Lernjournal

Java Zertifizierung

Autor: Ramon Schilling

Dozent: Michael Reiser

04. Juni 2013

Inhalt

Applikation

Die Idee der Applikation ist eine einfache Rezeptverwaltung.

Es kann eine Liste mit allen Zutaten und eine Liste mit allen Rezepten angezeigt werden. Ausserdem kann ein Prozess ausgelöst werden, um Zutaten oder Rezepte welche in Files gespeichert sind aus einem Verzeichnis auszulesen.

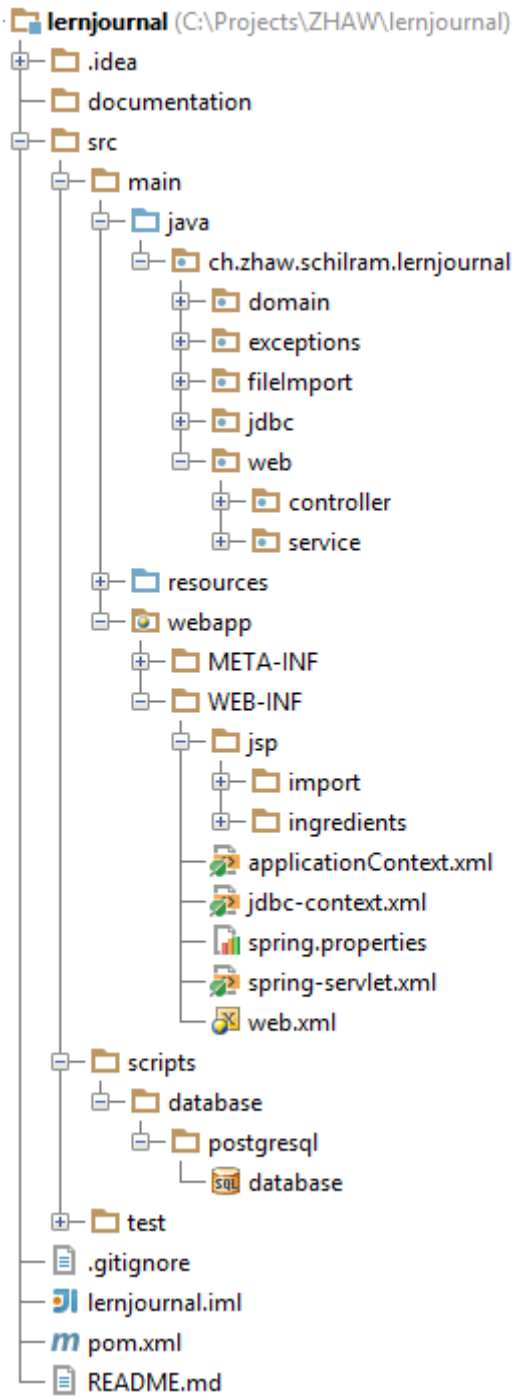
Die Files müssen jeweils die Endung .ingredient oder .recipe haben und darin muss jeweils eine Zutat oder ein Rezept gespeichert sein.

Verwendete Features

Grundsätzlich ging es darum die Anforderungen umzusetzen und die im Kurs gewonnenen Kenntnisse zu nutzen. Für die Webapp habe ich Spring eingesetzt. Mit Spring MVC habe ich eine einfache Möglichkeit die Web Requests zu handeln. Für diesen Teil der Applikation habe ich ausserdem noch SpringJDBC genutzt.

Ich habe aber bewusst darauf geachtet für den file Import das java.sql API zu nutzen um damit Erfahrungen zu sammeln.

