

## Advanced Internet Technologies

Java im Webumfeld,  
Tomcat Applikationsserver

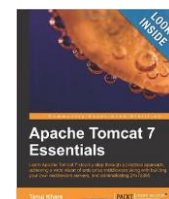
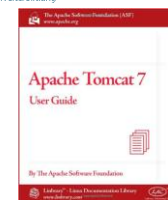
FH Zentralschweiz

## Buch und Literaturreferenz

- Bücher:
  - Apache Tomcat 7 User Guide, Fultus Corporation  
ISBN-13: 978-1596822719
  - Apache Tomcat 7 Essentials, Packt Publishing  
ISBN-13: 978-1849516624

- Links:
  - <http://tomcat.apache.org/>
  - [http://www.ntu.edu.sg/home/ehchua/programming/howto/Tomcat\\_More.html](http://www.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_More.html)
  - <http://www.coreservlets.com/Apache-Tomcat-Tutorial/tomcat-7-with-eclipse.html>

Hochschule Luzern  
Technik & Architektur  
Weiterbildung



## Ziele

- Hintergründe und Zusammenhänge der sogenannten **Web Application Server** kennen
- Sehen wie ein Web Application Server, am Beispiel des Tomcat7, installiert, konfiguriert und betrieben bzw. mit Web Applikationen bestückt wird
- Eine eigenes Servlet zu programmieren und anschliessend auf dem Tomcat7 zum Laufen zu bringen

Folie 3, 6. November 2013



## Was ist Tomcat?

- Der Tomcat Server ist ein Java-basierender Web Application Container der Servlets oder Java Server Pages (JSP) Web Applikationen laufen lässt
- Tomcat wurde zur Referenz-Implementation sowohl für die Java Servlets als auch für die Java Server Pages Spezifikation
- Tomcat wurde im Umfeld des Apache Webservers entwickelt:

<http://tomcat.apache.org/>



Folie 4, 6. November 2013

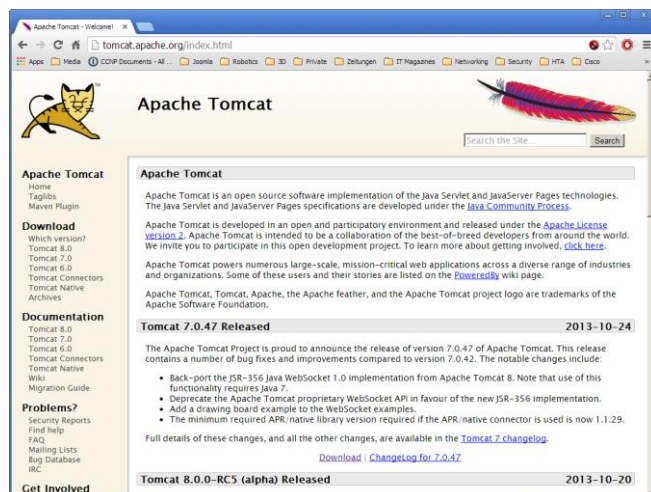
## Was ist Tomcat?

- Apache Tomcat wird verwendet um Java Servlets und Java Server Pages bereit zu stellen
- Tomcat ist ein komplexes Stück Software, es ist hilfreich wenn man die Dokumentation "zur Hand" hat (<http://tomcat.apache.org/tomcat-7.0-doc/index.html>)
- Tomcat kann als Standalone-Produkt verwendet oder kann zusammen mit dem Apache-Server verwendet werden

