
Webshopping mit Spring Boot und React

eine PraxisProjekt von Ayman ABOUHALI

8. April 2024

Diese Arbeit ist von mir selbständig angefertigt und verfasst. Es sind keine anderen
als die angegebenen Quellen und Hilfsmittel benutzt worden.

ABOUHALI Ayman
.....

Unterschrift

Diese Arbeit wurde betreut von:

Betreuer: Prof Dr.rer.nat Herr Rohde

Abstrakt

Im heutigen digitalen Zeitalter ist es essenziell, Webanwendungen zu entwickeln, die sowohl schnell als auch benutzerfreundlich sind. Benutzer erwarten nahtlose Erfahrungen auf verschiedenen Geräten, was dazu führt, dass moderne Web-Entwicklung vermehrt auf einfache Seiten und Applikationen setzt. Die Hauptziele sind Effizienzsteigerung, schnelle Antworten und einfache Handhabung. Single-Page-Applikationen (SPA) vereinen diese Ansprüche, indem sie nur aus einem HTML-Dokument bestehen und Inhalte dynamisch nachladen. Dies ermöglicht eine schnellere Benutzererfahrung, da keine ständigen Seitenwechsel nötig sind.

React, ein häufig genutztes JavaScript-Framework, bietet die Basis für SPA-Entwicklung. Es optimiert Gestaltung und Aussehen für verschiedene Geräte und ermöglicht einen effizienten Offline-Modus. React zeichnet sich durch einen einseitigen Datenfluss aus und bietet starke Performance, auch bei komplexen Anwendungen. Der unidirektionale Datenfluss vereinfacht den Aufbau und die Wechselwirkungen der Komponenten.

Trotz der Vorteile von React kann die Zustandsverwaltung zu Komplexität und undurchsichtigem Quellcode führen. Redux bietet hier eine Lösung durch zentrale Zustandsverwaltung. Insgesamt bietet React eine ideale Plattform für die Entwicklung komplexer Webanwendungen mit einheitlicher Benutzerführung.

Auf der Backend-Seite bietet Spring Boot eine leistungsfähige Plattform zur Entwicklung von robusten und skalierbaren Anwendungen. Mit Spring Boot können Backend-Services und APIs schnell und effizient entwickelt werden. Spring Boot vereinfacht die Konfiguration und den Betrieb von Spring-Anwendungen erheblich und ermöglicht eine schnelle Implementierung von RESTful APIs.

Durch die Kombination von React für die Frontend-Entwicklung und Spring Boot für das Backend bietet sich eine optimale Lösung zur Entwicklung von Webanwendungen an. React bietet eine ideale Plattform für die Entwicklung von benutzerfreundlichen Benutzeroberflächen, während Spring Boot eine leistungsstarke und skalierbare Backend-Plattform bereitstellt. Diese Kombination ermöglicht die Entwicklung von modernen und ansprechenden Webanwendungen, die den Anforderungen der heutigen digitalen Landschaft gerecht werden.

Inhaltsverzeichnis

1	Projektbeschreibung	1
1.1	Funktionale Spezifikation	1
1.2	Non-Functional Specification	2
2	die Konzeption von Klassendiagrammen und API	3
2.1	Klassendiagramm der Anwendung	3
2.2	API-Dokumentation mit Swagger	4
3	Verwendete Technologien	7
3.1	Backend-Entwicklung	7
3.1.1	Java (Version 17.0.6)	7
3.1.2	Spring Framework	7
3.1.2.1	Spring Boot (Version 3.2.2)	7
3.1.2.2	Spring Data (Version 3.2.2.)	7
3.1.2.3	Spring Web und Spring REST (Version 3.2.2)	7
3.1.2.4	Spring Security (Version 6.2.1)	7
3.1.2.5	Spring Validator (Version 3.0.2)	8
3.1.2.6	Spring Devtools (Version 2.5.4)	8
3.1.2.7	Lombok (Version 1.18.20)	8
3.1.2.8	Stripe (Version 20.62.0)	8
3.2	Front-End Entwicklung	8
3.2.1	React.js (Version 18.2.0)	8
3.2.2	Redux (5.0.1)	8
3.2.3	MUI : Material-UI (Version 5.15.6)	8
3.2.4	Tailwind CSS (Version 3.4.1)	9
3.2.5	JavaScript / HTML / CSS / SCSS	9
3.3	Datenbank-Management	9
3.3.1	MySQL	9
3.4	Weiter Technologien	9
3.4.1	Git	9
3.4.2	IDEs (Integrierte Entwicklungsumgebungen)	9