Package Aufbau



Die Java Klassen sind in folgende Packeges unterteilt:

domain – Domain Klassen welche in der Datenbank persistiert werden und von der Klasse StorableItem abgeleitet sind.

exceptions – Veschiedene Exceptions

fileimport – alle für den Import aus Files benötigten Klassen

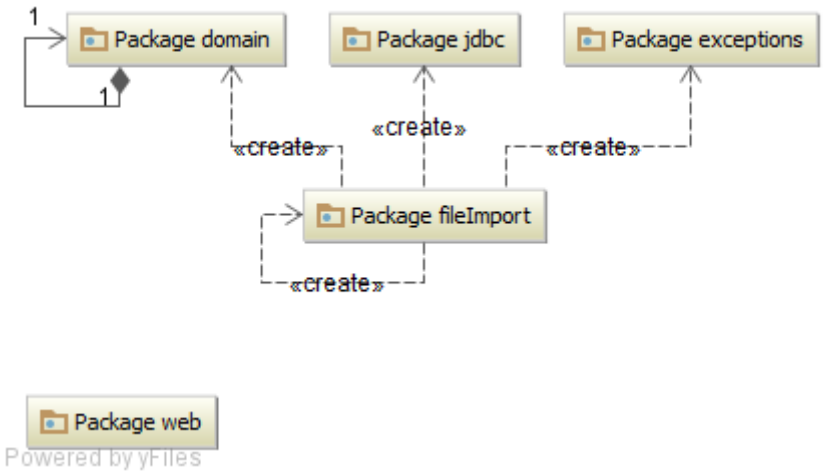
jdbc – Klassen welche für die Verbindung mit der Datenbank genutzt werden

web/controller – Controller Klassen welche die Webanfragen entgegennehmen

web/service – Service Klassen mit welche mit SpringJDBC auf die Datenbank zugegriffen wird

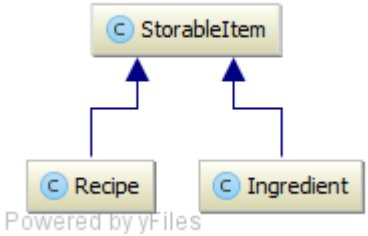
Im Ordner webapp sind Konfigurationsdateien z.B. für Spring sowie die web.xml und sämtliche jsp Dateien gespeichert.

Packagediagramm

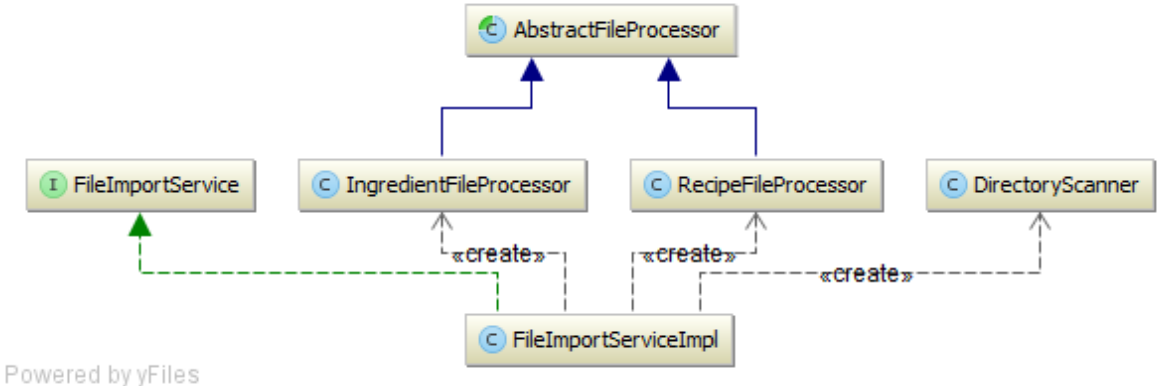


Klassendiagramme

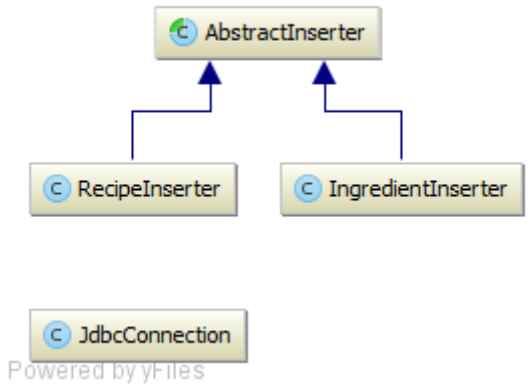
Package domain



Package fileImport



Package jdbc



Importieren von Files

Struktur der Files

Ingredient

Name; Beschreibung; Geschmack (durch Semikolon getrennt)

Wobei Geschmack eine der 5 Geschmacksrichtungen *SALTY, SWEET, SOUR, BITTER* und *UMAMI* sein kann. Beim Geschmack spielt die Gross/Kleinschreibung keine Rolle. Wird aber ein Geschmack angegeben der nicht existiert wird er ignoriert.

