

Recherchebericht

der Gruppe SWP12-2

Inhaltsverzeichnis

1. Begriffe	2
1.1. GUI	2
1.2. RDF	2
1.3. Semantik Web	2
1.4. Linked Data	2
1.5. MVC	2
1.6. DBS	2
1.6.1. SQL	3
1.6.2. SPARQL	3
1.6.3. MySQL	3
1.7. Servlet	3
1.8. Framework	3
1.9. Hibernate	4
1.10. LIMES	4
2. Konzepte	4
2.1. Semantik Web, RDF & Linked Data	4
2.2. WebGUI	5
2.3. MVC & Erweiterbarkeit	5
2.4. User Management	5
2.5. Tab-Management	6
2.6. Link Specs.	6
2.7. Framework	6
3. Vergleich elementarer Frameworks entsprechend der Projektanforderungen	7
3.1. Vaadin	7
3.2. Apache Tapestry	7
3.3. Wicket	8
3.4. Apache Cocoon	9
3.5. PlayFramework	9
3.6. Jpublish	10
3.7. Hibernate	10
Quellen	10

1. Begriffe

1.1. GUI

Die GUI, oder das *Graphical User Interface* ist eine grafische Schnittstelle, welche dem Nutzer erlaubt, über Eingabegeräte und grafische Symbole, mit einem Programm zu interagieren. Die grafischen Symbole werden hierbei verwendet, um die Computerlogik den gemeinen Nutzer ersichtlicher zu machen. Die GUI bildet die Schnittstelle zwischen Mensch und Maschine, daher der oft verwendete Name der „Mensch-Maschine-Schnittstelle“.

1.2. RDF

Resource Description Framework ist eine Standardisierung zur Beschreibung formaler Informationen über Objekte, die durch eindeutige URIs identifizierbar sind. Ursprüngliche Idee der RDFs war das Beschreiben von Metadaten im World Wide Web, die wahre Aussagen über Ressourcen treffen. Diese Aussagen werden dabei als Tripel modelliert, also als eine Sequenz aus 3 Elementen, dem sogenannten Subjekt (die Ressource, über die eine Aussage getroffen wird), dem Prädikat (eine Eigenschaft des Subjekts) und dem Objekt (das Argument des Prädikats).

1.3. Semantik Web

Das Semantic-Web stellt eine Erweiterung des World Wide Web dar, die es möglich macht, für Maschinen verwertbare Inhalte zur Verfügung zu stellen. Dafür ist eine maschinell automatisch interpretierbare Verknüpfungsstruktur notwendig, die verschiedene Orte, Personen und Dinge in Beziehung setzt.

1.4. Linked Data

Linked Data ist der Standard, über den man Daten im Semantic Web zu verlinkt, um eine maschinell lesbare Einheit aus ähnlichen Daten verschiedener Quellen zu schaffen. So werden Daten mit einem Uniform Resource Identifier (URI) identifiziert, um so auch auf andere Daten verweist, die eventuell weitere Informationen zum selbigen Gegenstand bereithalten. Das wohl bekannteste Beispiel für URIs ist der Uniform Resource Locator (URL), der die Lokation einer Ressource im Internet angibt.

1.5. MVC

Das *Model-View-Kontroller*-Konzept ist ein Architekturmuster das der strukturierung beim Softwareentwicklungsprozess helfen soll. Ziel des Musters ist ein flexibler Programmwurf, bei dem ein späteres Weiterverarbeiten des Quellcodes und das Wiederverwenden einzelner Programmkomponenten erheblich vereinfacht wird.

1.6. DBS

Ein Datenbanksystem ist ein System zur elektronischen Datenverwaltung, dessen Ziel eine Speicherung großer Datenmengen ohne Widersprüche über lange Zeiträume ist.

1.6.1. SQL

Structured Query Language ist eine Datenbanksprache die es einem erlaubt, Daten(, -sätze) einzufügen, zu bearbeiten, zu löschen, oder sie einfach nur abzufragen. Durch den einheitlichen Einsatz von SQL auf Datenbanken soll eine Unabhängigkeit von eingesetzten Datenbankmanagementsystem erreicht werden.

1.6.2. SparQL

SparQL ist eine Graph-basierte Anfragesprache für RDF-Graphen. Sie funktioniert von der Abfrageweise also wie eine SQL-Abfrage, findet allerdings keine einfachen Datenbankeinträge, sondern RDF-Daten im Web, um mit der Ausgabe dieser RDFs die Anfrage zu beantworten. Da Daten in RDF-Form so gut standardisiert sind, ist die Möglichkeit einer Suche, wie SPARQL sie anwendet, überhaupt erst gewährleistet.