Abbildungsverzeichnis

2.1	Klassendiagramm der Anwendung	3
2.2	Ausschnitt der API-Dokumentation 1	4
2.3	Ausschnitt der API-Dokumentation 2	5

1 Projektbeschreibung

Das E-Commerce-Projekt zielt darauf ab, das Online-Shopping-Erlebnis zu revolutionieren, indem verschiedene Funktionalitäten nahtlos integriert werden, um die Benutzerfreundlichkeit und Zufriedenheit zu verbessern. Diese ambitionierte Initiative stellt ein umfassendes System vor, das nicht nur eine effiziente Produktsuche und -kauf ermöglicht, sondern auch eine reibungslose Auftragsabwicklung und Kundenbindung sicherstellt.

Im Kern konzentriert sich das E-Commerce-Projekt darauf, eine robuste Online-Plattform aufzubauen, die die Benutzerfreundlichkeit maximiert und sichere Transaktionen fördert. Durch die Implementierung fortschrittlicher Technologien und benutzerzentrierter Designprinzipien zielt das Projekt darauf ab, ein nahtloses Einkaufserlebnis über verschiedene Geräte und Plattformen hinweg zu bieten.

Die Hauptziele des E-Commerce-Projekts umfassen:

- Aufbau einer benutzerfreundlichen und responsiven Website, die den Anforderungen moderner Online-Shopper gerecht wird.
- Implementierung eines sicheren Authentifizierungs- und Autorisierungssystems zum Schutz von Benutzerdaten und Transaktionen.

1.1 Funktionale Spezifikation

- **Administrator API** Diese API umfasst Funktionen, die es Administratoren ermöglichen, den Zustand jeder Station zu überwachen und Daten abzurufen. Zusätzlich bietet sie Mechanismen zur Verwaltung von Produkten sowie zur Einsichtnahme in Kunden- und Bestelldaten im Dashboard. Auch die Funktionen für die Anmeldung/Registrierung und die Passwortwiederherstellung sind enthalten, um die sichere Verwaltung der Software zu gewährleisten.

- **Benutzer-API**

Das Projekt bietet eine Benutzer-API, die es Entwicklern ermöglicht, Registrierungs- und Anmeldefunktionalitäten für Kunden zu implementieren sowie Tracking-Anwendungen. Außerdem können Benutzer Reservierungen auf der E-Commerce-Plattform vornehmen.

- **Relationale Datenbank mit MySQL**

Das Projekt verwendet MySQL als das zugrunde liegende Datenbanksystem, um alle Arten von API-Daten zu speichern, einschließlich Benutzerkonten, Produktinformationen, Bestellungen und Transaktionsdaten.

- **Benutzer-/Administrator-API**

Eine API, die speziell entwickelt wurde, um Benutzerkonten innerhalb der Anwendungen zu verwalten, einschließlich Benutzerregistrierung, Anmeldung und Profilverwaltung.

- **Administrator-Interface zur Überwachung des Systems**

Eine benutzerdefinierte und benutzerfreundliche Oberfläche, die den Überwachungsanforderungen des E-Commerce-Systems entspricht. Sie bietet ein ansprechendes Design für Echtzeitüberwachung und Befehlssendung.

1.2 Non-Functional Specification

- **Leistung:**

Die Anwendung sollte in der Lage sein, einen hohen Lastanstieg zu bewältigen, insbesondere während des Spitzenverkehrszeiten wie Black Friday oder anderen Verkaufsaktionen.

- **Benutzerfreundlichkeit und Design:**

+Das Design der Webseite sollte benutzerfreundlich sein und den Benutzern eine intuitive Navigation ermöglichen, um die Auffindbarkeit von Produkten und die Abschlussrate von Bestellungen zu erhöhen.