Folie

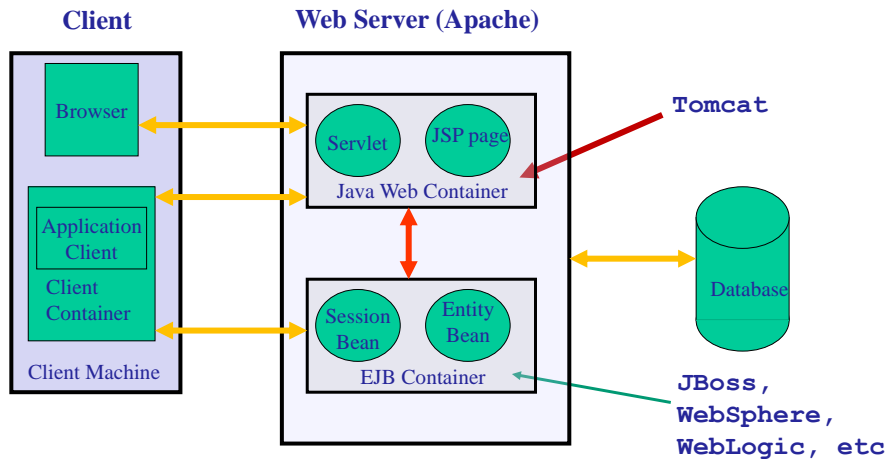
## Was ist Tomcat?

- Aktuell ist Version 7
- Version 8 hat noch Alpha Status



Folie 6, 6. November 2013

## Generelle Übersicht Requests/Responses



Folie 7, 6. November 2013

## Was ist eine Web Applikation?

- Web Applikationen sind zusammengesetzt aus klassischen Web-Komponenten wie HTML/CSS Seiten. Dazu kommen nun weitere Komponenten wie Servlets, JSP Seiten erzeugt durch mit der JavaServerPages Technologie, Web Filter und Web Event Listener.
- Diese Komponenten werden auf einem Web Server ausgeführt und antworten auf HTTP Requests von Web Clients (Browser, spez. Clientprogramme, etc.)
- Servlets, JSP Seiten und Filter können verwendet werden um HTML Seiten zu erzeugen, welche das Application User Interface bilden.

Folie 8, 6. November 2013

## Was ist ein Web/Servlet Container?

- Eine Web Applikation läuft innerhalb eines Web Containers. Dieser Container stellt die benötigte Runtime-Umgebung für den entsprechenden Namensraum und Lifecycle Management zur Verfügung.
- Einige Webserver stellen zusätzliche Services wie Security oder Nebenläufigkeits-Steuerung zur Verfügung.

Folie 9, 6. November 2013

## Was ist ein Servlet?

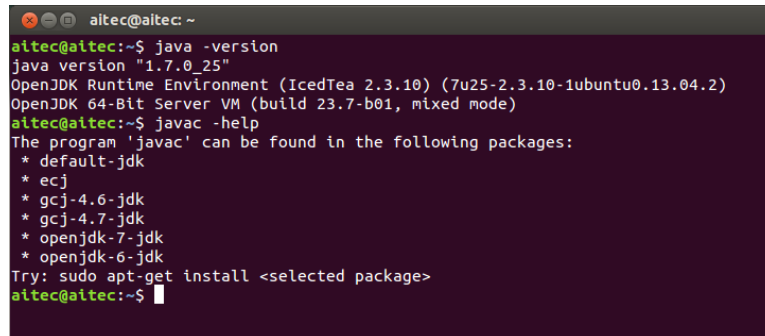
- Servlets sind serverseitige Java Programme welche die Funktionalität des Web Servers massiv erweitern.
- Servlets sind Java Klassen welche dynamisch die Requests verarbeiten und eine Response erzeugen unabhängig vom Protokoll!
- Servlets sind definiert in der Java Servlet API specification; (<http://tomcat.apache.org/tomcat-7.0-doc/servletapi/index.html>)
- Servlets sind prinzipiell **protokoll-unabhängig**, das bedeutet, jedes Protokoll kann dazu verwendet werden. Jedoch verarbeiten in der Praxis Servlets meistens HTTP Requests und erzeugen die HTML Response.