1.6.3. MySQL

MySQL ist ein von MySQL AB (bis 2008, bis 2010 SUN, ab 2010 Oracle) entwickeltes und quelloffenes SQL-Datenbankmanagementsystem. Mit Hilfe von MySQL werden Daten in Tabellen gespeichert, geändert, gelöscht oder abgefragt. Es verwendet die standardisierte Anfragesprache SQL.

1.7. Servlet

Als Servlet werden Java Klassen bezeichnet, deren Instanzen Anfragen von Clients innerhalb eines Java-Webserver annehmen und beantworten. Sie bilden einen festen Bestandteil aller Java-EE-Anwendungsserver. Ihre Antwort kann dynamisch erstellt werden und muss nicht statisch auf dem Webserver verfügbar sein. Ihr typisches Aufgabenfeld umfasst die Kontrolle von Downloads und Uploads, sowie Sicherheitsüberprüfung und Protokollierung.

1.8. Framework

Ein Framework stellt den „Rahmen“ zur Entwicklung einer Software bereit, innerhalb dessen Anwendungen erstellt werden. Dem Framework liegen bestimmte Entwurfsmuster zugrunde, wodurch auch die Struktur der hiermit entwickelten Anwendung beeinflusst wird. Das Framework selbst ist nicht als fertiges Programm zu verstehen. Es kontrolliert weiterhin den Kontrollfluss der Anwendung, sowie dessen Schnittstellen, und gibt letzten Endes auch die Anwendungsarchitektur vor. Eine genauere Framework-Definition ist nicht allgemein möglich, da es zu viele Frameworks mit zu vielen zentralen Unterschieden gibt, und somit müssen die Frameworks im Einzelnen beschrieben werden.

1.9. Hibernate

Hibernate ist ein Java-Framework, um Objekte mit Attributen und Methoden in Datenbanken zu speichern (Object-Relational Mapping) und aus diesen Datensätzen wieder Objekte zu erzeugen. Die Beziehungen zwischen einzelnen Objekten werden auf Datenbank-Relationen abgebildet. Hibernate kann in Java-Applikationen und Servlet-Engines benutzt oder in Applikationsserver eingebunden werden.

1.10. LIMES

LIMES ist ein Framework, welches Links zwischen Entitäten aus Linked Data Sources finden soll. Es kombiniert mathematische Charakteristika des metrischen Raums als auch Suffix-, Präfix- und Positionsfilterung, um eine Approximation der Gleichheit von Datensätzen zu kalkulieren. Diese Annäherung wird genutzt um eine große Anzahl Datensatzpaaren auszufiltern, die nicht die Ansprüche des Mappings erfüllen.

2. Konzepte

Was wollen wir eigentlich erreichen? Um diese Frage zu beantworten, müssen wir die Gegebenheiten erst einmal im Einzelnen betrachten.

2.1. Semantik Web, RDF & Linked Data

Der gemeine Internetnutzer kennt das Web als ein riesiges Geflecht aus Webseiten, doch was für den Nutzer verständlich ist, erkennt eine Maschine nicht so leicht. Da das Web sehr unstrukturiert ist, ist es für Maschinen in aller Regel unmöglich, automatisiert einen Bezug zwischen 2 Gegenständen herzustellen, die doch eigentlich dieselbe Sache meinen. An dieser Stelle kommt das Semantik Web ins Spiel. Man hat sich zum Ziel gesetzt, die Bedeutung aller Informationen so zu strukturieren, dass auch eine Maschine sie verstehen kann. Die Struktur solcher Informationen wird in RDF-Formaten gehalten. Da nun Informationen in einem Standard strukturiert sind, ist es möglich automatisierte Prozesse laufen zu lassen, die „ähnliche“ Daten finden und über ihre Uniform Resource Identifiers (URIs) Links zueinander erstellen. Solche Links bezeichnet man als Linked Data. Mittels dieser verlinkten Daten können auch Datenbanken erstellt werden, die ähnliche URIs sammeln und somit „ähnliche Treffer“ sehr viel leichter Auffindbar machen.

Das ist das Prinzip das auch LIMES verfolgt.

