

aFH Erfurt
Angewandte Informatik
Module: Programmierung Java 2
Dozent:

Allgemeine Projektdokumentation

Student_self_organization-Service

vorgelegt von:

Felix Zwicker - DevLead

Matr.-Nr.: 120048201

E-Mail: felix.zwicker@fh-erfurt.de

Friedemann Taubert

Matr.-Nr.: 120060157

E-Mail: friedemann.taubert@fh-erfurt.de

Maximilian Keller

Matr.-Nr.: 120063608

E-Mail: maximilian.keller@fh-erfurt.de

Jenny Wagner

Matr.-Nr.: 120061365

E-Mail: jenny.wagner@fh-erfurt.de

Tim Eisenberg

Matr.-Nr.: 120066175

E-Mail: tim.eisenberg@fh-erfurt.de

Abgabetermin: 07. August 2022

Inhaltsverzeichnis

1. Allgemeines
2. Was macht der Service?
3. Projekt Struktur
4. Die wichtigsten Klassen
 - 4.1. Allgemein
 - 4.2. Faq
 - 4.3. Forum
 - 4.4. ToDo Liste
5. Tests

1. Allgemeines

Dieses Projekt wurde im Rahmen des Moduls "Programmierung Java 2" erstellt und ist ein fortgeführtes Projekt vom Modul "Programmierung Java 1".

Eingeordnet waren wir im Team 2 Service 5, der Studentischen Selbstverwaltung. Unser Auftrag bestand daraus eine Service zu erstellen, mit dessen Hilfe Studenten sich gegenseitig unterstützen können und wichtige/hilfreiche Informationen erhalten. Im ersten Teil erstellten wir die Logik und im jetzigen Modul die dazugehörige Datenbankbindung.

Teil des Teams waren Friedemann Taubert, Jenny Wagner, Maximilian Keller, Tim Eisenberg und Felix Zwicker.

2. Was macht der Service?

Da uns keine klare Aufgabe zugeteilt wurde, entschieden wir uns, mit vorheriger Rücksprache ein Forum, Faq und eine ToDo Liste im Bereich der Studentischen Selbstverwaltung zu erstellen.

Im Faq werden die am häufigsten und wichtigsten Fragen von Studenten gesammelt und zur Verfügung gestellt. Dabei gibt es keine Interaktionsmöglichkeit. Es dient lediglich zur vereinfachten Informationsbeschaffung.

Im Forum können Fragen gestellt und beantwortet werden. Es ist eine Möglichkeit der gegenseitigen Unterstützung und kann von Studenten wie Professoren genutzt werden. Es könnte von der Funktionsweise mit Reddit verglichen werden.

Die ToDo Liste bietet Studenten die Möglichkeit, ihren Alltag durch verschiedene Einträge zu organisieren. Hier können eigene Notizen erstellt, bearbeitet und gelöscht werden.

3. Projekt Struktur

Der Service wurde als Maven Projekt erstellt und besitzt das Hauptmodul "student_self_organization-service".

In diesem Modul sind alle Klassen und Funktionen unseres Service gelagert.

Unterteilt ist der Service weiter in Core, Storage und Resource. Core beinhaltet unsere models, enums und exceptions, Storage die JPA-Datenbankanbindung mit den Grundlegenden Funktionen und Resource alle nötigen Klassen für die Rest-API. Zusätzlich gibt es noch das Package Util, dieses beinhaltet lediglich einige Testdaten.

Das Modul "student_self-organization-client" ist entfallen, da kein anderer Service Funktionen von uns benötigte.

4. Die wichtigsten Klassen

4.1. Allgemein

RepositoryImpl:

Die RepositoryImpl implementiert die Repository Interfaces und gibt die Funktionalitäten durch Zugriff auf die DAOs.

BaseResource:

Die BaseResource ist der Einstiegspunkt der REST Schnittstelle. Sie ist über den Pfad "<http://localhost:8080/api>" erreichbar. Außerdem kann sie mit den anderen Pfaden auf bestimmte Ressourcen verweisen.

IGenericDao:

IGenericDao ist ein Interface, welche eine sichere Nutzung der Methoden ermöglicht. Das Interface ist auch in der Klasse JpaGenericDao zu finden, da sie diese Erweiterung hat.

RepositoryFactory:

Ist das Kernstück und verwaltet die Repositories, die DAOs und die EntityManagerFactory, welche mehr oder weniger als Datenbank betrachtet werden kann. Da diese EntityManagerFactory so groß ist, handelt es sich um Singleton und ein Factory Pattern, da es Instanzen von dem DAO und den Repos ausgibt.

4.2. Faq

Element:

Die Klasse Element wird genutzt um die Inhalte des FAQs zu erstellen, indem man ihr die benötigten Daten übergibt. Dabei wird das Builder-Pattern benutzt, um das Erstellen und Updaten zu vereinfachen. Besonders wenn in einem späteren Stand des Services verschiedene Attribute für ein FAQ-Element benötigt oder nicht benötigt werden. Zudem besitzt die Klasse einen Override toString, um das Arbeiten mit der Klasse zu vereinfachen.

IFaqDao:

Dieses Interface beinhaltet, als Erweiterung zur IGenericDao, noch Spezifische Funktionen für das Faq.

4.3. Forum

ForumRepository:

Im Forum gibt es die Möglichkeit neue Fragen zu erstellen bzw. bestehende zu löschen. Wichtige Bestandteile des Forums sind hierbei die Antwort, die Nachricht und die Frage an sich.

Answer, Message und Question:

Beinhalten alle die AbstractDatabaseEntity und verfügen über Set- und Get-Methoden.

4.4. ToDo Liste

Task:

Die Klasse Task wird genutzt, um den Inhalt des Service Todo zu erstellen. Hierfür wird das Builder Pattern verwendet, was das Erstellen und Updaten vereinfacht. Durch den Override toString wird das Arbeiten mit der Klasse vereinfacht.

TodoRepository:

Das TodoRepository gibt die Möglichkeit neue Tasks zu erstellen, sowie bestehende zu löschen. Außerdem ermöglicht es, alle oder einen bestimmten Task zu bekommen. Zudem kann ein Task angepasst beziehungsweise geupdated, sowie nach Checked, Unchecked oder Priorität sortiert werden.

TaskResource:

TaskResource ermöglicht CRUD Operations für eine Todo-Webapplikation.

5. Tests

Die Test für unseren Service fallen größtenteils trivial aus, weshalb wir nur für "Question" Tests geschrieben haben.