

Essensliste

PHP/DB Schulprojekt 2019

Michael Vogel

E2FI4

Table of Contents

Beschreibung.....	3
Quellcode.....	3
Dateien.....	3
CSS.....	3
Config.....	3
Controller.....	3
AppFixtures.....	3
Entities.....	3
Forms.....	4
Repository.....	4
DoctrineConfig.....	4
Security.....	4
Templates.....	4
Base.....	4
Entry.....	4
Registration.....	4
Security.....	4
User.....	4
Tabellen.....	5
User.....	5
Meal.....	5
Workday.....	5
Entry.....	5
Probleme und Unerfülltes.....	5

Beschreibung

Es wurde eine Seite entwickelt an der man sich anmelden kann und für unterschiedliche Tage zum Essen eintragen kann. Die Idee ist dadurch entstanden das schon länger geplant war so etwas in unserem Unternehmen zu entwickeln damit die Mitarbeiter sich leichter zum Mittagessen eintragen können ohne das sie physikalisch anwesend sein müssen und ganz viel Papier verbraucht wird.

Quellcode

[Projekt auf Github](https://github.com/Vogaeael/dessertlist) (<https://github.com/Vogaeael/dessertlist>)

Dateien

CSS

Die Scss-Dateien unter Assets sind dazu da die css Dateien zu erstellen. Sie sind unterteilt in Allgemeine Sachen und einzelne Seiten.

Config

Die ganzen Yaml-Dateien geben Symfony an wie er etwas handhaben soll, wo welche Dateien liegen, welche Url's von welchem Controller gehandhabt werden.

Das resetDatabase.script habe ich nur erstellt um einfacher Die Datenbank neu zu bauen, da ich wegen eines Fehlers mit Usern löschen öfters testen musste.

Controller

Controller kümmern sich um die ganzen aufrufe auf eine Seite. Sie entscheiden was je nach Aufruf angezeigt wird. Es gibt einen BaseController der alle Methoden beinhaltet die mehrere Controller gebrauchen und ist parent von allen anderen. Der EntryController kümmert sich um alle Aufrufe bezüglich der Entries, also das Erstellen, Bearbeiten, Löschen und Anzeigen. Der IndexController kümmert sich um die Index-Seite, der RegistrationController um die Registration-Seite und der SecurityController um die Login-Seite. Der UserController kümmert sich um alle Aufrufe für User-Seiten, also User bearbeiten, löschen und anzeigen.

AppFixtures

AppFixtures ist dazu da um Zufällige Daten in die Datenbank einzutragen damit man besser testen kann.

Entities

Entity-Klassen stellen Einträge aus der Datenbank dar. So stellt die Entry-Klasse die Einträge aus der Entry-Tabelle, die Meal-Klasse die Einträge aus der Meal-Tabelle, die User-Klasse die Einträge aus der User-Tabelle und die Workday-Klasse die Einträge aus der Workday-Tabelle da.

Forms

Die FormTypes sind dazu da die Forms zu erstellen. EntryType erstellt eine Form um Essenseinträge zu erstellen oder zu bearbeiten. RegistrationFormType erstellt wiederum die Form zum registrieren. Die UserType erstellt die Form zum erstellen und bearbeiten von Usern.

Repository

Repository-Klassen sind dazu da Daten aus der Datenbank zu holen. Für jede Tabelle gibt es eine eigene Klasse. Natürlich erben alle von einer Klasse damit es nicht haufenweise doppelten Code gibt.

DoctrineConfig

Die .orm.yml Dateien sind dazu da Symfony zu zeigen wie welche Entity-Klasse mit der dazugehörigen Tabelle verbunden ist, welche Spalte welche Variable darstellt usw.

Security

Der LoginFormAuthenticator handelt das man nicht auf Seiten auf die man nicht sollte zugreifen kann und auf die Login Seite weiter geleitet wird.

Templates

Base

Die base.html.twig ist dazu da das html welches alle Seiten beinhalten dazustellen.

Die index.html.twig im base Ordner ist dazu da das html von der Startseite dazustellen

Entry

Die .html.twig Dateien unter dem Pfad entry sind dazu da das html der Seiten bezüglich der Essenseinträge dazustellen, dazu gehören create für das erstellen, edit für das bearbeiten, view um sich ein einzelnes anzusehen und viewAllByUser um sich alle Einträge eines Users anzusehen.

Registration

Die register.html.twig Datei ist dazu da das html der Registration-Seite anzuzeigen.

Security

Die login.html.twig Datei ist dazu da das html für die Login-Seite dazustellen.

User

Die .html.twig Dateien unter dem Pfad user sind dazu da das html der Seiten bezüglich der Benutzer dazustellen. Dazu gehörten edit fürs bearbeiten, view zum ansehen einzelner User und viewAll zum ansehen aller User. Diese wurden aber nicht realisiert.

Tabellen

User

Der User mit den Attributen, id als Identifikator, name für seinen Namen, email für seine Email, roles welche Rollen er hat (bisher gibt es nur user) und password welches sein gehashtes Passwort ist.

Meal

Das Gericht mit den Attributen, id als Identifikator, name für seinen Namen, link falls es im Internet eine Anleitung oder Beschreibung dafür gibt und description falls eine Beschreibung selber angehängt sein soll.

Workday

Der Arbeitstag an den man sich zum Essen eintragen lassen kann. Er hat die folgenden Attribute. Id als Identifikator, date an welchem Datum er ist und mealId, damit ein Gericht dem Tag zugeordnet werden kann.

Entry

Ein Eintrag in die Liste. Die Tabelle hat die folgenden Attribute. Id als Identifikator, note falls man eine Notiz (zB. Bitte ohne Fleisch) hinzufügen möchte, userId welchem User sie zugeordnet ist und workdayId, an welchem Tag man sich eintragen will.

Probleme und Unerfülltes

Zuerst wurde versucht das ganze Projekt mit dem Mini-Framework Silex aufzubauen. Wegen vielen Problemen und da Silex schon länger nicht mehr unterstützt wird ist man dann umgestiegen auf Symfony4.

Da man keine User löschen kann während sie noch angemeldet sind wurde das löschen von Usern komplett entfernt, da es auch keine Anforderung war. Das Bearbeiten von Usern war ursprünglich auch geplant ist aber wegen zu vieler Probleme und da es nicht angefordert ist ebenfalls weggefallen.

Was ebenfalls nicht verwirklicht wurde ist eine Seite wo man sehen kann wer sich alles für einen gewissen Tag eingetragen hat oder auch eine Seite in der man sich die Gerichte genauer ansehen kann.