+Die Benutzeroberfläche sollte reaktionsschnell sein und sich an verschiedene Bildschirmgrößen und Geräte anpassen, um eine konsistente Benutzererfahrung auf Desktops, Tablets und Smartphones zu gewährleisten.

- **Sicherheit :**

+Die Anwendung sollte vor häufigen Sicherheitsbedrohungen wie Cross-Site-Scripting (XSS), SQL-Injektion und Cross-Site-Request-Forgery (CSRF) geschützt sein.

+Die Benutzerdaten sollten verschlüsselt und sicher gespeichert werden, um Datenschutzbestimmungen einzuhalten und Datenlecks zu vermeiden.

2 die Konzeption von Klassendiagrammen und API

2.1 Klassendiagramm der Anwendung

Im folgenden Abschnitt wird das Klassendiagramm der Anwendung präsentiert. Das Diagramm bietet eine visuelle Darstellung der Klassenstruktur und ihrer Beziehungen zueinander. Durch die Visualisierung der Architektur erhalten Leser einen umfassenden Überblick über die Softwarestruktur, was das Verständnis und die Analyse der Anwendung erleichtert.

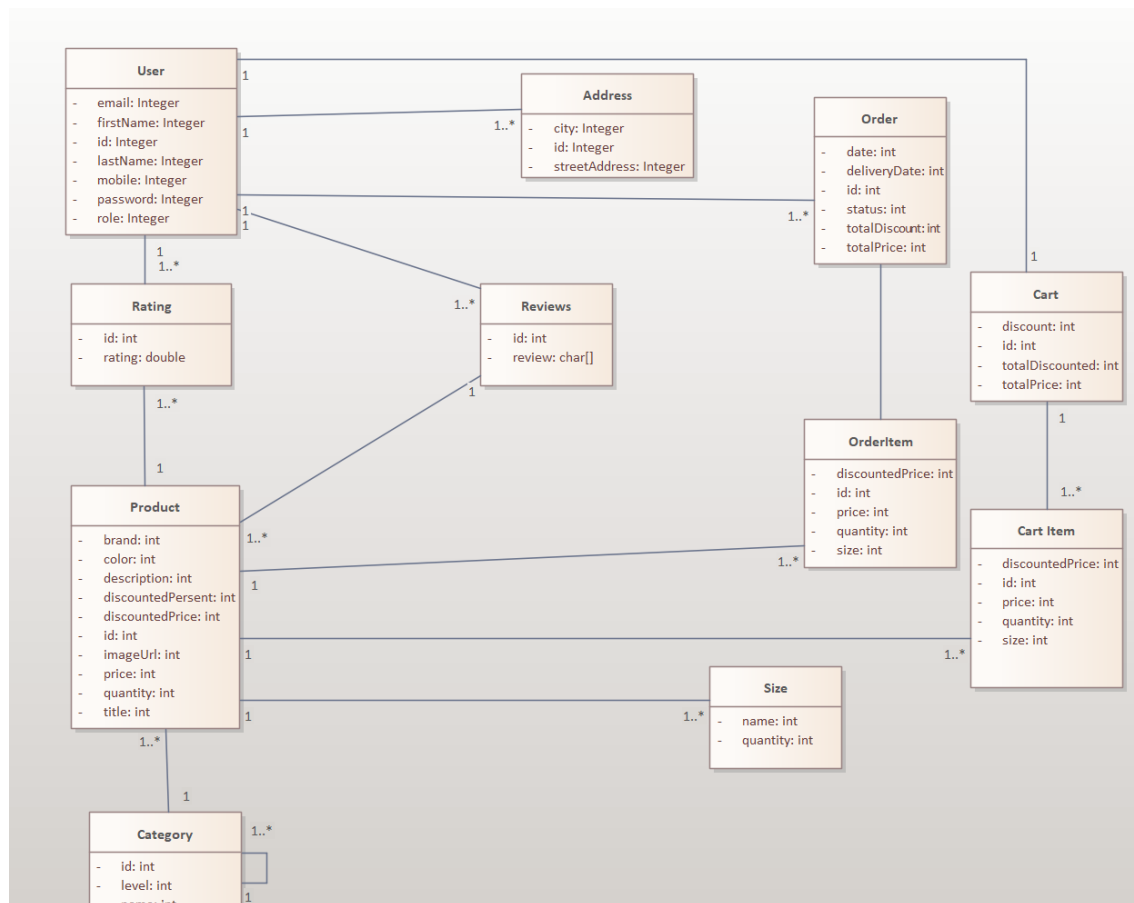


Abbildung 2.1: Klassendiagramm der Anwendung