2.2. WebGUI

Bevor man sich mit der Arbeit an einer GUI für ein System wie LIMES auseinandersetzt, stellt sich natürlich die Frage, was genau eine WebGUI denn eigentlich ausmacht. Sie bezeichnet ein System, in dem Administratoren die Möglichkeit haben, Aussehen und Features für Nutzer verschiedener „Zugriffslevel“ zu verwalten. Solche Features umfassen mitunter das Recht, Inhalte zu editieren, sie überhaupt erst Einsehen zu dürfen, und sogar solche Rechte, wie die Verfolgung von Update-Tätigkeiten anderer Nutzer. WebGUIs bieten allerhand Möglichkeiten, was die Gestaltung derer Inhalte angeht. Dazu gehört auch eine Unterstützung von Datenbanken die mithilfe der WebGUI verwaltet werden sollen.

2.3. MVC & Erweiterbarkeit

Strukturierter Quellcode ist unerlässlich. Wie man die Struktur aufbaut, kann von Projekt zu Projekt variieren, jedoch sollte man sich dabei auf ein ausgemachtes Muster stützen können. Fehlt dies, wird es mit jeder Zeile an Quellcode aufwändiger und schwieriger, sich in den Gegebenen Code einzuarbeiten, um ihn zu verstehen und eventuell zu erweitern.

Erweiterbarkeit hat für uns in diesem Projekt sogar eine hohe Priorität. Da wir bei der Arbeit an einer WebGUI auch sehr nah am Endnutzer arbeiten, fällt die Wahl des Musters für uns zwangsläufig auf das altbewährte MVC-Muster zurück. MVC hat sich schon lange bewährt und bietet für eine solche Aufgabe alle nötigen Vorteile. Die Datenschicht wird strikt von der Aufmachung und dem Kontrollpanel der Nutzer getrennt. Sollte also beispielsweise eine Erweiterung der grafischen Oberfläche gewünscht sein, ergeben sich damit auch im späteren Verlauf keinerlei Probleme für die hintergründigen Funktion oder die Richtigkeit der dargestellten Daten.

2.4. User Management

Das User Management in einer WebGUI wurde bereits erwähnt. LIMES soll nur seine Nutzer nur auf 2 Arten unterscheiden. Es gibt die Gastnutzer, die die Seite besuchen und nutzen dürfen, ohne irgendwelche Angaben machen zu müssen, und es gibt die registrierten User. Für registrierte User ist ein Username sowie ein Passwort nötig. Alle registrierten User sind gleichwertig und haben untereinander dieselben Rechte. Ein Administrator ist in dieser Form nicht vorgesehen. Welche Rechte genau Gäste, und welche Rechte registrierte User spielt zu diesem Zeitpunkt der Entwicklung noch keine Rolle.

2.5. Tab-Management

Es ist gewünscht, dass ein Nutzer der WebGUI mehrere „Tabs“ öffnen kann. Ein Tab stellt ein Fenster innerhalb der WebGUI dar, sodass mehrere Anfragen gleichzeitig verarbeitet werden können, oder einfach nur um gewisse Informationen nicht schließen zu müssen, wenn man eine neue Aufgabe beginnen möchte, sodass man durch anklicken des Tabs diese Informationen jederzeit wieder Einblenden kann. Jeder Tab ist eine Instanz die dieselben Möglichkeiten und Rechte hat wie alle anderen. Natürlich bezieht sich dies nur auf dieselben Rechte, wie sie der Anwender mit seinem Zugriffslevel auch hat, sodass ein Gast keine Rechte bekommt, wie sie sonst nur ein registrierter User hätte.

2.6. Link Specs

Link Specs sind das Ergebnis eines LIMES-Arbeitsvorgangs. Der Nutzer erhält eine .nt-Datei, die eine Liste von allen Paaren von Links enthält, von denen vermutet wird, die RDFs seien sich sehr ähnlich. Diese Datei kann der Nutzer privat Speichern, erneut Verändern, und auch ein erneutes Laden ist möglich. Es besteht die Möglichkeit, die Specs öffentlich zu machen, dies lehnt jedoch mit Wahrscheinlichkeit an den Zugriffslevel des Nutzers an.

Auch das Anlernen von Link Specs soll über Batch und Active Learning möglich sein, indem ein Nutzer Daten auswählt und hochlädt. Eine Feedbackfunktion soll hierbei integriert sein, um Schwachstellen und stärken über die Erfahrung vieler Nutzer erkennbar zu machen.

