

# Apache Tomcat



Rechner und Netzarchitektur  
SS 2003

Johannes Jabornig  
Daniel Peintner

---

## Inhalt

---

- Einleitung
  - Was sind Servlets und JSP
  - Vorteile
- Architektur
  - Catalina
  - Jasper
  - Konnektoren
- Installation / Konfiguration
- Integration in Apache
- Ausblick
- Beispiele
  - Servlets
  - JSP

- Zu Beginn des Internets wenig Interaktion durch statische Webseiten
- Dynamische Webseiten wurden erforderlich
  - Steigerung der Interaktivität
  - Möglichkeiten zur automatischen Generierung von HTML Code wurden gesucht
  - Datenbankverbindungen, ...
- Verschiedene Lösungen durch ...
  - CGI (Common Gateway Interface)
    - Bei jeder Anfrage an den Webserver wird ein neuer Prozess gestartet → daher langsam
  - Perl
  - PHP
  - ASP
  - **Servlets & JSP**

## Servlets

---

- Es handelt sich hierbei um eine Servererweiterung – eine JAVA Klasse die dynamisch geladen werden kann
  - Im Unterschied zu Applets keine Browsererweiterung notwendig
  - Laufen innerhalb einer JVM
- Servlets arbeiten mit allen wichtigen Webserver zusammen
  - Apache
  - IIS
  - ...
- Referenzimplementierung der SUN-Servletspezifikation ist der Tomcat-Server
  - Open-Source-Lizenz

- Effizienz & Leistungsfähigkeit
  - Im Gegensatz zu CGI muss kein Prozess gestartet werden und eine Interpreter aufgerufen werden
  - Ist ein Servlet geladen bleibt es als einzelne Objektinstanz im Speicher → leichtgewichtiger Methodenaufruf
  - Mächtigkeit der Java-API
- Erweiterbarkeit und Flexibilität
  - API leicht erweiterbar
  - Flexibilität durch einfache *out.println* oder Templates
- Sicherheit
  - Strenge Typensicherheit der Sprache Java
  - Exceptionhandling
- Integration
  - Zusammenarbeit mit dem Webserver
  - Möglichkeiten vom Server nutzen (Autorisierung, Dateipfadübersetzung, ...)
- Einfachheit
  - Klassenbibliotheken für den Zugriff auf Formularfelder, Session-Tracking oder Cookies
  - Java bietet die Infrastruktur für parallele Programme mit Threads
  - Einfache Möglichkeit auf Datenbanken mittels JDBC zuzugreifen
- Portabilität

## Java Server Pages (JSP)

---

- Im Gegensatz zum Servlet wird bei einer Java Server Page (JSP) vom Benutzer keine komplette JAVA-Anwendung geschrieben
  - JAVA-Code wird in eine HTML-Datei inkludiert
  - Der Server weiß JSP von normalen HTML-Seiten zu unterscheiden und compiliert mit Hilfe eines JSP-Übersetzers daraus ein Servlet
  - Prinzipiell könnten JSP auch andere Programmiersprachen einbetten
  - Der Übersetzungsvorgang von JSP in ein Servlet muss dann nur einmal getätigt werden, danach benutzt der Servlet-Container direkt die übersetzte Klasse
- Der JSP-Übersetzer ist ein Schwachpunkt, denn hier muss ein Compiler eingebunden werden
  - Tomcat löst das Problem, in dem der Sun-Compiler in tools.jar verwendet wird
  - Andere Hersteller müssen hier eigene Übersetzer einbinden
  - Das macht es natürlich für nicht Compilerhersteller schwierig eine Servlet/JSP-Umgebung anzubieten.
- Neben dem Inhalt der HTML-Datei (der Template genannt wird) lassen sich drei unterschiedliche JSP-Konstrukte in eine Seite einbinden
  - `<% .. %>`  
Eingebettete Java-Stücke (Scriptlets) besitzen dieses Format  
Hier lassen sich beliebige Deklarationen tätigen und Ausgaben machen
  - `<%= .. %>`  
Der Ausdruck wird in die Seite eingebaut

- Tomcat definiert zwei Teilprojekte mit den Namen
  - Catalina  
... ist für Servlets zuständig
  - Jasper  
... für Java Server Pages
- Bei der Installation sind beide Teile aktiv