Recipe

Name; Beschreibung; Instruktion (durch Semikolon getrennt)

Installation und Start der Anwendung

Um die Applikation lauffähig zu haben muss Tomcat als Webserver laufen und PostgreSQL mit der Datenbank eingerichtet sein.

Voraussetzung:

Webserver

lauffähiger Tomcat Server (kann unter http://tomcat.apache.org/ heruntergeladen werden).

DatenbankServer

PostgreSQL (kann unter http://www.postgresql.org/download/ heruntergeladen werden)

Einrichten der Datenbank

In der Datenbank muss ein User erstellt werden und die für die Applikation nötige Datenbank.

Für die Applikation wird der User „*lernjournal*“ mit dem Passwort „*lernjournal*“ auf der Datenbank „*lernjournal*“ genutzt.

Zuerst muss also der User angelegt werden. Dies kann wahlweise über das GUI oder direkt über das SQL Statement

```
CREATE ROLE lernjournal LOGIN ENCRYPTED PASSWORD
'md56b9e139f33472d38325f9069dca0fa16'
  VALID UNTIL 'infinity';
```

gemacht werden.

Danach muss die Datenbank und die Tabelle *ingredients* angelegt werden. Dafür kann wahlweise folgendes SQL Statement genutzt werden.

```
-- Database: lernjournal
-- DROP DATABASE lernjournal;
CREATE DATABASE lernjournal
  WITH OWNER = lernjournal
    ENCODING = 'UTF8'
    TABLESPACE = pg_default
    LC_COLLATE = 'German_Switzerland.1252'
    LC_CTYPE = 'German_Switzerland.1252'
    CONNECTION LIMIT = -1;


-- Table: ingredients
-- DROP TABLE ingredients;
CREATE TABLE ingredients
(
  id serial NOT NULL,
  name character varying(32),
  description text,
  flavour character varying(32),
  CONSTRAINT ingredient_id PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE ingredients
  OWNER TO lernjournal;
```

```
-- Table: recipes
-- DROP TABLE recipes;
CREATE TABLE public.recipes
(
  id serial,
  name character varying(64),
  description character varying(255),
  instructions text,
  CONSTRAINT recipe_id PRIMARY KEY (id)
)
WITH (
OIDS = FALSE
)
;
ALTER TABLE public.recipes
OWNER TO lernjournal;
```

Tomcat - Deploy Application

Das *lernjournal.war F*ile kann direkt im Tomcat depolyed werden. Dal File *lernjournal.war* ist im Verzeichnis *target* zu finden.

Starten der Anwendung

Wenn alle Installationspunkte beachtet wurden, kann nun über http://localhost:8080/lernjournal/lernjournal/ auf die Applikation zugegriffen werden. Wichtig ist, dass sowohl Tomcat als auch PostgreSQL laufen.

Schlusswort

Grundsätzlich finde ich es gut, dass wir alles gelernte einmal zusammen einsetzen mussten, auch wenn es dadurch z.T. etwas konstruierte Situationen gibt. Ich habe es gut gefunden, dass man sich vertieft mit der Materie befassen musste um so einen besseren Einblick zu erhalten. Es gab mir ausserdem die Gelegenheit Spring etwas anzuschauen, obwohl ich natürlich nur an der Oberfläche kratzen konnte.

Für den Kurs fände ich persönlich es besser, wenn der gesamte Lehrstoff schneller durchgenommen würde. Dadurch, dass der Stoff über das ganze Semester verteilt wird, hat man zwar immer wieder Zeit um an dem Lernjournal zu arbeiten. Da man aber doch relativ viel Wissen aus den „späten“ Vorlesungen braucht, kann man am Anfang nicht wirklich vorwärts machen. Ich hätte es deshalb begrüsst mehr Stoff in kürzerer Zeit durchzunehmen um dann wenn man alle Kenntnisse hat mit dem Lernjournal loslegen zu können.