Universität Bayreuth
Fakultät für Informatik

Bachelorarbeit

im Studiengang Informatik

zur Erlangung des akademischen Grades
Bachelor / Master of Science

Thema:	Integration of JPA-conform ORM-Implementations in Hibernate Search
Autor:	Martin Braun <martinbraun123@aol.com> Matrikel-Nr. 1249080
Version vom:	July 15, 2015
1. Betreuer:	Dr. Bernhard Volz
2. Betreuer:	Prof. Dr. Bernhard Westfechtel

Zusammenfassung

Abstract

Contents

List of Figures

List of Tables

Listingverzeichnis

1 Introduction

1.1 An overview of different database paradigmas

When it comes to persisting data in applications, nowadays there are a lot of different paradigmas around that one has to choose from. Following is a short overview over the most common ones.

1.1.1 relational databases

1.1.2 NoSQL databases

1.1.3 Object databases

1.2 Object-relational impedance mismatch

While the NoSQL approach is undeniably rising in popularity, the demand for relational solutions is still unmatched as it has been used and proven in practice for many years now.

Nowadays, many popular languages like Java, C#, etc. are object-oriented. While SQL solutions for querying relational databases exist for these languages, there is a big discrepancy between the relational and the object-oriented paradigm. [1]

This is where Object Relational Mappers (ORM) come into use. They map tables to entities (classes) and enable users to write queries against classes instead of tables. This is especially useful if used in big software products as not all programmers have to know the exact details of the underlying database. The database system could even be completely replaced for another with the business logic still being in place.

[1] Wikipedia, see [1]

1.2.1 JPA

1.2.2 fulltext search

1.2.3 compatibility of Hibernate Search with JPA

1.2.4 aims of this thesis

2 Showcase of basic Hibernate Search

3 Ausblick

4 Fazit

Abbildung ?? [S.??]

Überschrift 1	Überschrift 2
Info 1	Info 2
Info 3	Info 4

```
<dataConfig>
  <dataSource type="JdbcDataSource"
              driver="com.mysql.jdbc.Driver"
              url="jdbc:mysql://localhost/bms_db"
              user="root"
              password=""/>
  <document>
    <entity name="id"
        query="select id, htmlBody, sentDate, sentFrom, subject, textBody
        from mail">
    <field column="id" name="id"/>
    <field column="htmlBody" name="text"/>
    <field column="sentDate" name="sentDate"/>
    <field column="sentFrom" name="sentFrom"/>
    <field column="subject"  name="subject"/>
    <field column="textBody" name="text"/>
    </entity>
  </document>
</dataConfig>
```

Listing 1: Die Datei `data-config.xml` dient als Beispiel für XML Quellcode

```
/* generate TagCloud */
Cloud cloud = new Cloud();
cloud.setMaxWeight(_maxSizeOfText);
cloud.setMinWeight(_minSizeOfText);
cloud.setTagCase(Case.LOWER);

```

```
7  /* evaluate context and find additional stopwords */
8  String query = getContextQuery(_context);
9  List<String> contextStoplist = new ArrayList<String>();
10 contextStoplist = getStopwordsFromDB(query);
11
12 /* append context stoplist */
13 while(contextStoplist != null && !contextStoplist.isEmpty())
14   _stoplist.add(contextStoplist.remove(0));
15
16 /* add cloud filters */
17 if (_stoplist != null) {
18   DictionaryFilter df = new DictionaryFilter(_stoplist);
19   cloud.addInputFilter(df);
20 }
21 /* remove empty tags */
22 NonNullFilter<Tag> nnf = new NonNullFilter<Tag>();
23 cloud.addInputFilter(nnf);
24
25 /* set minimum tag length */
26 MinLengthFilter mlf = new MinLengthFilter(_minTagLength);
27 cloud.addInputFilter(mlf);
28
29 /* add taglist to tagcloud */
30 cloud.addText(_taglist);
31
32 /* set number of shown tags */
33 cloud.setMaxTagsToDisplay(_tagsToDisplay);
```

Listing 2: Das Listing zeigt Java Quellcode

Die Zuordnung aller möglichen Werte, welche eine Zufallsvariable annehmen kann nennt man *Verteilungsfunktion* von X.

> Die Funktion F: $\mathbb{R} \to [0,1]$ mit $F(t) = P(X \leq t)$ heißt Verteilungsfunktion von X.[2]

> Für eine stetige Zufallsvariable $X : \Omega \to \mathbb{R}$ heißt eine integrierbare, nichtnegative reelle Funktion $w : \mathbb{R} \to \mathbb{R}$ mit $F(x) = P(X \leq x) = \int_{-\infty}^{x} w(t)dt$ die *Dichte* oder *Wahrscheinlichkeitsdichte* der Zufallsvariablen X.[3]

[2] Konen, vgl. [**?**] [S.55]
[3] Konen, vgl. [**?**] [S.56]

Literaturverzeichnis

Anhang

Literaturverzeichnis

[1] Object-relational impedance mismatch, Wikipedia `https://en.wikipedia.org/wiki/Object-relational_impedance_mismatch`, 07/15/2015

Eidesstattliche Erklärung

Eidesstattliche Erklärung zur <-Arbeit>

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

$Unterschrift$: $Ort, Datum$: