

Content Management mit Java und DB4O

Michael Dienert

25. Juni 2013

Inhaltsverzeichnis

1	Content Management und Datenbanken	1
1.1	html	1
1.2	A patchy server	2
1.3	Apache Tomcat	3
1.4	Webseiten als Dateien	3
1.5	html mit Datenbanken	3
1.6	Weitere Funktionen eines CMS	3
1.7	Verbreitete CMS	3
1.8	Jease	4
2	Installation und Start von Jease	4
3	Content-Types von Jease	5
4	Jease, JSP und JSTL	6
A	Erstes Übungsprojekt	6

1 Content Management und Datenbanken

Content Management Systeme (CMS) sind Softwareanwendungen, die die Inhalte (Content) eines Webauftritts mit Hilfe einer *Datenbank* und einer speziellen Software verwalten (Management).

Durch die Softwareverwaltung wird es möglich, dass mehrere Autoren - ähnlich wie in einer Zeitungsredaktion - zusammen die Inhalte erstellen und bearbeiten (gemeinschaftliches Arbeiten).

1.1 html

Klassische Webseiten werden mit der Auszeichnungssprache *html* geschrieben. *html*-Dateien sind *Textdateien*, die *Text* und *Formatierungsanweisungen* enthalten.

Für die Formatieranweisungen verwendet man den Fachbegriff *markup*. Der *markup* bestimmt, wie der Text dargestellt werden soll (Hoch-, Tiefstellen, fett, kursiv, ...).

Damit der Text vom markup unterschieden werden kann, werden die markup-Befehle in spitze Klammern gesetzt.

Diese Befehle in spitzen Klammern werden auch *Tags* genannt. *Tag* ist ein englisches Wort und bedeutet etwa so viel wie "Preisschild". Auch der Begriff *markup* stammt aus dem Handel und bedeutet dort einfach *Preiserhöhung*.

Neben den einfachen Formatier-Tags gibt es noch eine Unzahl weiterer Tags, die an dieser Stelle nicht behandelt werden sollen. Ein einfachstes html-Beispiel könnte etwa so aussehen:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
  <head>
    <title>Seitentitel</title>
  </head>

  <body>
    <h1>Eine &Uuml;berschrift</h1>
    ein bisschen blabla.

    eine horizontale Linie
    <hr/>
    <address>dienert@wara.de</address>
  </body>
</html>
```

1.2 A patchy server

Wenn man nun Webseiten über das Internet betrachten möchte, benötigt man dazu zwei Programme:

- Einen Webserver, der den html-Text über eine tcp-Verbindung versendet
- Ein Client-Programm, das Seiten über Port 80 bei einem Webserver anfordert und dann den empfangenen html-Text grafisch aufbereitet. Diese Client-Programme werden *Browser* genannt. Genauer ist der Begriff *http user agent*.

User-Agent und Server verwenden dabei das Protokoll *http*, dem fest der Port 80 auf der Serverseite zugeordnet ist.

Erfunden wurden html und http von Tim Berners-Lee am Kernforschungszentrum CERN in Genf.

Dort wurde auch der erste Webserver entwickelt. Da viele Leute weltweit an der Entwicklung dieses Servers teilgenommen haben und ständig neue Patches hinzukamen, war der Server sozusagen ein *Software-Patchwork*, also 'a patchy server'. Daraus wurde dann der Name des Servers: *Apache* und später dann eine grosse Open-Source-Entwicklergemeinschaft, die sich um die Weiterentwicklung des Webserver *Apache* kümmert (aktuelle Version: 2.2).

Neben dem Apache-Webserver pflegt die *Apache Software Foudation* eine grosse Zahl¹ weiterer Projekte.

¹ Aktuell: über 100 Top-Level-Projekte

1.3 Apache Tomcat

Ein sehr bekanntes Apache-Projekt ist *Tomcat*. Tomcat ist ein in Java geschriebener Webserver, der dynamische Webseiten mit Hilfe von Java Servlets und Java Server Pages (JSP) erzeugen kann.

1.4 Webseiten als Dateien

Einen Webauftritt kann man nun komplett mit html-Textdateien und einem Webserver wie z.B. Apache gestalten. Bei modernen, anspruchsvolleren Seiten kommen natürlich noch Multimedia-Daten hinzu, die ebenfalls im Dateisystem des Rechners liegen, auf denen die Apache-Serversoftware läuft.

Der ganze Inhalt (Content) des Webauftritts ist also in vielen Dateien in einer *Ordnerstruktur* gespeichert.

1.5 html mit Datenbanken

Bei grossen Webauftritten wird es zunehmend schwieriger, die Inhalte zu pflegen, weshalb Softwaresysteme entwickelt wurden, die mit Hilfe einer *Datenbank* den gesamten Content verwalten.

Für den User-Agent erfolgt dieser Einsatz völlig transparent: nach wie vor liefert der Webserver über die tcp-Verbindung html-Text und Multimedia-Inhalte zurück. Nur wird dieser nicht aus einer Datei sondern aus einer Datenbank gelesen.

1.6 Weitere Funktionen eines CMS

Viele CMS sind als sog. *Autorensysteme* konzipiert. Ein Autorensystem kann man sich wie eine Zeitungsredaktion vorstellen: verschiedene Autoren schreiben Beiträge und reichen sie bei der Redaktion ein.

Der Chefredakteur bestimmt anschliessend, was davon veröffentlicht wird.

Bei vielen CMS sind Webanwendungen integriert, mit denen auch Autoren ohne html- und Programmierkenntnisse eine Seite oder einen Webauftritt (Site) gestalten können.

1.7 Verbreitete CMS

Sehr weit verbreitete CMS sind z.b.

- Joomla
- Typo3

Beide arbeiten mit der *Kuchensprache*² und benötigen eine Datenbank im Hintergrund (mySQL/mariaDB) und den Apache-Webserver.

²Kuchensprache = PersonalHomePage-Sprache, wer die *Abkürzung* (igitt) ausspricht oder schreibt, muss einen Kuchen spendieren

1.8 Jease

Jease ist ebenfalls ein CMS, unterscheidet sich von Joomla und Typo3 jedoch:

- Als Webserver wird *Apache-Tomcat* verwendet.
- Jease selbst ist eine Webanwendung, die innerhalb von Tomcat läuft. Dadurch lässt sich Jease leicht mit eigenen Java Server Pages erweitern.
- Jease verwendet als Datenbank db4o und nicht das unvermeidliche MySQL. db4o ist eine *Objektdatenbank* und ist natürlich auch zu 100% in Java geschrieben.
Damit wird ermöglicht, dass ein Java-Objekt als Ganzes (mit Eigenschaften und Methoden) dauerhaft in der Datenbank gespeichert und auch wieder in die Java-VM geladen werden kann.
- Jease bringt mit Lucene (von Apache) eine eigene Volltextsuche mit.
- Jease erlaubt es, mit dem eingebauten ZK-Web-Framework Ajax-basierte Webanwendungen zu entwickeln.

2 Installation und Start von Jease

Die Installation und der Start von Jease sind sehr einfach.

Einzige Voraussetzung ist ein Rechner, auf dem das Java-Development-Kit in der Standard-Edition (z.B. Java SE Development-Kit 6 oder 7) installiert ist.

Dann sind folgende Schritte auszuführen:

1. Von hier:

```
http://www.jease.org/download
```

Die neueste Release herunterladen und im gewünschten Verzeichnis entpacken.

2. Im Verzeichnis in das Jease entpackt wurde, das Startskript mit dem Parameter *run* ausführen:

```
bin/catalina.sh run
```

3. Zur Konfiguration des Admin-Accounts, diese Seite aufrufen:

```
http://localhost:8080/cms/setup
```

Passwortanforderungen: Mind. 8 Zeichen, Gross- und Kleinbuchstaben, Ziffern und Sonderzeichen.

4. Anschliessend kann man sich als Administrator hier anmelden:

