

Seminar Hibernate

Seminar Hibernate

ZHAW - Zürcher Hochschule für Angewandte Wissenschaften

Ramon Schilling

`schilram@students.zhaw.ch`

11. Juni 2013

Inhaltsverzeichnis

1	Einleitung	3
1.1	Aufgabenstellung	3
1.2	Deliveries	3
1.3	Zielsetzung	3
1.4	Motivation	3
2	Projektmanagement	4
2.1	Projektplanung	4
2.2	Hilfsmittel	4
2.3	Aufwand	4
2.4	Hilfsmittel	4
2.4.1	Versionskontrolle	4
2.4.2	Dokumentation	4
2.4.3	Programmierung	5
2.4.4	Datenbank	5
3	Einarbeitung	6
3.1	Vorgehen	6
4	Anwendung	7
4.1	Frameworks	7
4.2	Architektur	7
4.2.1	Dependencies	7
4.2.2	Struktur	7
4.2.3	Klassendiagramm	8
4.2.4	model Klassendiagramm	8
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	10
6	Literaturverzeichnis	11

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 einzelnene 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

\TODO:

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 Hilfsmittel

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.

2.3 Aufwand

Der Aufwand für die Seminararbeit sollte 50 h betragen.

Beschreibung	Soll	Ist
Einarbeitung in Hibernate	8 h	30 h
Einarbeitung in Spring	8 h	9 h
Einrichten der Entwicklungsumgebung inkl. Hibernate Test	4 h	12 h
Programmieren der Anwendung	24 h	5 h
Dokumentation	6 h	14 h
Total	154 h	158 h

2.4 Hilfsmittel

2.4.1 Versionskontrolle

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

2.4.2 Dokumentation

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

2.4.3 Programmierung

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

2.4.4 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.

4.2 Architektur

4.2.1 Dependencies

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.sem_hib` genutzt. Die Anwendung hat den Namen `sem_hib`. Die JAVA Klassen sind in verschiedene Pakete unterteilt.

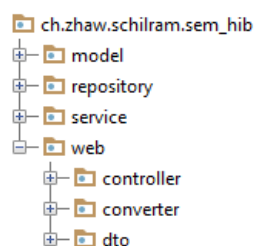


Abbildung 4.1: Paketstruktur

model

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

repository

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

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 Klassendiagramm

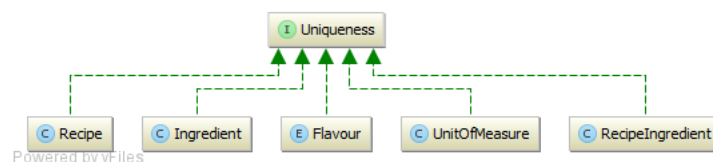


Abbildung 4.2: Klassendiagramm Paket *model*

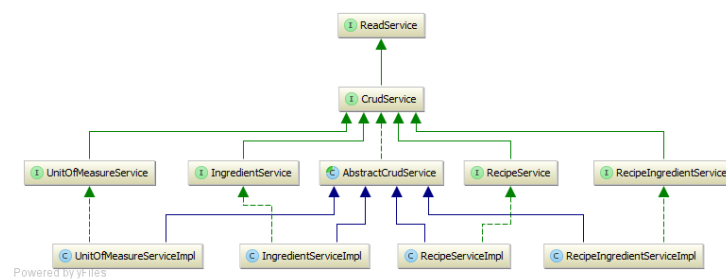
4.2.5 service Klassendiagramm

4.3 Installation

4.3.1 Voraussetzungen

4.3.2 Datenbank einrichten

4.3.3 Tomcat einrichten

Abbildung 4.3: Klassendiagramm Paket *service*

5 Abbildungsverzeichnis

4.1	Paketstruktur	7
4.2	Klassendiagramm Paket <i>model</i>	8
4.3	Klassendiagramm Paket <i>service</i>	9

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.