Folie 10, 6. November 2013

## Prüfen der Installation

```
#java -version
```

```
#javac -help
```



```
altec@altec: ~  
altec@altec:~$ java -version  
java version "1.7.0_25"  
OpenJDK Runtime Environment (IcedTea 2.3.10) (7u25-2.3.10-1ubuntu0.13.04.2)  
OpenJDK 64-Bit Server VM (build 23.7-b01, mixed mode)  
altec@altec:~$ javac -help  
The program 'javac' can be found in the following packages:  
* default-jdk  
* ecj  
* gcj-4.6-jdk  
* gcj-4.7-jdk  
* openjdk-7-jdk  
* openjdk-6-jdk  
Try: sudo apt-get install <selected package>  
altec@altec:~$
```

(hier fehlt offensichtlich der Java Compiler)

Folie 11, 6. November 2013

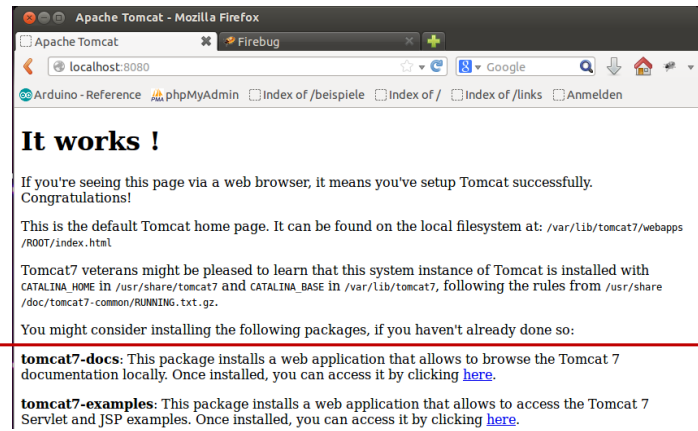
## Falls nicht ok:

- **Step1:** Voraussetzungen schaffen
  - #sudo apt-get install java
- **Step2** Download der Tomcat Packages, z.B. von <http://mirror.switch.ch/mirror/apache/dist/tomcat/tomcat-7/v7.0.47/> und entpacken der Sourcen entpacken
  - #tar xvfz apache-tomcat-7.0.47.tar.gz
  - oder
  - #sudo apt-get install tomcat7
  - #sudo apt-get install tomcat7-docs
  - #sudo apt-get install tomcat7-examples
  - #sudo apt-get install tomcat7-admin
- **Step3:** Setup der Pfade für Catalina (System und Versionsabhängig!)
  - #export CATALINA\_HOME=/usr/share/tomcat7
  - #export CATALINA\_BASE=/var/lib/tomcat7

Folie 12, 6. November 2013

## Mal schauen ob der Tomcat läuft?

- Zugreifen via: <http://localhost:8080>



Folie 13, 6. November 2013

## und sonst Starten und Stoppen



### - Starten:

- `#!/usr/share/tomcat7/bin/startup.sh`  
startet den Tomcat Server



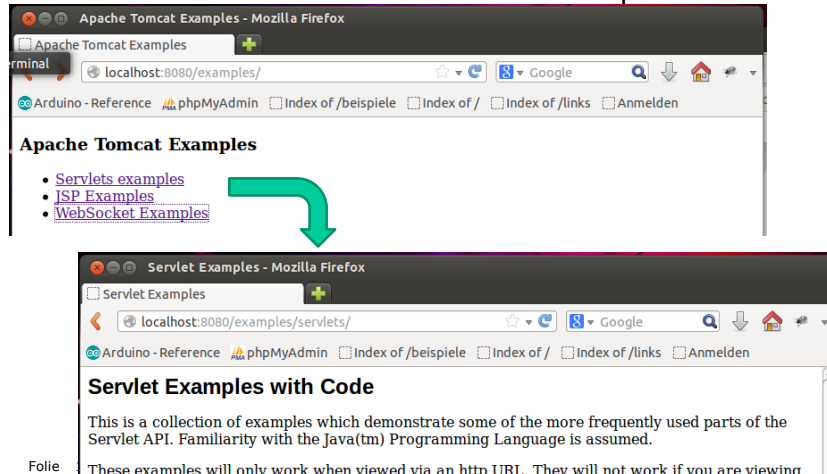
### - Stoppen:

- `#!/usr/share/tomcat7/bin/shutdown.sh`  
beendet den Tomcat Server

Folie 14, 6. November 2013

## Ok, mal Testen der Installation (1)

- Aufrufen der installierten Servlets-Beispiele:



**Apache Tomcat Examples**

- [Servlets examples](#)
- [JSP Examples](#)
- [WebSocket Examples](#)

**Servlet Examples with Code**

This is a collection of examples which demonstrate some of the more frequently used parts of the Servlet API. Familiarity with the Java(tm) Programming Language is assumed.

