Webshop Dokumentation

A Spring Boot & React App

BBW

Modul 150

Inhalt

[Einleitung 2](#_Toc4360320)

[Produkte 2](#_Toc4360321)

[Mockup 2](#_Toc4360322)

[Backend 3](#_Toc4360323)

[Spring Initializr 3](#_Toc4360324)

[Lombok 3](#_Toc4360325)

[Implementierung und verwendete Vorlage 3](#_Toc4360326)

[Warenkorb 3](#_Toc4360327)

[Projektstruktur Backend 3](#_Toc4360328)

[Frontend 4](#_Toc4360329)

[Setup 4](#_Toc4360330)

[Implementierung 4](#_Toc4360331)

[Projektstruktur Frontend 4](#_Toc4360332)

[Fazit 5](#_Toc4360333)

# Einleitung

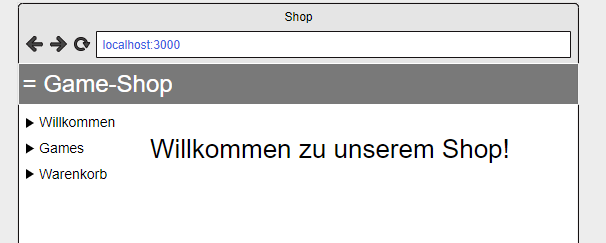
Dieses Dokument dokumentiert unseren Webshop, welchen wir im Rahmen des Moduls 150 bei Luigi Cavuoti erstellt haben. Die Aufgabenstellung lautete, einen simplen Webshop mithilfe von React und Spring Boot zu erstellen. Da wir beide noch fast keine Erfahrung bezüglich React und Spring Boot hatten, kam eine grosse Herausforderung auf uns zu, welche wir im Verlaufe der Arbeit mehr oder weniger gut bewältigen konnten.

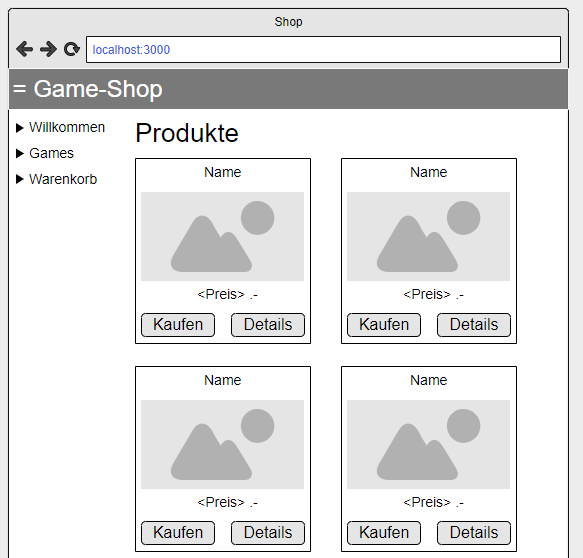
# Produkte

Nach gemeinsamem Abklären, habe wir uns darauf geeinigt, (Computer) Spiele im Webshop zu verkaufen. Die Produkte haben einen Namen, Preis und ein Bild.

# Mockup

Um den Webshop gut planen zu können, wurden Mockups benötigt.  
Da ein gratis Tool dafür benutzt wurde, konnte es nicht sehr realitätsnah dargestellt werden.  
Die grössten Unterschiede bestehen darin, dass es Keine Icons gibt und das Menü immer ausgeklappt ist, wobei das in der eigentlichen Webapp ausklappbar sein soll.

Homepage:  


Produkte-Darstellung:  


# Backend

Das Backend wurde, wie definiert, mit Spring Boot erstellt und realisiert.

## Spring Initializr

Spring Initializr ermöglicht es uns, schnell und einfach eine Basis für unser Projekt zu erstellen, indem wir einfach die gewünschten Dependencies auswählen und der Initializr uns dann ein leeres Projekt mit einer MainApplication und einem pom.xml-File erstellt. Das pom.xml-File enthält logischerweise die im Initializr ausgewählten Dependencies.  
Für unseren Webshop haben wir folgende relevanten Dependencies ausgewählt:

* Lombok
* Web
* H2

(Wir haben noch mehr, diese sind aber für die Dokumentation nicht relevant)

Wir werden im Folgenden nur auf Lombok noch speziell eingehen, da dies eine spezielle Dependency ist und nicht unbedingt notwendig, im Gegensatz zu H2 und Web.

### Lombok

Lombok erspart es uns, Getter und Setter sowie Konstruktoren schreiben zu müssen, da man die Klassen einfach mit den gewünschten Annotationen annotieren kann.   
@Getter generiert beim Kompilieren des Codes die Getter der Klasse, welche man annotiert hat. @Setter die Settermethoden und für die Konstruktoren verwendet man @AllArgsConstructor und @NoArgsConstructor. Ausserdem gibt es noch die Annotation @EqualsAndHashCode. Die Namen sind selbsterklärend. Selbstverständlich gibt es mit Lombok noch viele weitere Möglichkeiten, dies sind aber diese, welche wir verwendet haben.

