

Web-Technologien Kick-Start

Software-Praktikum im Grundstudium
WS 2004/2005

Dipl.-Inform. Michael Kirchhof
Dipl.-Inform. Bodo Kraft
Prof. Dr.-Ing. Manfred Nagl

Department of Computer Science III
Software Engineering
Ahornstr. 55
52074 Aachen, Germany

<http://www-i3.informatik.rwth-aachen.de>

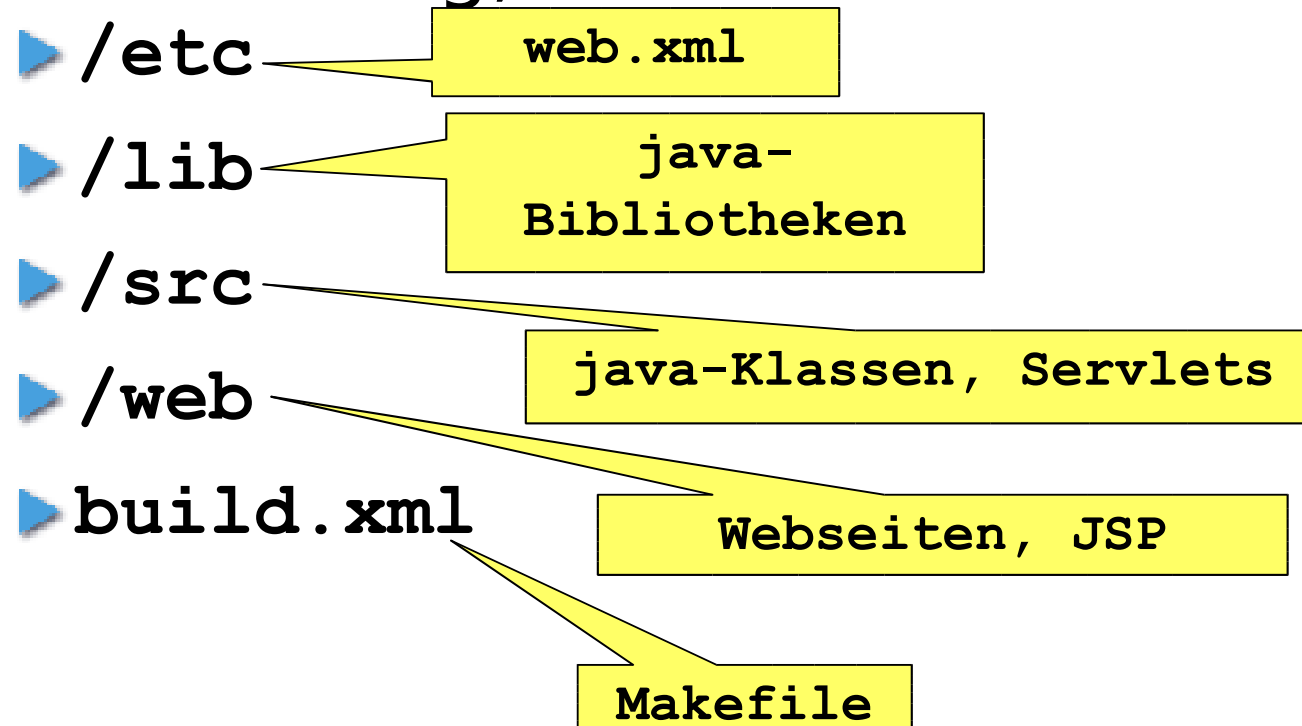
WS 2004/2005

Agenda

- ▶ Struktur von Web-Applikationen
- ▶ Compile & Deploy
- ▶ Ant
- ▶ Beliebte Kommandos

WAR-Verzeichnisstruktur

- ▶ ein Verzeichnis für jede Web-Anwendung, darin enthalten:



In `build.xml` `name` und `app.name` setzen
Übersetzung und Installation im Tomcat mit

```
ant -Dtomcat.home=$TOMCAT_HOME_LOCAL
```