Folie 15

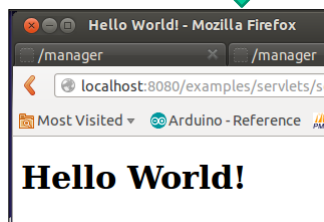
## Ok, mal Testen der Installation (2)

Tip: To see the cookie interactions with your browser, try turning on the "notify when setting a cookie" option in your browser preferences. This will let you see when a session is created and give some feedback when looking at the cookie demo.

Hello World

 [Execute](#)

 [Source](#)



**Hello World!**

### Source Code for HelloWorld Example

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

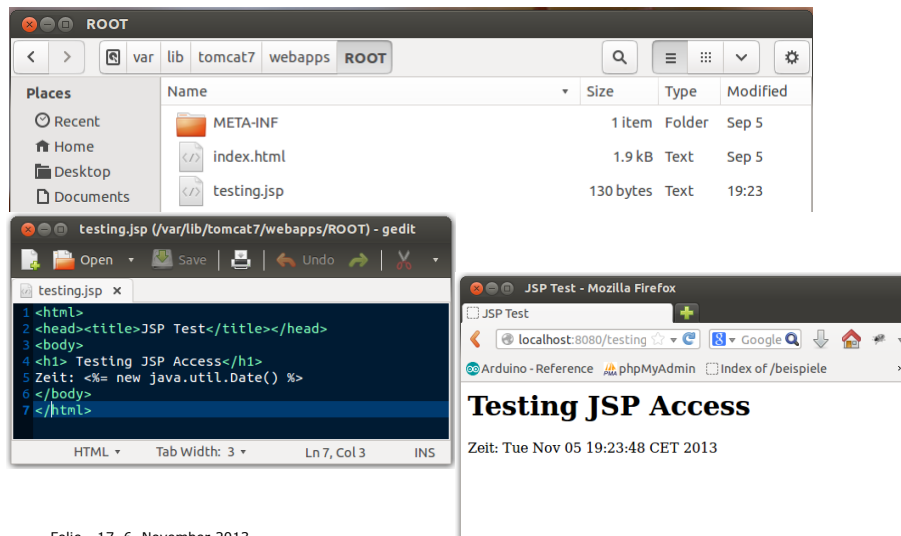
public class HelloWorld extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

Folie 16, 6. November 2013



## Erster eigener JSP Test



Folie 17, 6. November 2013

## Erster Servlet Test

- zuerst JAVA JDK runterladen:  
`#sudo apt-get install openjdk-7-jdk`
- anschliessend noch Environment Variabeln setzen:  
`#export`  
`CLASSPATH=/usr/share/tomcat7/lib/servlet-api.jar:/usr/share/tomcat7/lib/jsp-api.jar`