Großer Wert wird dabei auf eine weitere Funktion gelegt, und zwar auf die Möglichkeit der Bewertung von Link Specs. Zuerst einmal ist es möglich, eine Anzahl zufälliger Link Specs auszuwählen. Der Nutzer kann diese nun völlig frei Bewerten und auch unter negativen sowie positiven Link Specs einordnen. diese Bewertung kann gespeichert werden. Auch diese Funktion kann als Feedback dienlich sein.

2.7. Framework

Ein weiterer wichtiger Aspekt ist das gewählte Framework. Wie schon unter dem Aspekt beschrieben, ist ein Framework ein Art Grundgerüst, in dem die Entwicklungsarbeit stattfindet. Die Wahl eines Frameworks kann die Entwicklungsarbeit nicht nur um ein vielfaches erleichtern, es kann auch das Ergebnis der Programmierarbeit um einen Faktor verbessern. Die Wahl des Frameworks muss jedoch mit Sorgfalt getroffen werden, denn man möchte sich immerhin Vorteile verschaffen, und durch die Wahl des falschen Frameworks, könnte man alle Vorteile in den Sand setzen. Um

3. Vergleich elementarer Frameworks entsprechend

der Projektanforderungen

3.1. Vaadin

Vaadin ist ein Webframework für Rich Internet Applications mit serverseitiger Architektur. Auf Client bzw. Browserseite wird hingegen das Google Web Toolkit verwendet (Darstellung), auf dessen Grundlage es auch basiert. Es nutzt das MVC Entwurfsmodell und unterstützt Browsertabs, sowie den Back-Button und Deep-links. Es ist möglich per Java oder HTML Layouts für dieses Framework und eine Kompatibilität zur OSGi-Plattform zu erstellen. Ebenfalls ist eine Drag-and-Drop-Unterstützung vorhanden. Vaadin stellt eine Vielzahl von Interfaces und Komponenten bereit. Weiterhin ist es möglich vorgefertigte, sowie eigene erstellte „Themes“ für das Erscheinungsbild der Applikation zu nutzen bzw. diese noch auf CSS Grundlage zu bearbeiten.

Vorteile:

Durch die Möglichkeit, die gesamte Anwendung in Java zu erstellen und der strikten Trennung nach MVC Entwurfsmuster steht der Erweiterbarkeit einer erstellten Applikation nichts im Wege. Ein gemeinsames nutzen von Vaadin und Hibernate ist auch möglich, so dass man auf unkomplizierte weise das User Management und die persistente Haltung bzw. Bearbeitung von Link Specs mit Bewertungen und Kommentaren erreichen kann. Ein weiterer zu betrachtender Vorteil von Vaadin ist die Performance und Kompatibilität, sie einerseits durch Delta-Rendering (nur Änderungen übertragen und nicht Seite neu laden) und andererseits durch übertragen des Erscheinungsbildes über eine UIDL (User Interface Definition Language) welche dann durch die GWT Rendering Engine in eine Oberfläche transformiert wird. Die erwünschten Tabs können durch Vaadins Tab-Unterstützung und das dank Java mögliche öffnen mehrerer Instanzen in diesen Tabs geschehen.

3.2. Apache Tapestry

Apache Tapestry ist ein Wicket sehr ähnliches Framework, ebenfalls von Apache bereitgestellt, da beide als oberstes Ziel hatten, die Entwicklung von Web-Anwendungen zu vereinfachen und komfortabler zu gestalten. Im Gegensatz zu Wicket hat sich Tapestry allerdings eher an der Optimierung des Quelltextes orientiert, wodurch sowohl der Code kleiner als auch das Endprodukt weniger Prozessorleistung und Arbeitsspeicher verbraucht. Trotz dieser Optimierungen sind mit Tapestry erstellte Web-Anwendungen einfacher zu handhaben als viele herkömmliche Servlets, da der kleinere Basiscode das Framework übersichtlicher gestaltet. Ebenso muss man sich nicht um die Verwaltung der Systemressourcen kümmern, wie zum Beispiel Speicherverbrauch oder Multithreading, allerdings kann man dies tun, wenn man bestimmte Ziele verfolgt (minimaler Speicherverbrauch o.ä.). Tapestry verfolgt eine Aufteilung der Entwicklung in einen Layout-Teil und einen Code-Teil, was der Übersichtlichkeit des Frameworks ebenfalls zugunsten kommt. Der Nachteil dieser Aufteilung ist allerdings, dass das Framework das Layout stark über XML zur Verfügung stellt und dadurch sehr XML-lastig wird.

