

SCR4 Gruppe 2 – SS 2020

Übungsprojekt - Einkaufslist

Michael Eder

3. Mai 2020

Anleitung

1.1 Test User

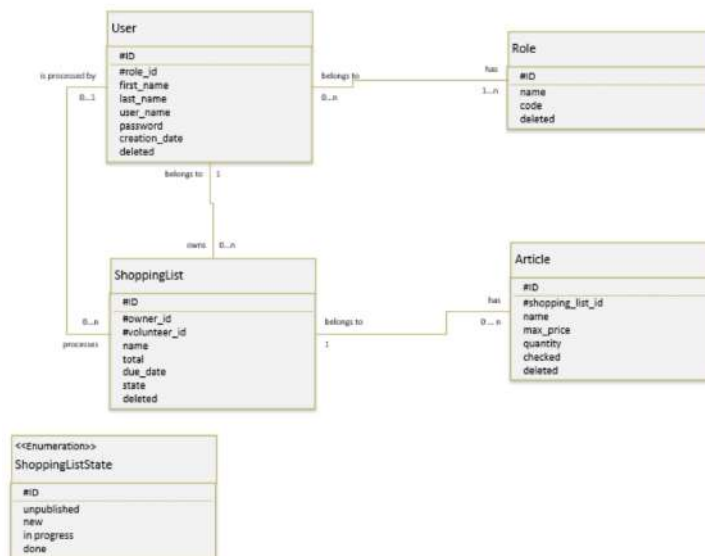
ID	Username	Password	Active
1	meder	welcome1234!	yes
2	wferro0	welcome1234!	yes
3	amcgroarty1	welcome1234!	yes
4	rmccreagh2	welcome1234!	yes
5	cpendrey3	welcome1234!	yes
6	bmoore	welcome1234!	no

1.2 Dump

Im Folder **/docs/dump** findet sich die ein Dumpfile mit den entsprechenden Testdaten. Datenbankname ist wie vereinbart **fh_2020_scm4_S1810307008**. Benutzername und Password **fh_2020_scm4**. Für die Datenbankverbindung ist bereits ein **.env** File beigelegt, die die nötigen Einträge enthält. Es muss lediglich die Dumpfile importiert werden und dann kann gestartet werden.

1.3 ERD

Das **ERD-Diagramm** ist auch als File beigelegt.



2. Logging

Die Logfiles werden automatisch generiert. Sollte noch kein Folder logs existieren, so legt die Applikation eigenständig ein neues Verzeichnis logs in **htdocs** an. Die Logs werden dann je Datum wiederum in Folder unterteilt. Für jeden Tag wird ein neuer Folder angelegt, um es übersichtlicher zu gestalten und spätere Logauswertungen zu vereinfachen. Innerhalb der Files kann man dann die IP, den Benutzer, einen Zeitstempel (UTC), die Aktion / Nachricht und den Loglevel sehen.

Name	Date modified	Type	Size
bookstore	27/04/2020 09:19	File folder	
dashboard	27/04/2020 09:15	File folder	
img	27/04/2020 09:15	File folder	
logs	03/05/2020 09:22	File folder	
shoppinglist	01/05/2020 16:04	File folder	
webolizer	27/04/2020 09:15	File folder	
xampp	27/04/2020 09:15	File folder	
applications	27/08/2019 16:02	Chrome HTML Do...	4 KB
bitnami	27/08/2019 16:02	CSS File	1 KB
favicon	16/07/2015 17:32	Icon	31 KB
index	27/02/2020 14:32	PHP Source File	1 KB