Folie 18, 6. November 2013

## Zuerst mal ein Servlet programmieren

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

public class TestingServlet extends HttpServlet {

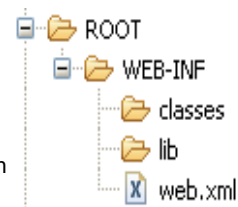
    public void doGet (HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
        out.println("<html>");
        out.println("<head><title>TestingServlet</title><head>");
        out.println("<body>");
        out.println("<h1>Testing Servlet</h1>");
        out.println("</body></html>");
    }
}
```

übersetzen mit :  
#javac TestingServlet.java  
→ man erhält ein .class File

Folie 19, 6. November 2013

## Struktur einer Web Applikation

- Web Applikationen verwenden eine in den Servlet Spezifikationen festgelgte **Verzeichnis Struktur**
- Beim Entwicklen von Web Applikationen für J2EE Plattformen **muss man sich daran halten**, damit man diese auf allen kompatiblen Web Applikationsservern deployen kann!
- Das ROOT Verzeichnis wird als "document root" bezeichnet. Es wird zum "context path" gemapped
- Dieses (private) ROOT Verzeichnis enthält ein weiteres Verzeichnis namens "WEB-INF". Alles innerhalb des ROOT Verzeichnisses, ausser diesem WEB-INF Verzeichnis, ist durch eine URL im Browser öffentlich zugreifbar.
- Web Container werden den Inhalt dieses Verzeichnisses nicht "herausrücken". Jedoch ist der Inhalt des "WEB-INF" Verzeichnisses zugreifbar für die Klassen der Web Applikation.



Folie 20, 6. November 2013

## WEB-INF Verzeichnis

- WEB-INF Verzeichnis beinhaltet:
  - ✓WEB-INF/web.xml Deployment Descriptor
  - ✓WEB-INF/classes Verzeichnis
  - ✓WEB-INF/lib Verzeichnis
- CLASSES - Verzeichnis wird zur Speicherung der kompilierten Servlet- und andere Klasse verwendet.
- LIB - Verzeichnis wird zur Speicherung der jar Dateien verwendet. Falls Applikationen irgendwelche gebundenen jar Dateien oder Third-Party Bibliotheken verwenden müssen diese hier abgelegt werden.
- Alle Klassen und Ressourcen im WEB-INF/classes Verzeichnis plus Klassen und jar Dateien im WEB-INF/lib Verzeichnis sind sichtbar für die umgebende Web Applikation.

Folie 21, 6. November 2013

## Man braucht noch einen Einstieg

- Eine HTML Datei wo man auf Knopfdruck das Servlet aufrufen kann

```
<html>
<head>
  <title>Testing Servlet Beispiel</title>
</head>
<body>
  <h1>Test eines einfachen Servlets</h1>
  <form method="GET" action="/TestingServlet">
    <input type="submit" value="Teste Servlet"/>
  </form>
</body>
</html>
```

- Diese Datei kommt ins **beispiel/pages** Verzeichnis

Folie 22, 6. November 2013

## Mapping erzeugen

- Um das Servlet auf dem Web-Server nutzen zu können, müssen **Metainformationen** bereitgestellt werden.
- Diese **Metainformationen** werden hinterlegt in einer XML-Datei namens **web.xml**, dem sogenannten **Deployment Descriptor**.
- Anschliessend wird daraus zusammen mit der Servlet Klasse (und ev. weiteren benötigten Klassen und Dateien) eine **Archiv-Datei** erzeugt (.war -Datei).
- Diese "Web-Archiv" genannte Datei wird dem Servlet-Container via Web Application Manager übergeben (deployed) oder reinkopiert.
- Zur Laufzeit greift der Webserver auf den Servlet-Container zu, der wiederum eine Instanz des Servlets erzeugt und die entsprechende Methode startet.

Folie 23, 6. November 2013

## Beispiel einer web.xml

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0">

  <display-name>Tomcat Servlet Testing App</display-name>
  <description>
    A small Servlet Testing App
  </description>

  <servlet>
    <servlet-name>TestingServlet</servlet-name>
    <servlet-class>TestingServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>TestingServlet</servlet-name>
    <url-pattern>/TestingServlet</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>/pages/TestingServlet.html</welcome-file>
  </welcome-file-list>
</web-app>
```