3.3. Wicket

Wicket ist ein Framework von Apache, welches darauf ausgelegt ist, die Entwicklung von Weboberflächen ähnlich zu gestalten wie die Entwicklung von Desktopbenutzeroberflächen. Daraus folgert sich, dass Programmierer, die bereits ein wenig Erfahrung im Umgang mit GUI-Programmierung haben, einen einfacheren Einstieg in das Framework haben werden und das Ergebnis dem Endbenutzer vertraut vorkommen sollte. Um dieses Ergebnis zu erreichen, setzt Wicket sogenannte Panels ein, mit denen man die Seite nach eigenen Wünschen ändern bzw. aufbauen kann. Da diese Panels alle auf einem Grundgerüst (eben dem „Panel“) basieren, lassen sich sehr leicht wiederverwendbare Komponenten entwickeln, welche auch mit eigenen Funktionen und Scripts ausgestattet werden können, sollte dies nötig sein. Diese Funktionen werden im allgemeinen mit AJAX (Asynchronous JavaScript And Xml) eingebunden, welche in Wicket entgegen dem Namen fast ohne Kenntnisse von JavaScript möglich ist. Um dies zu erreichen, stellt Wicket einzelne Klassen bereit, die diese Funktionalitäten beinhalten. Die an Desktopbenutzeroberflächen angelegte Entwicklung von Wicket verschafft dem Framework leider den Nachteil, dass es durch die ähnlichen Programmierparadigmen mehr Arbeitsspeicher benötigt. Ebenso ist die Dokumentierung des Frameworks nur teilweise vorhanden, da die letzte Version des Javadocs auf Version 1.4 basiert, aber die aktuelle Version von Wicket 1.5 ist. Allerdings gibt es dagegen sehr viele Tutorials im Internet, mit denen man sich in die Arbeit mit Wicket einarbeiten kann. Zwar ist das Projekt Open Source, sodass man argumentieren könnte, dass man sich im Wicket SVN informieren könnte, allerdings ist ein Javadoc deutlich übersichtlicher und in den meisten Fällen auch nützlicher als der Quellcode, solange man nicht direkt im Quellcode arbeitet, sondern sich über bestimmte Funktionen informieren will. Um Daten zu speichern, muss man mit Wicket andere Bibliotheken oder Frameworks wie zum Beispiel Hibernate einbinden, da Wicket keine eigenen Klassen besitzt, die diese Funktionen einbinden können.

3.4. Apache Cocoon

Apache Cocoon basiert auf dem Struts-Framework und im Vordergrund steht die Komponenten-basierte Entwicklung von Applikationen. Dazu müssen in der Regel je eine

Datei für Darstellung, Logik und Inhalt angelegt werden. Es werden multiple Ausgabeformate unterstützt und Daten werden in Form von XML-Dateien gespeichert. Modularisierung erfolgt in Cocoon in Form von Blöcken, wobei Blöcke unter anderem Servlet Services, Services die Pipelines zur Verfügung stellen, Services von Komponenten oder aber Klassen und Ressourcen Container enthalten. Diese Blöcke werden als *Java Archiv* (JAR) zur Verfügung gestellt. Datenkomponenten sind in Cocoon von der Ansicht getrennt.

Vorteile:

Die Trennung und Aufspaltung der Codierung von Darstellung, Logik und Inhalt bietet zwar einerseits einen Mehraufwand bei der Erstellung und Pflege der Daten, aber es bietet die Möglichkeit einzelne Komponenten zu ersetzen und diese eventuell so zu modifizieren, dass es der Erweiterbarkeit des Programms dienlich sein kann. Durch die Möglichkeit, Cocoon durch beliebige eigene geschriebene (oder aus anderen Frameworks stammende) Komponente zu erweitern, ist auch der Aspekt der Erweiterbarkeit genüge getan und zudem können dadurch nahezu alle Aspekte, mit vorhanden sein der notwendigen Zeit um neue Komponente zu erstellen, erfüllt werden. Cocoon kann auch zusammen mit Hibernate genutzt werden, was eine Hilfe bei der Erstellung des User Managements und der Speicherung und Bearbeitung von Link Specs sowie der dazugehörigen Bewertung und Kommentierung bietet.