## Tomcat Architektur – Catalina (1/3)

---

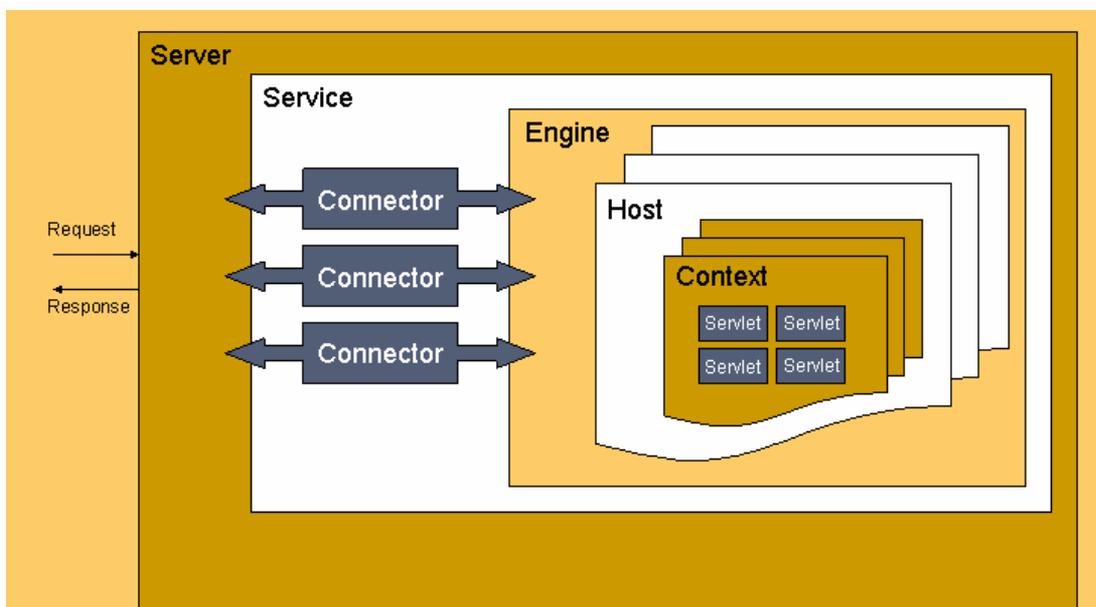
Catalina definiert folgende Schnittstellen

- Server
  - Repräsentant für die JVM-Instanz, in der Tomcat läuft
  - In jeder Tomcat-Konfiguration existiert genau ein Server
    - Konfigurationsdatei: `conf/server.xml`
- Service
  - Beinhaltet alle Konnektoren die auf eine Engine zugreifen
- Connector
  - Ein Connector verbindet Container mit der Außenwelt
    - Engine
    - Host
    - Context
  - Er sorgt dafür, dass die Requests und Responses weitergeleitet werden

- Engine
  - Stellt die oberste Ebene der Verarbeitung dar
  - Sie entscheidet anhand der Details aus dem Request-Objekt, an welchen der ihr untergeordneten Container weitergeleitet wird
  - Im Add-On Betrieb ist die Engine nicht notwendig - ihre Aufgabe übernimmt der Webserver
- Host
  - Damit mehrere Websites auf dem gleichen Webserver betrieben werden können, wurde das Konzept der "virtuellen Hosts" entwickelt
  - Im Add-On Betrieb gewährleistet dies wieder der Webserver
- Context
  - Context repräsentiert die Schnittstelle `javax.servlet.ServletContext` aus der Java Servlet API
  - Der Context beinhaltet mehrere Servlets die gemeinsam eine Web-Application darstellen

## Tomcat Architektur – Catalina (3/3)

- Schema von Catalina



- Jasper die Komponente von Tomcat die für das Übersetzen von Java Server Pages zuständig ist
  - Eigenständige Komponente
  - Akzeptiert werden Seiten in JSP und XML-Syntax
  - Jasper unterstützt on-demand und on-request Kompilierung
    - Wahlweise können alle Dateien einer Anwendung, oder nur bestimmte Bereiche, (*vor*)kompiliert werden
- Jasper erzeugte Servlet-Klassen
  - ... können Abhängigkeiten von Jasper-spezifischen Runtime Support Klassen besitzen
    - Diese können als JAR-File ins *WEB-INF/lib* Verzeichnis der Web-Anwendung gestellt werden
  - ... können von jedem Servlet-Container verwendet werden (auch ohne JSP-Compiler)