2.2 API-Dokumentation mit Swagger

In diesem Abschnitt wird ein Ausschnitt der API-Dokumentation mithilfe von Swagger präsentiert. Das Swagger-UI bietet eine interaktive Darstellung der verfügbaren API-Endpunkte, ihrer Parameter und Rückgabewerte. Durch die Nutzung von Swagger können Entwickler die API-Funktionalität erkunden, testen und verstehen, ohne die Anwendung zu verlassen. Der folgende Screenshot vermittelt einen Eindruck davon, wie die API mithilfe von Swagger dokumentiert ist und wie Benutzer mit den verschiedenen Endpunkten interagieren können.

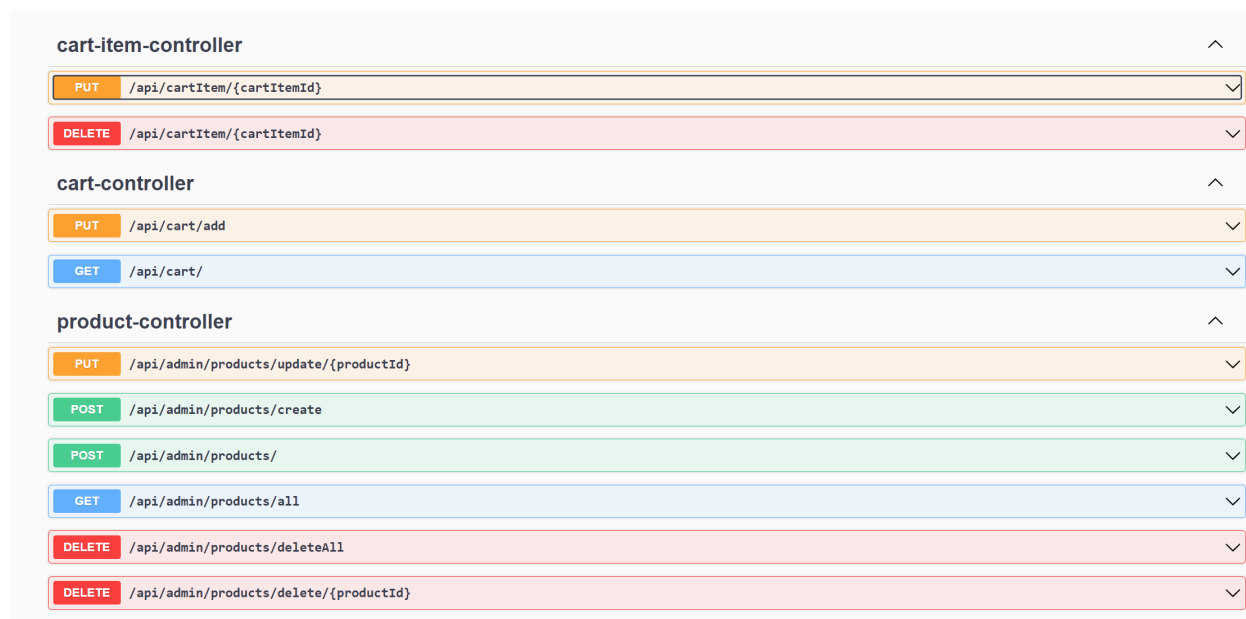


Abbildung 2.2: Ausschnitt der API-Dokumentation 1

order-adm-controller ^	
PUT	/api/admin/order/{orderId}/ship
PUT	/api/admin/order/{orderId}/delivered
PUT	/api/admin/order/{orderId}/confirmed
PUT	/api/admin/order/{orderId}/cancel
GET	/api/admin/order/
DELETE	/api/admin/order/{orderId}/deleted
auth-controller ^	
POST	/auth/signup
POST	/auth/signin
review-controller ^	
POST	/api/reviews/create
GET	/api/reviews/product/{productId}