- Diese web.xml wird im **beispiel/WEB-INF** Verzeichnis abgelegt

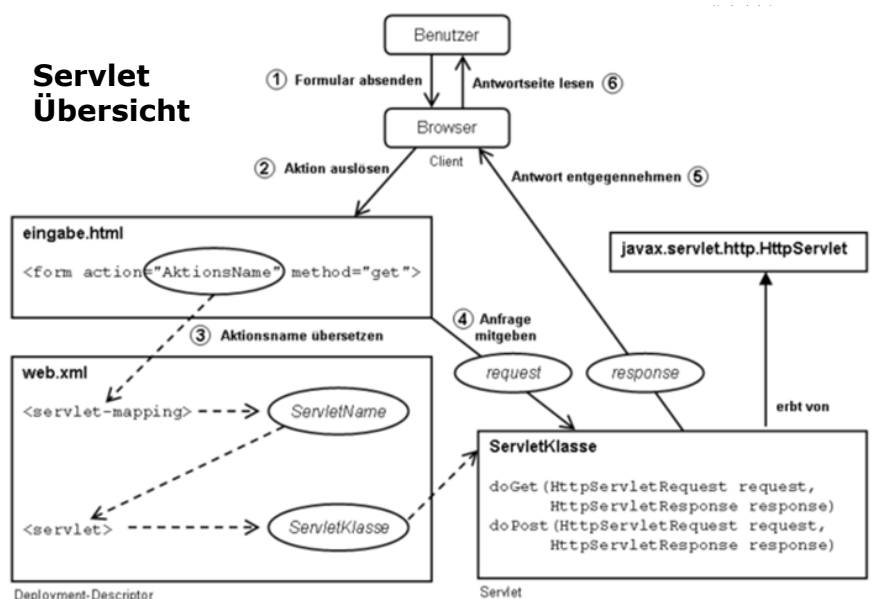
Folie 24, 6. November 2013

## Jetzt noch die .war Datei erzeugen

- Zuerst die vorher erzeugte Java Klasse am richtigen Ort speichern:  
`# cp TestingServlet.class beispiel/WEB-INF/classes`
- Nun die .war Datei erzeugen:  
`# cd beispiel`  
`# jar cvf TestingServlet.war *`

Folie 25, 6. November 2013

## Servlet Übersicht



(Quelle: <http://de.wikipedia.org/wiki/Servlet>)

Folie 26, 6. November 2013

## Servlet Übersicht

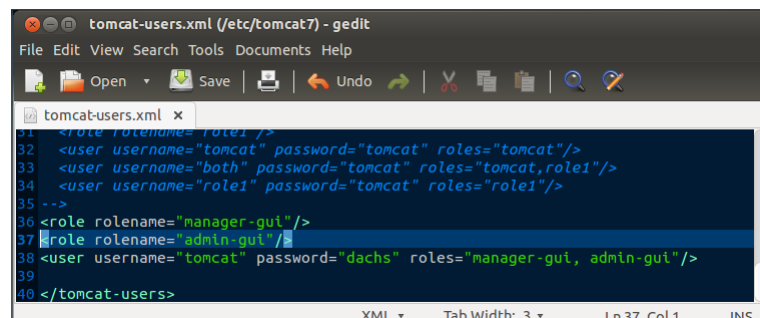
1. Der Benutzer sendet ein ausgefülltes Formular im Browser ab.
2. Der Browser löst die damit verbundene Aktion aus, indem er die Formulardaten an den Webserver sendet.
3. Der Webserver übersetzt den Aktionsnamen des Formulars in den Namen einer **Servlet Klasse**. Er verwendet dazu Informationen aus dem **Deployment-Deskriptor**, einer Datei namens "web.xml".
4. Der Webserver ruft die Methode **request** des Servlets auf und übergibt dabei die Anfragedaten ("request") als Parameter. Die Methode **response** erzeugt eine Ausgabe, die der Webserver an den Browser sendet.
5. Der Browser nimmt die Antwort entgegen und stellt sie dar.
6. Der Benutzer liest die Antwort im Browser.

Folie 27, 6. November 2013

## Tomcat Manager konfigurieren

- User erzeugen in: `/etc/tomcat7/tomcat-users.xml`

- Achtung:  
Editor mit `#sudo gedit` starten (root!)



