

Shoppinglist-Backend with SpringBoot

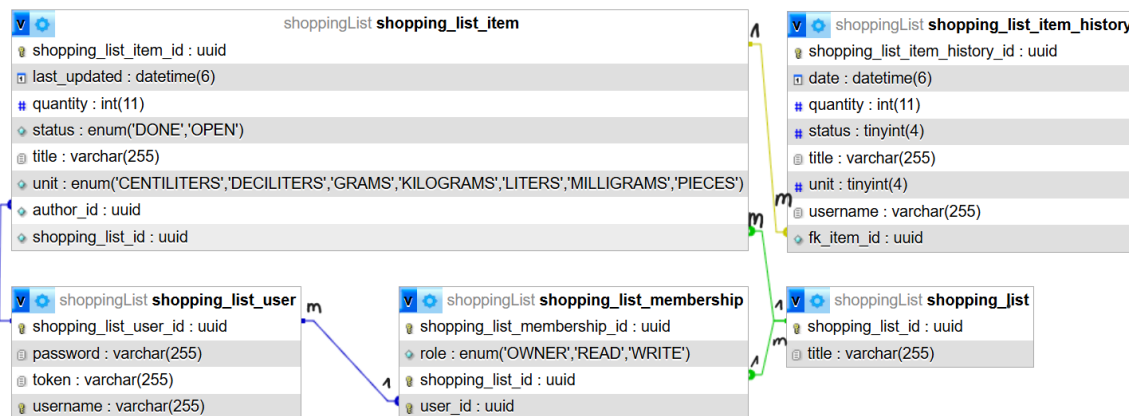
Inhaltsverzeichnis

Einleitung.....	2
Datenmodell UML	2
Testdaten	2
Testen mit Insomnia	3

Einleitung

Das Ziel dieses Projekt war es ein Backend (API) mit Java und Springboot zu erstellen, welches Daten direkt in eine Datenbank speichert. Wir haben uns dabei entschieden, ein Backend für eine Applikation zu bauen, mit der man miteinander Einkaufslisten erstellen, führen und verwalten kann. Dabei haben wir besonderen Fokus auf ein Rollensystem gelegt, bei dem der Ersteller einer Liste sagen kann, wer in die Liste schreiben und wer sie nur ansehen kann. Das Ziel von uns war es, dass Jemand der nicht die entsprechenden Rechte für eine Aktion hat an keine Daten herankommt, die er auf Grund seiner Rolle nicht sehen oder bearbeiten können soll. Ausserdem haben wir grossen Wert auf eine gute und sichere Authentifikation gelegt. Dafür speichern wir alle Passwörter gehasht in der Datenbank und verlangen von dem User nur einmal das Passwort. Danach wird er mit einem Session-Token authentifiziert.

Datenmodell UML



Testdaten

Hier haben wir wie im Dokument beschrieben einen CommandLineRunner implementiert, der für jede bei uns mögliche Klasse beim Starten der Anwendung 10 Datensätze einfügt. Dafür haben wir For-Schleifen verwendet, was das ganze beträchtlich einfacher gemacht hat. Hier ist ein Beispiel an Hand der User-Entity.

```
Argon2 argon2 = Argon2Factory.create();  
List<ShoppingListUser> users = IntStream.range(1, 11).boxed().flatMap(i -> {  
    ShoppingListUser u = new ShoppingListUser();  
    u.setUsername("user" + i);  
    u.setPassword(argon2.hash("password" + i));  
    u.setToken(UUID.randomUUID().toString());  
    return u;  
}).collect(Collectors.toList());  
  
userRepository.saveAll(users);
```

Testen mit Insomnia

Um unsere Endpunkte zu testen haben wir in Insomnia eine neue HTTP-Kollektion angelegt. Wenn Sie die Request noch genauer sehen möchten, können Sie sich gerne bei uns melden. Wir haben leider nicht herausgefunden, wie wir unsere Collection exportieren können.

