val = x;
    }
}
```

Zustandsänderung und Antwort

```
    // ggf. Aktion durchführen
    String action = request.getParameter("action");
    if ("inc".equals(action))           state.val++;
    else if ("reset".equals(action))  state.val = 0;

    // Antwort-Typ setzen (vor dem Schreiben!)
    response.setContentType("text/html");

    // HTML-Seite ausgeben
    PrintWriter pw = response.getWriter();
    pw.println("<html>\n<head>\n<title>Adventure</title>\n</head>");
    pw.println("<body>");

    pw.println("mein Servlet, Zustand: " + state.val + "<br><br>");
    String link = response.encodeURL(request.getContextPath() + "?action=inc");
    pw.print("<a href=\"" + link + "\">inc</a> ");
    link = response.encodeURL(request.getContextPath() + "?action=reset");
    pw.print("<a href=\"" + link + "\">reset</a> ");

    pw.println("</body>\n</html>");
    pw.flush();
}
```

weitere Standardmethoden

```
    /** Servlet wird abgeräumt */
    public void destroy() {
        System.out.println("Mein Servlet wurde zerstört");
    }

    /** Info */
    public String getServletInfo() {
        return "Dies ist mein Servlet";
    }
}
```

Jetty

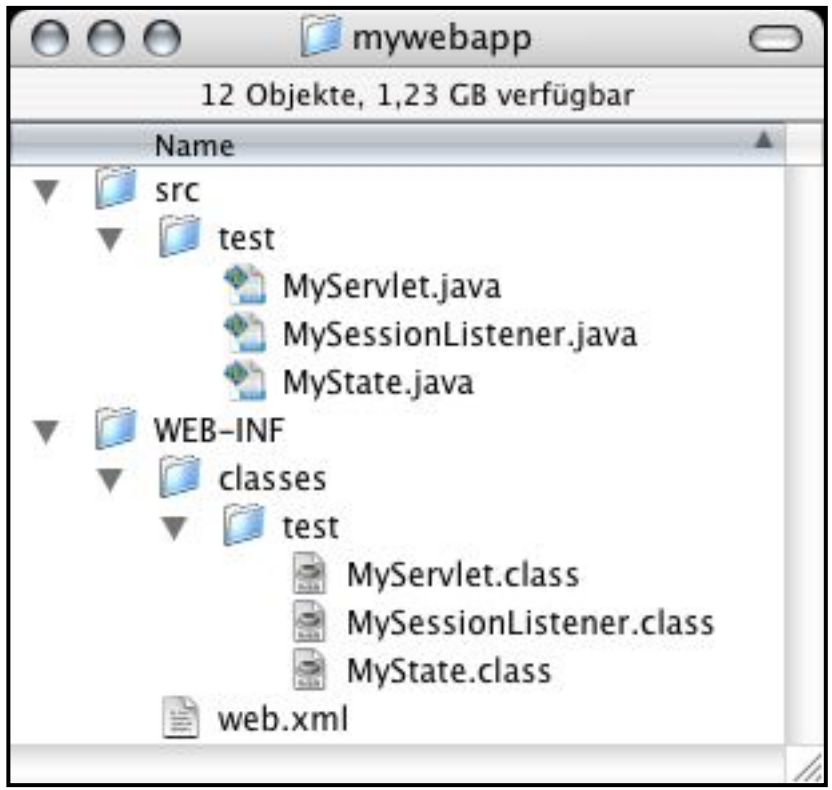
- Open Source Java HTTP Server und Servlet Container
- http://jetty.mortbay.com/jetty/index.html
- Einbinden in eigene Anwendung oder eigenständiger Server

- Start im jetty-Verzeichnis:
  java -jar start.jar myconfig.xml
  (Übung (CORBA): **jaco** -jar start.jar myconfig.xml)

myconfig.xml

```
<?xml version="1.0"  encoding="UTF-8"?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN"
"http://jetty.mortbay.org/configure.dtd">

Configure class="org.mortbay.jetty.Server">

  <Call name="addListener">
    <Arg>
      <New class="org.mortbay.http.SocketListener">
        <Set name="Port"><SystemProperty name="jetty.port" default="8080"/></Set>
      </New>
    </Arg>
  </Call>

  <Call name="addWebApplication">
    <Arg>/</Arg>
    <Arg><SystemProperty name="jetty.home" default="."/>/mywebapp/</Arg>
  </Call>
 </Configure>
```

mywebapp-Ordner



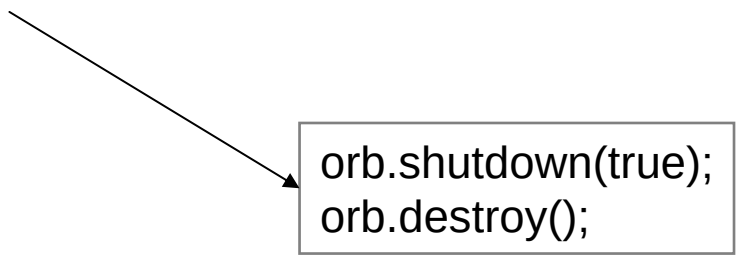
web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app
   xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" version="2.4">

  <display-name>My Web App</display-name>

  <listener>
    <listener-class>test.MySessionListener</listener-class>
  </listener>

  <servlet>
    <servlet-name>My Servlet</servlet-name>
    <servlet-class>test.MyServlet</servlet-class>
    <init-param>
        <param-name>start</param-name>
        <param-value>27</param-value>
    </init-param>
    <load-on-startup>0</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>My Servlet</servlet-name>
    <url-pattern>/myservlet/*</url-pattern>
  </servlet-mapping>

</web-app>
```

http://bla.mi.fu-berlin.de:8080/myservlet

MySessionListener

```
import javax.servlet.ServletException;
import javax.servlet.http.*;

public class MySessionListener implements HttpSessionListener {
    public void sessionCreated(HttpSessionEvent se) {
        [...]
    }
    public void sessionDestroyed(HttpSessionEvent se) {
        Object o = se.getSession().getAttribute("state");
            [Ressourcen freigeben]
    }
}
```



Hinweis: Übung 10

- ORB:
  init(java.lang.String[] args, java.util.Properties props) verwenden, nicht init() !
  z.B. ORB orb = ORB.init(new String[0], null);