## Implementierung und verwendete Vorlage

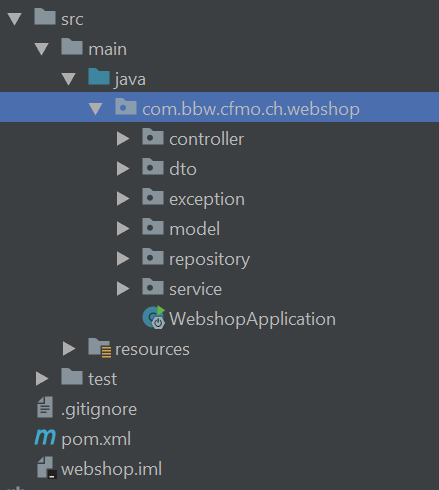
Bei der Implementierung des Backends mit Spring Boot, haben wir uns, bis auf die Umsetzung des Warenkorbs, stark an die Vorlage [dieser Seite](https://www.baeldung.com/spring-angular-ecommerce) gehalten. Diese Anleitung beschreibt Schritt für Schritt, wie man ein simples Backend für einen Webshop erstellen kann. Natürlich mit Spring Boot. Offensichtlich hätten wir auch alles selbst neu schreiben können, wir haben uns aber entschieden mit dieser Seite/Anleitung zu arbeiten, da man beim Programmieren das Rad nicht immer neu erfinden sollte und das verwenden, was es schon gibt.

## Warenkorb

Den Warenkorb in unserer Applikation haben wir selbst implementiert, ohne Vorlage oder Hilfe aus dem Internet. Unsere Lösung für den Warenkorb im Backend ist relativ simpel gehalten, dennoch erfüllt er seinen Zweck. Wir haben uns deshalb entschieden, den Warenkorb in der Datenbank zu speichern, da man ihn so über mehrere Sessions behalten könnte, sofern man Benutzerkonten etc. auch noch integrieren würde, was aber nicht Teil dieser Aufgabe war. Alternativ hätte man auch Cookies verwenden können, dies hätte aber unseren Wissensrahmen gesprengt, weswegen für uns die Datenbank eine gute Lösung darstellte.

## Projektstruktur Backend

Die Backendapplikation ist in drei Schichten aufgeteilt. Wir haben die Controller, welche die Rest-Schnittstellen verwaltet, die Services, welche die Zwischenschicht zwischen Controller und Repositories bildet und die Repositories, welche die Schnittstelle zur Datenbank verwaltet. Der Vorteil an dieser Architektur ist, dass man unabhängig von der einen Schnittstelle, die andere Austauschen kann, ohne die anderen Klassen anpassen zu müssen.



# Frontend

Das Frontend wurde nach den Vorgaben mit React umgesetzt.

## Setup

Das Grundlegendste der React App konnte mit dem Befehl «npx create-react-app <name>» initialisiert werden. Um alle wichtigen Module hinzuzufügen, braucht es noch eine Liste an weiteren Installierungen, die über die Konsole ausgeführt werden können.

npm install @material-ui/core  
npm install --save react-router-dom  
npm install --save react-router  
npm install @material-ui/icons

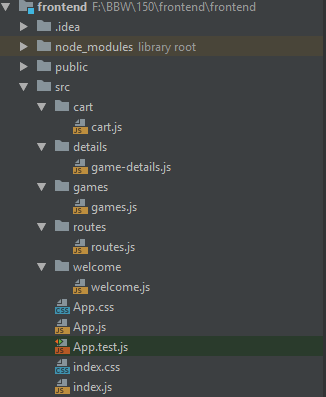
Mit npm start kann die Applikation dann gestartet werden.

## Implementierung

Im Frontend gibt es vier Komponenten: Die Home-Page, Produkte-Übersicht, Produkt-Details und der Warenkorb. Diese wurden nach der Planung umgesetzt.  
Wichtig hier zu vermerken ist, dass die Produkt-Details Seite als einzige nicht in der Menü-Bar enthalten ist, da man ein Produkt auswählen muss, um dieses genauer anzuschauen.

## Projektstruktur Frontend

Die einzelnen Seiten werden in «/src/<page>» definiert.  
Des Weiteren gibt es noch den Router, welcher die Menü-Bar Komponente ist.  
Ausserhalb des src Ordners, wird mit «App» generelle Eigenschaften vom Webshop definiert und im «index» werden die Ressourcen für die einzelnen Seiten definiert.



# Fazit

Alles in allem lief die Arbeit mehr oder weniger gut. Zu Anfangs hatten wir Probleme mit unserem VCS, was sich aber erledigte als wir auf Github umstiegen. Ausserdem stellte die ReactApp eine Herausforderung dar, da wir beide keinerlei Erfahrung mit React hatten. Das Backend stellte eine weniger grosse Schwierigkeit dar, da wir einerseits schon ein wenig Erfahrung hatten und andererseits, da das Internet eine sehr grosse Hilfe darstellte mit guten und detaillierten Erklärungen.