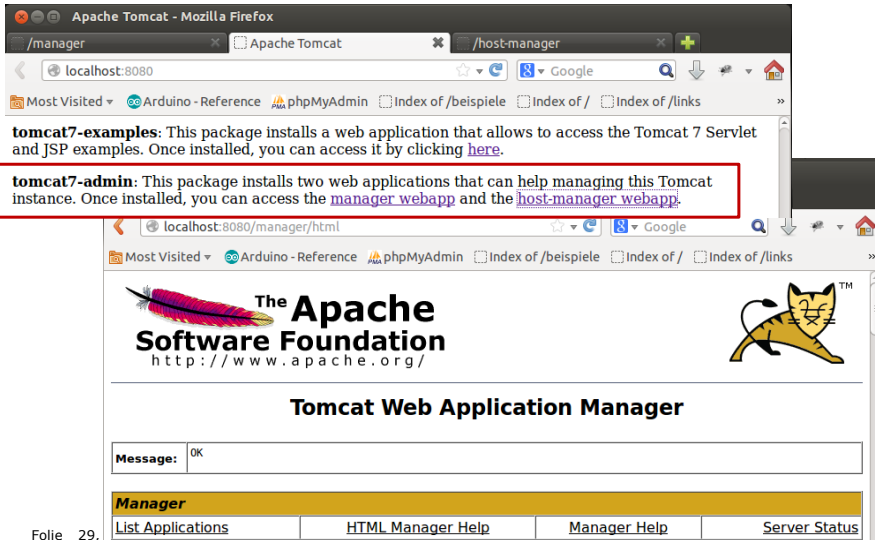
```
tomcat-users.xml (/etc/tomcat7) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Redo
tomcat-users.xml x
31 <role rolename="role1"/>
32 <user username="tomcat" password="tomcat" roles="tomcat"/>
33 <user username="both" password="tomcat" roles="tomcat,role1"/>
34 <user username="role1" password="tomcat" roles="role1"/>
35 -.->
36 <role rolename="manager-gui"/>
37 <role rolename="admin-gui"/>
38 <user username="tomcat" password="dachs" roles="manager-gui, admin-gui"/>
39
40 </tomcat-users>
```

Folie 28, 6. November 2013

## Zugriff auf Manager (nach Login)

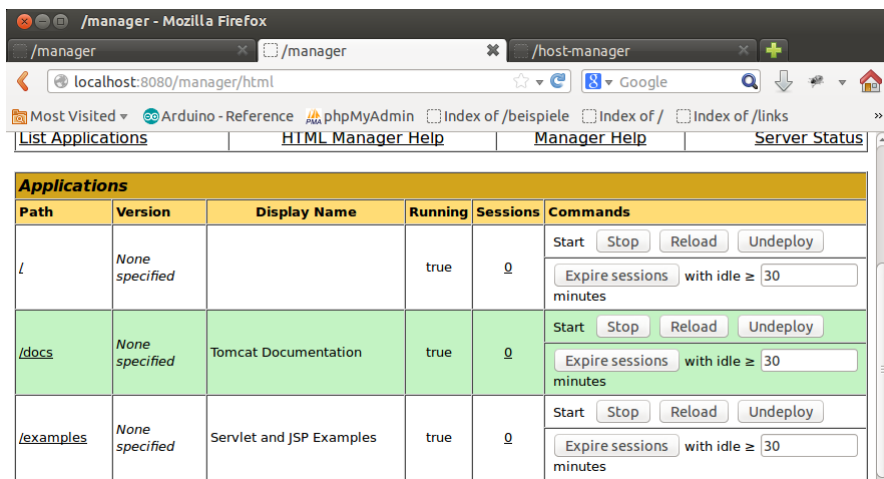
tomcat7-examples: This package installs a web application that allows to access the Tomcat 7 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

tomcat7-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).