Abbildung 2.3: Ausschnitt der API-Dokumentation 2

- **Authentifizierung:** Benutzer können sich mit JWT authentifizieren und Zugriff auf geschützte Ressourcen erhalten.
- **Produktsuche:** Kunden können Produkte nach bestimmten Kriterien suchen.
- **Warenkorb:** Kunden können Produkte hinzufügen und aus dem Warenkorb entfernen.
- **Bestellungen:** Kunden können Produkte aus dem Warenkorb bestellen.
- **Benutzerprofil:** Kunden können ihre Bestellungen einsehen. Darüber hinaus können sie Produktbewertungen und -rezensionen verfassen und hinzufügen.
- **Admin-Funktionen:** Der Administrator kann Produkte hinzufügen oder bearbeiten sowie Benutzerdaten ändern und Bestellungen einsehen.

3 Verwendete Technologien

Das Projekt nutzt eine Vielzahl von Technologien, die jeweils einem bestimmten Zweck dienen. Hier ist ein Überblick über die verwendeten Technologien:

3.1 Backend-Entwicklung

3.1.1 Java (Version 17.0.6)

Verwendet als Backend-Programmiersprache für die Entwicklung der Anwendung. Java bietet eine robuste und zuverlässige Grundlage für die Implementierung der Geschäftslogik und Datenverarbeitung.

3.1.2 Spring Framework

Eingebunden für den Aufbau der Backend-Infrastruktur, einschließlich RESTful-Webdienste, Dependency Injection und MVC-Architektur.

3.1.2.1 Spring Boot (Version 3.2.2)

Spring Boot bietet ein robustes und schlankes Framework für die Entwicklung von Java-basierten Anwendungen. Es vereinfacht die Konfiguration und den Aufbau von Anwendungen durch automatische Konventionen und Integrationen.

3.1.2.2 Spring Data (Version 3.2.2.)

Spring Data erleichtert Datenbankoperationen und die objektrelationale Abbildung (ORM) durch automatisierte Repository-Erstellung und Datenzugriffsmethoden.

3.1.2.3 Spring Web und Spring REST (Version 3.2.2)

Spring Web und Spring REST werden verwendet, um RESTful APIs zu erstellen und Webanfragen zu verarbeiten. Sie bieten Unterstützung für die Erstellung von RESTful Endpoints und die Verarbeitung von HTTP-Anfragen und -Antworten.

3.1.2.4 Spring Security (Version 6.2.1)

Spring Security gewährleistet die Authentifizierung und Autorisierung von Benutzern für einen sicheren Zugriff auf Ressourcen. Es bietet Funktionen wie Authentifizierung mit Benutzernamen/Passwort, OAuth, JWT usw.

3.1.2.5 Spring Validator (Version 3.0.2)

Spring Validator bietet Unterstützung für die Validierung von Dateninputs und die Sicherstellung der Datenintegrität. Es ermöglicht die Definition von Validierungsregeln für Formulardaten und Eingabefelder.

3.1.2.6 Spring Devtools (Version 2.5.4)

Spring Devtools bietet eine verbesserte Entwicklungserfahrung mit Funktionen wie automatischem Neustart der Anwendung, LiveReload und Hot Swapping. Es beschleunigt den Entwicklungsprozess und erhöht die Produktivität.

3.1.2.7 Lombok (Version 1.18.20)

Lombok vereinfacht die Java-Entwicklung, indem es automatisch Boilerplate-Code generiert. Es reduziert den Aufwand beim Schreiben von Getter, Setter, Konstruktoren und anderen häufig verwendeten Methoden.

3.1.2.8 Stripe (Version 20.62.0)

Stripe ist eine leistungsstarke Zahlungsabwicklungsplattform, die die Integration von Zahlungsfunktionen in Anwendungen erleichtert. Mit Stripe können Entwickler Zahlungen akzeptieren, verwalten und abwickeln, ohne komplexe Infrastruktur aufbauen zu müssen. Durch die Nutzung von Stripe können Entwickler Zeit sparen und sich auf die Kernfunktionen ihrer Anwendungen konzentrieren, während Stripe die Zahlungsabwicklung effizient und sicher abwickelt.