Tomcat-Anwendungen: web.xml

```
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
    2.2//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
```

```
<web-app>
  <!-- General description of your web application -->
  <display-name>HelloWorld</display-name>
  <description>
    Nur eine kurze Beschreibung.
  </description>
  <context-param>
    <param-name>webmaster</param-name>
    <param-value>sbecker@i3.informatik.rwth-aachen.de
    </param-value>
    <description>
      The EMAIL address of the administrator
    </description>
  </context-param>
```

Einstellungen für die Web-Applikation

Tomcat-Anwendungen: web.xml

...

```
<servlet>
```

Servlet anmelden

```
  <servlet-name>HelloWorld</servlet-name>
```

```
  <description>
```

Beschreibung

```
  </description>
```

```
  <servlet-class>HelloWorld</servlet-class>
```

```
</servlet>
```

Name der Java-Klasse

```
<servlet-mapping>
```

```
  <servlet-name>HelloWorld</servlet-name>
```

```
  <url-pattern>/HelloWorld</url-pattern>
```

```
</servlet-mapping>
```

Zuordnung URL zu Servlet
(via internen Name)

```
</web-app>
```

Immer erst alle <servlet>- , dann alle <servlet-mapping>-Elemente

Entwicklung v. Web-Anwendungen

▶ Ohne Tools:

- ▶ Servlets, JSPs & Co. implementieren
- ▶ kompilieren
- ▶ binden (als Web-App., d.h. WAR)
- ▶ Kompilate (+ x) kopieren (Deployment)
- ▶ Tomcat starten
- ▶ (auf zig Fehler stoßen, weil Dateien entweder fehlen oder veraltet sind)
- ▶ testen
- ▶ fertig!

Entwicklung v. Web-Anwendungen

- ▶ Mit Tools:
 - ▶ Servlets, JSPs & Co. mit IDE implementieren
 - ▶ Syntax-Fehler mit Hilfe der IDE ausmerzen
 - ▶ ant aufrufen
 - ▶ Tomcat starten
 - ▶ testen
 - ▶ fertig!

Ant

- ▶ Tool zum Übersetzen, Binden und Installieren von komplexen Anwendungen
- ▶ Sehr gut geeignet für Java und Web-Anwendungen
- ▶ Steuerung durch XML-Datei
- ▶ Weitere Infos: <http://ant.apache.org>

Ein Beispiel: build.xml

```
<project name="MyProject" default="dist" basedir=". ">
  <description>
    simple example build file
  </description>
  <!-- set global properties for this build -->
  <property name="src" location="src"/>
  <property name="build" location="build"/>
  <property name="dist" location="dist"/>

  <target name="init">
    <!-- Create the time stamp -->
    <tstamp/>
    <!-- Create the build directory structure used by compile -->
    <mkdir dir="${build}"/>
  </target>

  <target name="compile" depends="init"
    description="compile the source " >
    <!-- Compile the java code from ${src} into ${build} -->
    <javac srcdir="${src}" destdir="${build}"/>
  </target>
```

Ein Beispiel: build.xml (2)

...

```
<target name="dist" depends="compile"
    description="generate the distribution" >
    <!-- Create the distribution directory -->
    <mkdir dir="${dist}/lib"/>

    <!-- Put everything in ${build} into the
        MyProject-${DSTAMP}.jar file -->
    <jar jarfile="${dist}/lib/MyProject-${DSTAMP}.jar"
        basedir="${build}"/>
</target>

<target name="clean" description="clean up" >
    <!-- Delete the ${build} and ${dist} directory trees -->
    <delete dir="${build}"/>
    <delete dir="${dist}"/>
</target>
</project>
```

Beliebte Kommandos

▶ **tom**

- ▶ Tomcat starten

▶ **jerry**

- ▶ Tomcat beenden

▶ **spike**

- ▶ Tomcat killen, temporäre Tomcat-Dateien löschen (reset)

▶ **/rbi/eclipse/eclipse -data ...**

- ▶ Eclipse (IDE) starten