Folie 29,

## Verwalten der Applikationen



Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Folie 30, 6. November 2013

## Deployment einer Applikation

The screenshot shows the Tomcat Manager web interface in Mozilla Firefox. The browser address bar shows `localhost:8080/manager/html/upload?org.apache.catalina.filters.i`. The page has a menu bar with 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. Below the menu bar is a 'Most Visited' section with links to 'Arduino - Reference', 'phpMyAdmin', 'Index of /beispiele', 'Index of /', 'Index of /links', and 'Anmelden'. The main content area is titled 'Deploy' and contains two sections: 'Deploy directory or WAR file located on server' and 'WAR file to deploy'. The first section has input fields for 'Context Path (required)', 'XML Configuration file URL', and 'WAR or Directory URL', followed by a 'Deploy' button. The second section, 'WAR file to deploy', is highlighted with a red box and contains a 'Select WAR file to upload' label, a 'Browse...' button, the filename 'testingservlet.war', and another 'Deploy' button.

Folie 31, 6. November 2013

## Deployment kontrollieren

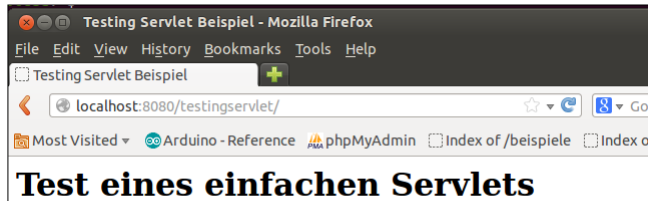
The screenshot shows the Tomcat Manager web interface in Mozilla Firefox. The browser address bar shows `localhost:8080/manager/html/upload?org.apache.catalina.filters.i`. The page has a menu bar with 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. Below the menu bar is a 'Most Visited' section with links to 'Arduino - Reference', 'phpMyAdmin', 'Index of /beispiele', 'Index of /', 'Index of /links', and 'Anmelden'. The main content area is a table listing deployed applications. The table has columns for 'Context Path', 'Name', 'Description', 'Status', 'Version', and 'Actions'. The 'testingservlet' application is highlighted with a red box. Below the table is a 'Deploy' section with input fields for 'Context Path (required)', 'XML Configuration file URL', and 'WAR or Directory URL', followed by a 'Deploy' button.

Context Path	Name	Description	Status	Version	Actions
/manager	None specified	Tomcat Manager Application	true	2	Expire sessions with idle ≥ 30 minutes
/servletexample	None specified	Tomcat Welcome Servlet	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/testingservlet	None specified	Tomcat Servlet Testing App	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

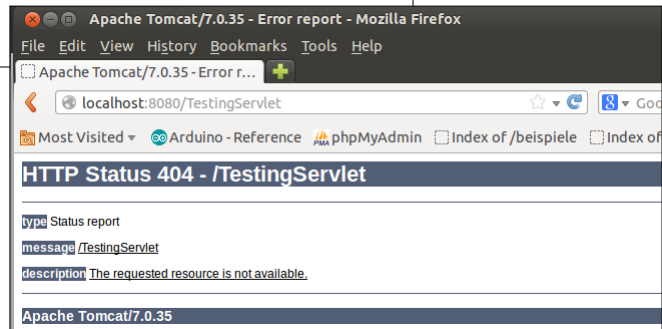
Folie 32, 6. November 2013



## Web Applikation aufrufen



Und Fehlersuchen  
wenn es nicht geht  
😊



Folie 33, 6. November 2013

## Weitere Informationen, Links und Beispiele

- <https://www.digitalocean.com/community/articles/how-to-install-apache-tomcat-on-ubuntu-12-04>
- <http://www.youtube.com/watch?v=6-yaGFK9Ng>
- <http://www.frankdopatka.de/veranstaltungen/informatik-2/installation-tomcat7.pdf>
- <http://tutorials.jenkov.com/java-servlets/index.html>
- <http://de.wikipedia.org/wiki/Servlet>
- <http://de.wikipedia.org/wiki/Web-Archiv>
- <http://www.nogoodatcoding.com/howto/deploy-a-servlet-on-tomcat>
- <http://www.heimetli.ch/ffh/tomcat.html>
- <http://a-weinert.de/pub/tomcat-tips.pdf>

Folie 34, 6. November 2013