Name	Date modified	Type
1_5_2020	01/05/2020 16:26	File folder
2_5_2020	02/05/2020 12:10	File folder
3_5_2020	03/05/2020 09:22	File folder

```

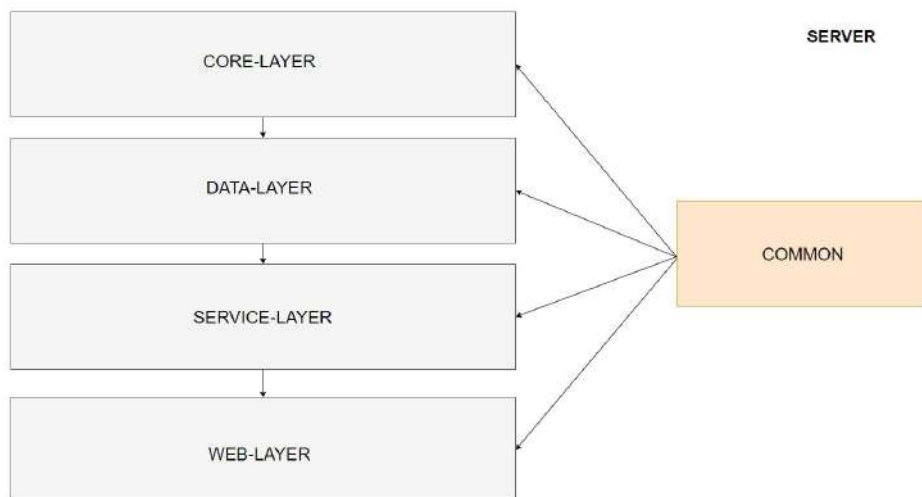
559 [INFO]-[Sunday, 03-May-20 11:39:50 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Get all shopping lists for owner]
560 [INFO]-[Sunday, 03-May-20 11:39:58 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Shopping list with ID 4 deleted!]
561 [INFO]-[Sunday, 03-May-20 11:39:58 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Get all shopping lists for owner]
562 [INFO]-[Sunday, 03-May-20 11:40:00 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Shopping list with ID 5 deleted!]
563 [INFO]-[Sunday, 03-May-20 11:40:00 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Get all shopping lists for owner]
564 [INFO]-[Sunday, 03-May-20 11:40:42 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Shopping list with ID 7 editing started and state set to UNPUBLISHED!]
565 [INFO]-[Sunday, 03-May-20 11:40:44 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Get all shopping lists for owner]
566 [INFO]-[Sunday, 03-May-20 11:41:10 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Shopping list with ID 7 editing started and state set to UNPUBLISHED!]
567 [INFO]-[Sunday, 03-May-20 11:41:11 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Shopping list with ID 7 published and set to state NEW!]
568 [INFO]-[Sunday, 03-May-20 11:41:11 UTC].-[ListController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Get all shopping lists for owner]
569 [INFO]-[Sunday, 03-May-20 11:41:13 UTC].-[UserController]-[IP :1]-[USER: amcgroarty1]-[ACTION: Logout]
570 [INFO]-[Sunday, 03-May-20 11:41:17 UTC].-[UserController]-[IP :1]-[USER: meder]-[ACTION: Login]
571 [INFO]-[Sunday, 03-May-20 11:41:17 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Get all shopping lists with state NEW]
572 [INFO]-[Sunday, 03-May-20 11:41:17 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Get all shopping lists with state NEW]
573 [INFO]-[Sunday, 03-May-20 11:41:52 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 taken over by volunteer with ID 1!]
574 [INFO]-[Sunday, 03-May-20 11:41:52 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Get all shopping lists with state IN PROGRESS!]
575 [INFO]-[Sunday, 03-May-20 11:42:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 processed!]
576 [INFO]-[Sunday, 03-May-20 11:42:18 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 12!]
577 [INFO]-[Sunday, 03-May-20 11:42:21 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Article with ID 33 checked!]
578 [INFO]-[Sunday, 03-May-20 11:42:22 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Article with ID 28 checked!]
579 [INFO]-[Sunday, 03-May-20 11:42:23 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Article with ID 27 checked!]
580 [INFO]-[Sunday, 03-May-20 11:42:42 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Article with ID 32 checked!]
581 [INFO]-[Sunday, 03-May-20 11:42:42 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Article with ID 31 checked!]
582 [INFO]-[Sunday, 03-May-20 11:42:42 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Article with ID 30 checked!]
583 [INFO]-[Sunday, 03-May-20 11:42:42 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Article with ID 29 checked!]
584 [INFO]-[Sunday, 03-May-20 11:42:42 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Article with ID 26 checked!]
585 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
586 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
587 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
588 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
589 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
590 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
591 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
592 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
593 [INFO]-[Sunday, 03-May-20 11:43:15 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 total price updated to 101.45!]
594 [INFO]-[Sunday, 03-May-20 11:44:02 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Shopping list with ID 7 set to state DONE!]
595 [INFO]-[Sunday, 03-May-20 11:44:02 UTC].-[ListController]-[IP :1]-[USER: meder]-[ACTION: Get all shopping lists with state DONE!]
596 [INFO]-[Sunday, 03-May-20 11:52:34 UTC].-[UserController]-[IP :1]-[USER: meder]-[ACTION: Logout]