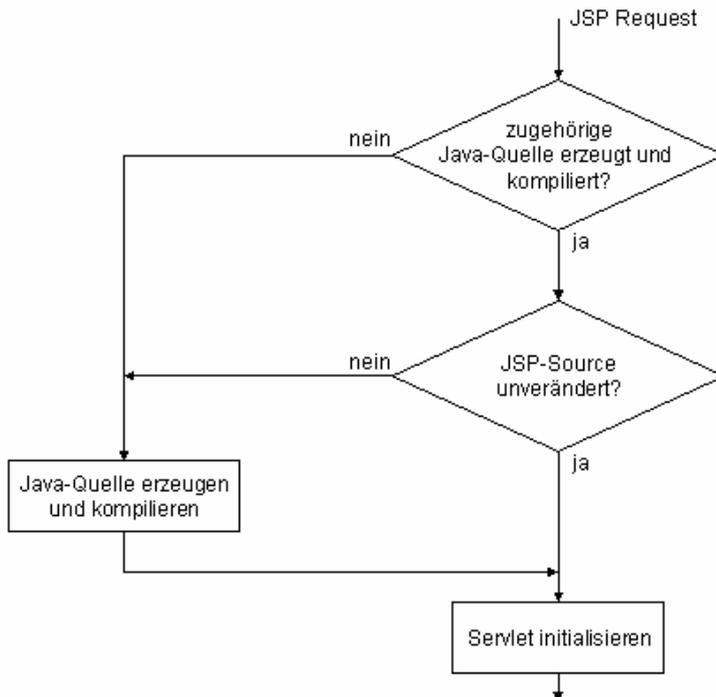
---

## Tomcat Architektur – Jasper (2/2)

---

- Übersetzvorgang
  - Parser
    - Dieser erzeugt aus einer Seite in JSP oder XML Syntax eine DOM (Document Object Model)-Ansicht und Debug-Information
      - Diese wird in die erzeugte Servlet Klasse mit eingefügt
    - Der Parser ist eine eigenständige Komponente
  - Transformer
    - Der Transformer bearbeitet die DOM und Debug-Information aus dem Parser mit den Übersetzungszeittransformationen (Translation Time Transformations)
    - Erzeugt wenn nötig einen überarbeiteten DOM-Baum und überarbeitete Debug-Information
  - Generator
    - Generiert aus dem DOM-Baum und den Debug-Informationen den Quellcode für das entsprechende Java-Servlet
  - Translator
    - Die Aufgabe des Translators ist die Kompilierung des Java-Quellcodes mit dem konfigurierten Java-Compiler zu einer Servlet-Klasse
  - Deployer
    - Der Deployer kann in einer stand-alone Umgebung Einträge für die WADD-

## □ JSP – Anfrage bearbeiten...



## Connectors

### □ HTTP/1.1 Connector

- Der HTTP-Connector ermöglicht die Verwendung von Tomcat als stand-alone Webserver
- Der Connector horcht auf einem Port (standard 8080 für non-secure HTTP) des Servers und leitet die empfangenen Anfragen an die ihm zugeteilte Engine weiter
- Beim Starten des Servers erzeugt jeder Connector eine Menge von Threads die für die Bearbeitung von Anfragen eingesetzt werden können (Thread-Pool)
- Jede Anfrage wird durch einen Thread bearbeitet, der nach dem Bearbeitungsende wieder in den Pool zurückgegeben wird

### □ WARP und AJP Connector

- Der WARP-Connector wird benutzt um Tomcat add-on zu einem anderen Server (z.B. Apache) einzusetzen
- Er verbindet Tomcat über einen Netzwerk-Socket mit dem Webserver (mit dem WARP-Protokoll)
- Tomcat ist damit von außen nicht sichtbar, alle statischen Seiten werden vom Webserver bearbeitet, nur Anfragen die Servlets oder JSP betreffen gehen an Tomcat
- Das gleiche gilt für den AJP-Connector
- Tomcat sollte aus mehreren Gründen eher als add-on den als stand-alone verwendet werden
  1. Tomcat ist als stand-alone beim Bearbeiten von statischen HTML-Seiten langsamer als der Apache.
  2. Apache ist flexibler in der Konfiguration