3.2 Front-End Entwicklung

3.2.1 React.js (Version 18.2.0)

Verwendet für die Entwicklung der Benutzeroberfläche der E-Commerce-Website. React bietet eine komponentenbasierte Architektur für die Erstellung interaktiver und reaktiver Benutzeroberflächen.

3.2.2 Redux (5.0.1)

Eingebunden zur Verwaltung des globalen Zustands der Anwendung. Redux ermöglicht eine zentrale Speicherung von Daten und erleichtert die Handhabung komplexer Anwendungsdaten.

3.2.3 MUI : Material-UI (Version 5.15.6)

Verwendet für die Implementierung von vorgefertigten React-Komponenten im Material Design-Stil. Material-UI verbessert die Benutzererfahrung durch konsistente und ansprechende Benutzeroberflächenkomponenten

3.2.4 Tailwind CSS (Version 3.4.1)

Eingesetzt für das Styling der Benutzeroberfläche. Tailwind CSS bietet nützliche CSS-Klassen, die eine schnelle Entwicklung und Anpassung von UI-Elementen ermöglichen.

3.2.5 JavaScript / HTML / CSS / SCSS

Wesentliche Web-Technologien, die für Client-seitiges Scripting, Strukturierung von Webinhalten und Styling.

3.3 Datenbank-Management

3.3.1 MySQL

Verwendet als relationales Datenbankmanagementsystem (RDBMS) für die Speicherung von Produktinformationen, Benutzerdaten, Bestelldetails und anderen relevanten Daten

3.4 Weiter Technologien

3.4.1 Git

Eingesetzt für die Versionskontrolle und Zusammenarbeit im Entwicklungsprozess.

3.4.2 IDEs (Integrierte Entwicklungsumgebungen)

In der Entwicklung unserer Anwendung habe ich Visual Studio für das Frontend und IntelliJ IDEA für das Backend mit Spring Boot eingesetzt. Diese bewährten integrierten Entwicklungsumgebungen boten uns eine benutzerfreundliche Plattform für das Codieren, Debuggen und Testen, was zu effizienter Entwicklung und hoher Anwendungsqualität führte.

Fazit

Zusammenfassend lässt sich sagen, dass das Ecommerce-Projekt sein Ziel, eine nahtlose und effiziente Plattform für den Online-Einkauf zu schaffen, erreicht hat. Durch die Integration fortschrittlicher Technologien und benutzerorientierter Designprinzipien bietet das Projekt eine umfassende Lösung für Kunden und Administratoren gleichermaßen.

Für Kunden bietet die E-Commerce-Plattform eine benutzerfreundliche Oberfläche, die ein einfaches Browsen, eine einfache Produktauswahl und eine einfache Kaufabwicklung ermöglicht. Funktionen wie Registrierung, Anmeldung und Auftragsverfolgung verbessern das gesamte Einkaufserlebnis, während sichere Zahlungsgateways die Sicherheit und Zuverlässigkeit der Transaktionen gewährleisten.

Administratoren profitieren von robusten Backend-Funktionen, einschließlich Bestandsverwaltung, Auftragsabwicklung und Kundensupport-Tools. Die Skalierbarkeit und Flexibilität des Systems ermöglicht eine einfache Anpassung an veränderte Geschäftsanforderungen.

Durch Echtzeit-Analysen und Datenvisualisierung ermöglicht das Ecommerce-Projekt Unternehmen, fundierte Entscheidungen in Bezug auf Bestandsmanagement, Marketingstrategien und Kundenbindung zu treffen. Durch den Einsatz von Spitzentechnologien wie React, Redux und Spring Boot bietet die Plattform eine hohe Leistung und Reaktionsfähigkeit auf verschiedenen Geräten und Betriebssystemen.

Im Wesentlichen stellt das Ecommerce-Projekt einen bedeutenden Fortschritt im Bereich des Online-Handels dar und bietet eine skalierbare, sichere und benutzerfreundliche Lösung für Unternehmen, die im digitalen Zeitalter erfolgreich sein wollen. Da sich die Technologie ständig weiterentwickelt, ist das Projekt bereit für weitere Verbesserungen und Innovationen, die die Zukunft des E-Commerce in Richtung größerer Effizienz, Nachhaltigkeit und Kundenzufriedenheit vorantreiben.