```

3. Architektur

Es wurde sich für eine Layer-Architektur entschieden.

Architekturdiagramm



assets

Die Clientscripte finden sich im Folder **assets** unter **JS** wieder. Darin sind **AJAX** - Request und alle möglichen Hilfsscripte wie zB **Utils** enthalten. Des Weiteren findet man dort auch unter **media** alle verwendeten Bilder. Unter **main.css** findet sich das eigene Styling wieder, sowie unter **material.min.css** das CSS - Framework Material

Design Light, das im Rahmen des Projekts verwendet wurde.

server

Die gesamte Serverlogik findet sich im Folder **server** wieder. Hier wurde wie folgt unterteilt:

Common

Enthält alle Klassen, die in der gesamten Applikation über alle **Layer** verwendet werden. zB. **Logger**, **Routing**, **Util**

Core

Im **Core-Layer** befindet sich die Kernschicht der Applikation - **Domain** enthält alle Objekte der Domäne. In **Session** findet sich eine Kapselung der Session wieder.

Data

Im **Data-Layer** findet sich eine Abstraktion der Datenbank in **Database** wieder. **Mapper** ist für das Mappen von der Datenbankrückgabe in eine oder mehrere Domainentities zuständig. Zu guter Letzt findet sich auch noch **Repository** darin vor. Dies ist nach dem **Repository-Pattern** eine Abstraktionsschicht, um für jede Entity ein Repository mit CRUD Operationen zur Verfügung zu stellen. Die Datenbank ist als **Singleton** implementiert worden. Die Connection-Daten werden über ein **Environment-File** zur Verfügung gestellt. Damit das **.env-File** funktioniert, wird in **bootstrap.inc.php** das **.env-File** ausgelesen, geparsed und im superglobalen ENV Array gesetzt. **Database** liest entsprechend die Variablen aus. Damit muss die Applikation nichts mehr von der Datenschicht wissen und die Umgebung kann dies über ein **.env-File** zur Verfügung stellen. (siehe <https://12factor.net/de/>)

Service

Im **Service - Layer** findet sich nun die Services wieder. Diese inkludieren **Singletons** der **Repositories** und sind für die **Controller** im **Web-Layer** da. Sie enthalten die **Business-Logik**.

Web

Im **Web - Layer** befinden sich die **Controller** je **Domäne**. Die Controller setzen die Kommandos via **Services** um und retournieren die geparsten Resultate aus den unteren Schichten. Im Folder **HTTP** findet sich eine Implementierung für die **HTTP Status Codes** wieder. (zB. 400 BadRequest, 401 Unauthorized, etc.). Zu guter Letzt findet man im Folder **View** alle Scripte für die Views wieder. **Partials** sowie auch einzelne **Views**.

Im Projekt sieht die Aufteilung der Folderstruktur nun wie folgt aus.

