

Seminar Hibernate

Seminar Hibernate

ZHAW - Zürcher Hochschule für Angewandte Wissenschaften

Ramon Schilling

schilram@students.zhaw.ch

11. Juni 2013

Inhaltsverzeichnis

1	Einleitung	2
1.1	Aufgabenstellung	2
1.2	Deliveries	2
1.3	Zielsetzung	2
1.4	Motivation	2
2	Projektmanagement	3
2.1	Projektplanung	3
2.2	Aufwand	3
2.3	Hilfsmittel	3
2.3.1	Versionskontrolle	3
2.3.2	Dokumentation	3
2.3.3	Programmierung	4
2.3.4	Datenbank	4
3	Einarbeitung	5
3.1	Vorgehen	5
4	Anwendung	6
4.1	Frameworks	6
4.2	Architektur	6
4.2.1	Dependencies	6
4.2.2	Struktur	6
4.2.3	Klassendiagramm	7
4.2.4	model Klassendiagramm	7
4.2.5	service Klassendiagramm	8
4.3	Installation	8
4.3.1	Voraussetzungen	8
4.3.2	Datenbank einrichten	8
4.3.3	Tomcat einrichten	8
5	Abbildungsverzeichnis	9
6	Literaturverzeichnis	10

1 Einleitung

Dieses Dokument wurde für das Seminar „SW Entwicklung mit Hibernate“ geschrieben.

1.1 Aufgabenstellung

- Einarbeitung in das Thema ORM mit Hibernate.
- Erstellung einer kleinen Anwendung mit Gui welche das Hibernate-Framework nutzt.
- Dokumentation welche die Architektur sowie die einzelnen Komponenten erklärt.

1.2 Deliveries

- Source Code und lauffähiger Maschinencode.
- Dokumentation

1.3 Zielsetzung

Das Ziel der Seminararbeit ist es, das O/R-Mapping Paradigma am Beispiel des Hibernate Frameworks zu verstehen und eine Anwendung unter Verwendung dieses Frameworks zu implementieren.

1.4 Motivation

Da ich seit Oktober letzten Jahres als Java Entwickler arbeite, war für mich klar, dass ich ein Seminar wählen wollte mit welchem ich meine Entwicklungskenntnisse vertiefen und neue Erfahrungen sammeln kann. Da das Thema ORM bei der Entwicklung von Software sehr wichtig ist, da die meisten Applikationen irgendwie mit einer Datenbank arbeiten war die Wahl schnell gefallen. Wir arbeiten in der Firma auch mit Hibernate und setzen auch das Framework Spring ein. Da ich aber noch kein Projekt von Grund auf mitaufgebaut habe, wollte ich diese Arbeit als Chance nutzen nicht nur Hibernate genauer kennenzulernen sondern auch Spring, bzw. Teile davon. Im Speziellen wollte ich Spring MVC und Spring Data - JPA genauer anschauen.

Zur Umsetzung der Anforderungen möchte ich eine Anwendung entwickeln in welcher Rezepte verwaltet werden können. Als Besonderheit soll es möglich sein nach Rezepten zu Suchen in dem man die vorhandenen Zutaten angibt.

2 Projektmanagement

2.1 Projektplanung

20. März 2013 - Kick Off Meeting
27. März 2013 - Einreichen der Aufgabe
12. Juni 2013 - Abgabe Schriftliche Arbeit und Anwendung
19. Juni 2013 - Präsentation

2.2 Aufwand

Der Aufwand für die Seminararbeit sollte 50 h betragen.

Beschreibung	Soll	Ist
Einarbeitung in Hibernate	4 h	2 h
Einarbeitung in Spring	4 h	2 h
Einrichten der Entwicklungsumgebung inkl. Hibernate Test	4 h	8 h
Programmieren der Anwendung	28 h	32 h
Dokumentation	8 h	8 h
Total	48 h	52 h

Die Aufwandschätzung fiel mir relativ schwer, was sich auch darin zeigt, dass die Soll und Ist Zeiten nicht übereinstimmen. Ich habe für die theoretische Einarbeitung in Hibernate und Spring weniger Zeit aufwendete als geplant um schneller mit der Umsetzung zu beginnen. Bis ich dann aber alles soweit konfiguriert hatte, dass das Object-Relation-Mapping funktionierte hatte ich viel mehr Zeit gebraucht als geplant. Beim Programmieren habe ich ebenfalls z.T. sehr viel Zeit gebraucht um Fehler zu finden welche ich mit etwas mehr Erfahrung wohl gar nicht erst gemacht hätte.

2.3 Hilfsmittel

2.3.1 Versionskontrolle

Um eine Versionskontrolle zu haben, habe ich in Github [1] ein Repository erstellt und sämtliche für das Projekt nötigen Dateien dort eingchecked.

2.3.2 Dokumentation

Die Dokumentation habe ich mit \LaTeX erstellt. Als Editor habe ich TeXnicCenter [5] genutzt.

2.3.3 Programmierung

Als Entwicklungsumgebung für die Java Programmierung habe ich IntelliJ IDEA [3] genutzt.

2.3.4 Datenbank

Als Datenbank habe ich auf PostgreSQL [2] zurückgegriffen.

3 Einarbeitung

3.1 Vorgehen

Um mich mit dem Thema ORM und Hibernate auseinanderzusetzen habe ich zuerst im Internet nach Informationen gesucht. Da mich die Integration zusammen mit Spring interessiert hat und ich die Seminararbeit ebenfalls als Gelegenheit nutzen wollte um mich mit Teilen von diesem Framework auseinanderzusetzen habe ich mein spezielles Augenmerk darauf gerichtet.