3.5. PlayFramework

Play ist ein open source Framework, welches in Java geschrieben ist und auf den Prinzipien der MVC-Architektur beruht. Es wurde mit der Idee des Rapid Application Development entwickelt um die Programmierung von Webanwendungen zu beschleunigen. Play ist einfach zu bedienen und intuitiv zu nutzen. Die geschriebenen Programme lassen sich auf dem mitgelieferten Webserver lokal sofort anzeigen und debuggen. Dabei ist kein Neustarten des Webservers nötig. Play debuggt und compiliert das Programm im Hintergrund. Ein einfaches „neu laden“ der Internetseite reicht um die Änderungen sichtbar zu machen. Dabei unterstützt Play auch bereits bewährte Java-Bibliotheken wie Hibernate. Play vereinfacht hierbei besonders die Arbeit mit Datenbanken. Neben der Einbindung von Hibernate und JPA stellt Play eine eigene Komfortschicht zur Verfügung, welche einen Großteil der Konfigurationen übernimmt. So werden zum Beispiel automatisch Datenbanktabellen erstellt, falls diese für eine neue Klasse noch nicht zur Verfügung gestellt wurden. Ein weiterer Vorteil den Play hier bietet, ist das CRUD-Modul (create, read, update, delete), welches die Grundlage für eine persistente Datenhaltung liefert. Eine weitere Komfoteigenschaft die Play mitbringt, ist das automatische Erstellen von Gettern und Settern von Methoden. Desweiteren ist Play komplett RESTful, es wird kein Java EE genutzt. Somit lässt sich Play äußerlich besser skalieren als vergleichbare Frameworks. Neben all den Vorteilen, bringt es aber auch einige Nachteile mit. Die eben beschriebene Unabhängigkeit ist gleichzeitig ebenso ein Nachteil – es basiert nicht auf Servlets. Ein weiterer temporärer Nachteil ist das relativ junge Alter von Play. Das Framework ist noch nicht so verbreitet wie andere. Ebenso ist die Dokumentation nicht in einem solchen Maß vorhanden wie bei

anderen etablierten Frameworks. Die hohe Leistungsfähigkeit von Play verschafft leider auch den Nachteil, dass es zum einen sehr Speicherfressend ist, je größer das Projekt wird, zum anderen steigen die Compilierungszeiten mit der Größe drastisch an.

3.6. Jpublish

JPublish ist ebenfalls ein Javaframework und enthält viele Eigenschaften von Play. So lässt sich der Quellcode während der Laufzeit kompilieren und debuggen. Ein neues laden der Internetseite reicht aus um das Programm zu aktualisieren. Jedoch gibt es hier bereits einen Unterschied zu Play. JPublish liefert keinen eigenen Webserver mit, so dass man zum Testen von Anwendungen auf einen eigenen konfigurierten Server zurückgreifen muss. Auch enthält JPublish von Haus aus nicht Hibernate. Jedoch ist das Einbinden von neuen Modulen und Komponenten einfach und problemlos.

3.7. Hibernate

Hibernate ist ein Framework für Java, das es ermöglicht Objekte mit Attributen und Methoden in relationalen Datenbanken abzuspeichern. Gleichzeitig kann es diese Datensätze wieder in Objekte umwandeln. Beziehungen zwischen Objekten werden als Relationen gespeichert. Abfragen lassen durch den für Hibernate eigens entwickelten SQL Dialekt HQL oder mittels der Hibernate Criteria-API durchführen, welche sich an den Javaquellcode anlehnt. Desweiteren lässt sich Hibernate durch neue SQL-Dialekte erweitern. Ein weiterer Vorteil ist die Unabhängigkeit von der genutzten Datenbank. Hibernate kann mit jeder Datenbank genutzt werden, solange der JDBC-Treiber implementiert ist.

Quellen

<http://de.wikipedia.org/wiki>

http://www2.math.uni-wuppertal.de/~axel/skripte/oop/oop1_3.html

[http://www.informatik.uni-](http://www.informatik.uni-leipzig.de/~meiler/MuP.dir/MuPWS11.dir/Vorlesung/Kap13_MVC.pdf)

[leipzig.de/~meiler/MuP.dir/MuPWS11.dir/Vorlesung/Kap13_MVC.pdf](http://www.informatik.uni-leipzig.de/~meiler/MuP.dir/MuPWS11.dir/Vorlesung/Kap13_MVC.pdf)

http://www.bluecoat.com/sites/default/files/documents/files/User_Management.8.pdf

http://www.theserverside.com/discussions/thread.tss?thread_id=63045

<http://it-republik.de/jaxenter/artikel/Webanwendungen-mit-dem-Java-Framework-Play!-2815.html?print=1>

<http://www.playframework.org/>

<http://code.google.com/p/jpublish/>