```
http://localhost:8080/cms/
```

3 Content-Types von Jease

Im Folgenden sind die allerwichtigsten Content-Types von Jease beschrieben. Jease bietet noch weitere Content-Types, deren Funktion in der Online-Dokumentation beschrieben sind.

Text Dieser Content-Type enthält einfachen Text (plain text) oder html-Text (Text und Markup). Bei der Erzeugung eines Text-Objekts startet eine Editor-Webanwendung und man kann direkt in einem Browserfenster den Text erzeugen. Dabei kann man den Text entweder in html (indem man auch dem Markup eingibt) oder mit Hilfe der Textverarbeitungsfunktionen des Editor formatieren. Im letzteren Fall wird der Markup automatisch hinzugefügt und es sind keinerlei html-Kenntnisse notwendig.

Ordner Ordner-Objekte erlauben, eine baumförmige Ordnung aufzubauen. Die Inhalte eines Ordners werden in einer Navigationsspalte angezeigt, wenn bei den entsprechenden Inhalts-Objekten das Häkchen "Sichtbar" gesetzt wurde.

Achtung: enthält ein Ordner nur Unterordner aber keine Dateien oder andere Objekte, wird keine Navigationsspalte angezeigt! Auch dann nicht, wenn bei den Unterordnern das Häkchen "sichtbar" gesetzt ist.

Komposition Eine Komposition ist ein Ordner, der seinen gesamten Inhalt als eine einzelne Seite darstellt.

Z.B. eignet sich eine Komposition dazu, Text und dazugehörige Bilder zu gruppieren.

Gliederung Mit einem Gliederungselement kann man ein Unterkapitel in er Navigationsspalte erzeugen.

Datei beliebige Binärdateien

Bild Wie Datei, als Content sind aber ausschliesslich Bilddateien erlaubt.

Zugang Mit diesem Objekt kann man einzelne Ordner schützen. Dazu wird ein neuer Ordner erstellt und darin wiederum das Zugangs-Objekt mit einem Benutzernamen, Passwort und evtl. Laufzeit.

Diskussion Hiermit wird ein Forum erzeugt. In der Standardeinstellung ist die Ansicht flach, d.h. die Benutzer können nur Kommentare zu einem Thema hinzufügen.

Über eine Konfigurationseinstellung³ lässt sich aber eine Baumansicht einschalten, so dass die Benutzer auch die Kommentare ihrerseits kommentieren können.

Skript Mit diesem Objekt kann man sehr einfach Seiten mit Funktionalität erzeugen. In der Voreinstellung wird JSP verwendet.

Auch die Verwendung der *JSP Standard Tag Library* ist möglich, dazu müssen wir aber erst die JSTL-Bibliothek in das lib-Verzeichnis vom jease kopieren. Vgl. Kap. 4

³CMS → System → Parameter. Dort `JEASE_DISCUSSION_PRESENTATION = thread` hinzufügen.

4 Jease, JSP und JSTL

Um die JSP Standard Tag Library verwenden zu können, muss man zwei Bibliotheken (in Form von jar-Dateien) in das lib-Verzeichnis von Jease kopieren. Das lib-Verzeichnis liegt direkt unterhalb des jease-Verzeichnisses.

Diese beiden jar-Archivdateien der JSTL-Library gibt es hier:

```
https://jstl.java.net/download.html -> JSTL Implementation
```

Als dieser Text geschrieben wurde waren die aktuellsten Versionen diese hier:

```
javax.servlet.jsp.jstl-1.2.1.jar  
javax.servlet.jsp.jstl-api-1.2.1.jar
```

Diese beiden Dateien müssen also heruntergeladen und nach `jease/lib/` kopiert werden. Dann sollte das folgende Beispiel auf Anhieb funktionieren:

Inhalt eines einfachen Skript-Objekts. Es wird eine JSP-Seite mit JSTL verwendet.

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  
  
<form method="post">  
  <input type="text" name="textfeld"/>  
  <input type="submit" name="hau wech!"/>  
</form>  
  
  Du hast <c:out value="${param.textfeld}" default="noch nix"/>  
  eingegeben.
```

Die offizielle Dokumentation zur JSTL findet sich hier:

```
http://docs.oracle.com/javaee/5/jstl/1.1/docs/tlddocs/
```

A Erstes Übungsprojekt

Erstelle den Webauftritt eines Vereins. Dabei soll enthalten sein:

- Die Startseite
- Eine Seite mit kurzen Steckbriefen der Vorstandsmitglieder und des Beirats. Mit je einem Foto zu jeder Person
- Eine Fotogalerie
- Ein Diskussionsforum
- Eine Seite mit den Jahrestermen (Hauptversammlung, Feste, Veranstaltungen, ...)
- Eine Fotogalerie
- Eine Seite mit aktuellen Daten. Z.B. Wettkampfergebnisse bei Sportvereinen
- Ein Impressum (vgl. Telemediengesetz, Impressumspflicht)