4 Anwendung

4.1 Frameworks

Die Anforderung an diese Seminararbeit war, dass Hibernate genutzt wird. Zusätzlich habe ich mich mit Spring auseinandergesetzt. Speziell nutzte ich für diese Arbeit Spring MVC und Spring Data. Um die Darstellung der Web Oberfläche einfach und ansprechend zu gestalten habe ich Bootstrap [?] benutzt.

4.2 Architektur

Die Anwendung Besteht aus einer Webapplikation. Ich habe ein Model-View-Controller Ansatz gewählt welcher mit dem Framework Spring MVC umgesetzt wurde. Web Anfragen werden von Spring entgegen genommen und an die Controller Klassen weitergeleitet, welche die Anfrage bearbeiten. Für das ORM habe ich Spring Data JPA und Hibernate verwendet.

4.2.1 Dependencies

Ich habe ein Maven Projekt gemacht, weshalb sämtliche Dependencies im pom.xml hinterlegt wurden.

4.2.2 Struktur

Der Source Code ist grundsätzlich in 3 Ordnern untergebracht:

- *java*: hier sind sämtliche JAVA Klassen in verschiedenen Paketen untergebracht.
- *resources*: hier werden Konfigurationsdateien gespeichert
- *webapp*: hier sind die für die Web Applikation benötigten Dateien (jsp, css, js) und das web.xml gespeichert

4.2.2.1 Pakete

Ich habe für die Anwendung den Paket Prefix ch.zhaw.schilram genutzt. Die Anwendung hat den Namen sem_hib. Die JAVA Klassen sind in verschiedene Pakete unterteilt.

model

Im Paket *model* sind die Model Klassen untergebracht welche von Hibernate für das ORM Mapping genutzt werden.

repository

Im Paket *repository* sind sämtliche Sämtliche Interfaces welche das JpaRepository Interface implementieren und für den Zugriff auf die Datenbank dienen.

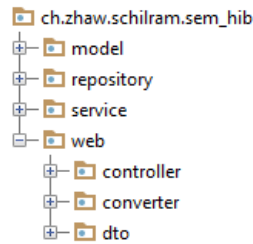


Abbildung 4.1: Paketstruktur

service

Das Paket *service* beinhaltet für jede Model Klasse eine Service Klasse welche die Methoden für den Zugriff auf die Datenbank ermöglicht.

web.controller

Im Paket *web.controller* sind die Controller Klassen abgelegt welche die Web Requests entgegennehmen und verarbeiten

web.converter

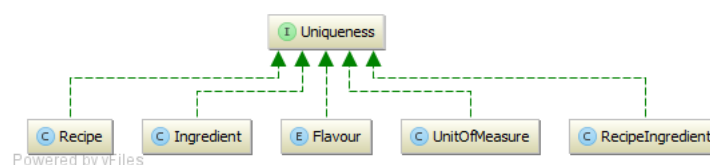
Hier sind einerseits die Converter Klassen gespeichert, welche statische Methoden zur Umwandlung einer Model Klasse in die entsprechende DTO Klasse anbieten, sowie auch die Converter, welche gebraucht werden um über in Web Formularen als String übermittelte ID das zugehörige persistierte Objekts zu finden.

web.dto

Im Pakte *web.dto* sind die DTO bzw. Formular Klassen welche für die Formulareingabe genutzt werden abgelegt.

4.2.3 Klassendiagramm

Unten sind die Klassendiagramme der Pakete *model* und *service* aufgeführt.

4.2.4 model KlassendiagrammAbbildung 4.2: Klassendiagramm Paket *model*

4.2.5 service Klassendiagramm

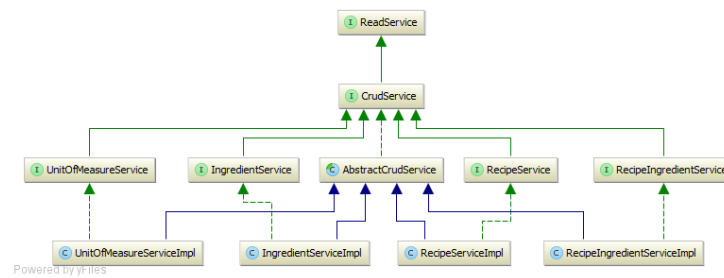


Abbildung 4.3: Klassendiagramm Paket *service*

4.3 Installation

4.3.1 Voraussetzungen

4.3.2 Datenbank einrichten

4.3.3 Tomcat einrichten

5 Abbildungsverzeichnis

4.1	Paketstruktur	7
4.2	Klassendiagramm Paket <i>model</i>	7
4.3	Klassendiagramm Paket <i>service</i>	8

6 Literaturverzeichnis

- [1] Github. Github. <https://github.com>, 06 2013. aufgerufen am 11. Juni 2013.
- [2] The PostgreSQL Global Development Group. Postgresql. <http://www.postgresql.org/>, 2013. aufgerufen 6. Juni 2013.
- [3] JetBrains. IntelliJ idea. <http://www.jetbrains.com/idea/>, 06 2013. aufgerufen am 11. Juni 2013.
- [4] Petri Kainulainen. *Spring Data*. Packt Publishing, Packt Publishing Ltd Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2012. ISBN 978-1-84951-904-5.
- [5] texniccenter.org. Texniccenter. <http://www.texniccenter.org/>, 2013. aufgerufen 6. Juni 2013.