## □ Installation

- Quellcode und Binaries sind frei verfügbar
- Zusätzlich Installer für die Windows-Version
- Dokumentation
  - BUILDING.txt
    - Installation mittels Code
    - Sehr viele Schritte notwendig
  - RUNNING.txt
    - Installationsanleitung für Binary-Versionen

## □ Konfiguration

- /conf/server.xml
  - Enthält Aufbau der Tomcat-Server Instanz
    - Tomcat Komponenten können auf verschiedene Weise miteinander kombiniert werden
- /conf/web.xml (Deployment Descriptor)
  - Definiert Angaben zu Servlets  
z.B. Session Timeout, MIME Types

---

# Integration in Apache

---

## □ Apache

- Connector *mod\_jk* verbindet Apache Webserver mit Tomcat-Servlet-Server
  - Kann von der Jakarta Homepage herunter geladen werden
  - Muss in das Modulverzeichnis des Apache-Servers kopiert werden
- conf/httpd.conf Anpassung
  - Laden des *mod\_jk* Moduls
  - Angabe eines *JkWorkersFile* (legt die Weiterleitung fest)
  - Bestimmte Verzeichnisse des Apache-Webserver mittels *JkMount* für die Weiterleitung definieren

## □ Tomcat

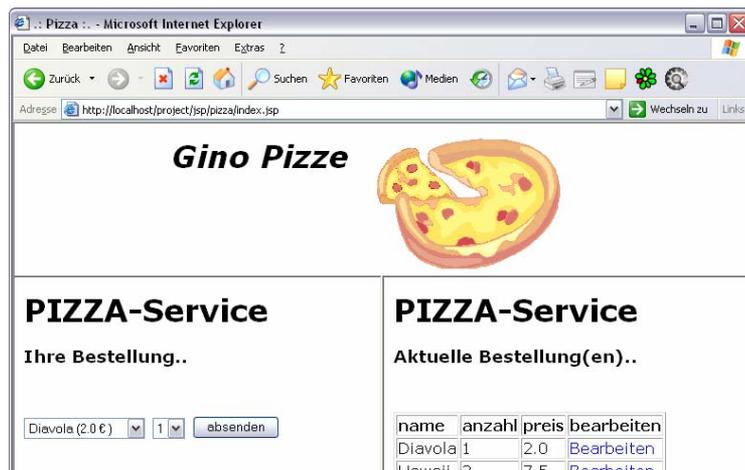
- Anlegen einer worker.properties Datei mit Informationen über
  - Tomcat-Instanz, die Anfragen bearbeiten soll
  - Protokoll, das zur Weiterleitung eingesetzt wird
  - Port, auf dem die Requests entgegengenommen werden

## □ Tomcat 5.0

- Servlet 2.4
- JSP 2.0
- Unterschiede zu Tomcat 4.1.2  
(<http://java.sun.com/webservices/docs/1.2/tomcat/ReleaseNotes.html>)
  - HTML Manager Webapp
  - Directories for Shared Classes
  - No Web server connectors
  - Ant Custom Tasks
  - New "-silent" option to installer

## Beispiele

- [Servlet](#)
- [Java Server Pages](#)
  - [Hello World](#)
  - [Gino Pizze](#)



**Gino Pizze**

**PIZZA-Service**  
Ihre Bestellung..

Diavola (2.0 €) 1 absenden

**PIZZA-Service**  
Aktuelle Bestellung(en)..

name	anzahl	preis	bearbeiten
Diavola	1	2.0	Bearbeiten
Luigi	2	7.5	Bearbeiten

# Danke für die Aufmerksamkeit!

Johannes & Daniel

## Literatur:

- <http://jakarta.apache.org/tomcat/>
- <http://www.galileocomputing.de/>
- <http://www.web-container.com/>
- <http://www.fh-wedel.de/~si/seminare/ws02/Ausarbeitung/a.tomcat/tomcat0.htm>
- Java Servlet Programmierung, O' Reilly 2002