

Meine Buchverwaltungsanwendung

Modul 294 - Frontend einer interaktiven Webapplikation realisieren

Modul 295 - Backend für Applikationen realisieren



Prasanna Thanikasalam

UIFZ-2424-018

Dozent: Patrick Meier

1	Inhalt	
2	Projektidee	3
3	Anforderungskatalog.....	4
4	Klassendiagramm	5
5	Storyboard	6
5.1	Fall 1: Neues Buch erfassen	6
5.2	Fall 2: Buch bearbeiten.....	8
5.3	Fall 3: Statistik ansehen.....	11
6	Screen-Mockups.....	13
7	REST-Schnittstellen.....	16
7.1	Übersicht.....	16
7.2	REST-Schnittstellen - Bücher	16
7.2.1	Alle Bücher abrufen	16
7.2.2	Neues Buch anlegen	17
7.2.3	Einzelnes Buch abrufen.....	19
7.2.4	Buch aktualisieren	20
7.2.5	Buch löschen	22
7.3	REST-Schnittstellen - Autoren.....	23
7.3.1	Alle Autoren abrufen	23
7.3.2	Neuen Autor anlegen.....	24
7.3.3	Einzelnen Autor abrufen	25
7.3.4	Einzelnen Autor abrufen	26
8	Testprotokoll.....	27
8.1	JUnit-Tests - Backend.....	27
8.1.1	BookValidationTest	27
8.1.2	BookTest	28
8.1.3	AuthorTest.....	29
8.2	JUnit-Tests - Frontend	31
8.2.1	AuthorFormTest	31
8.2.2	AuthorListTest	32
8.2.3	BookEdit.test.....	34
8.2.4	BookFilter.test	35
9	Installationsanleitung	36
9.1	Allgemein	36
9.2	Backend starten	36
9.3	Frontend starten.....	36
10	Hilfestellungen.....	37
11	Bildquelle.....	38
12	Troubleshooting	39

2 Projektidee

Stellen Sie sich vor, Sie möchten Ihre persönliche Büchersammlung digital organisieren. Schnell, flexibel und übersichtlich! Mit diesem Buchverwaltungs-Tool können Nutzer ganz einfach neue Bücher erfassen, indem sie Titel, Autor, Genre und Einband eintragen und jedem Buch einen Status wie „gelesen“, „aktuell am Lesen“ oder „Wunschliste“ zuweisen.

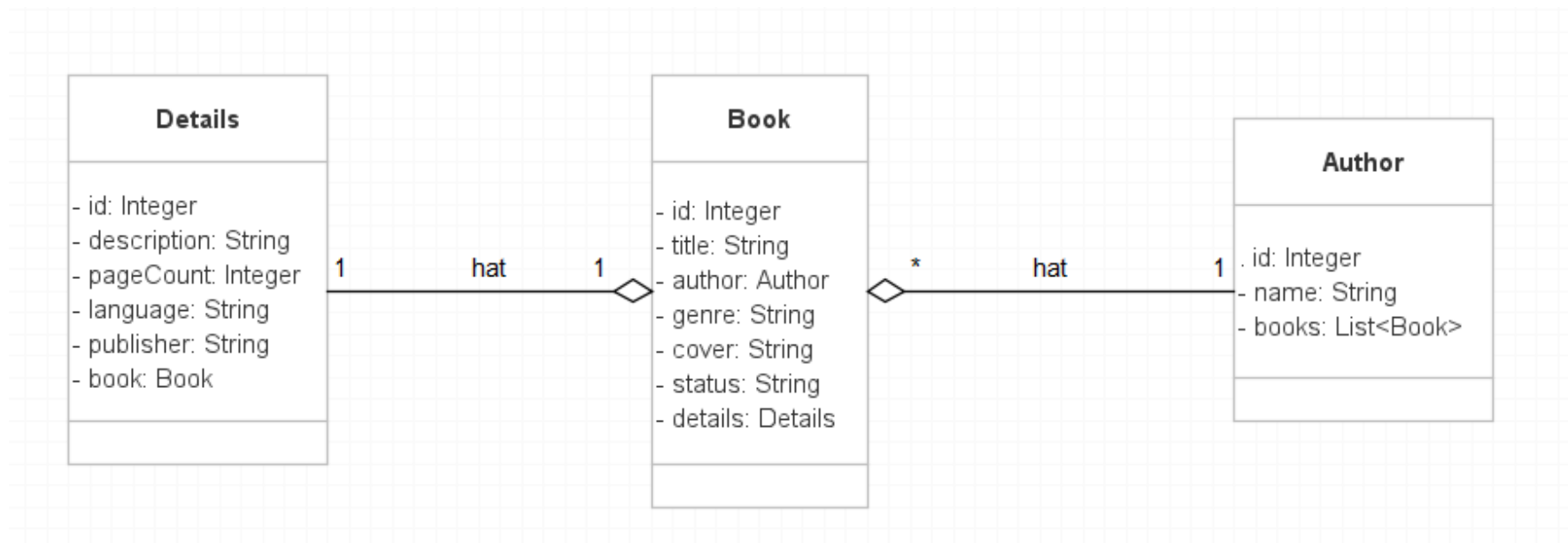
Die Bücherliste ist auf einen Blick filterbar. Sie können nach Status, Genre oder Einband (Hardcover, Softcover, eBook) filtern, um gezielt bestimmte Bücher zu finden oder Ihre Sammlung zu sortieren. Das Hinzufügen oder Bearbeiten von Büchern ist jederzeit möglich. So bleibt Ihre Liste immer aktuell.

Mit der integrierten Statistikfunktion sehen Sie auf einen Blick, wie viele Bücher Sie bereits gelesen haben, welche Sie gerade lesen und welche auf Ihrer Wunschliste stehen. Damit behalten Sie Ihren Lesefortschritt im Überblick und finden immer das passende Buch für jede Lese Laune.

3 Anforderungskatalog

Buchverwaltung		
	A1	Anlegen von Büchern
	A2	Bearbeiten von Büchern
	A3	Löschen von Büchern
	A4	Anzeigen der Bücher
Filterfunktionen		
	B1	Filtern nach Status
	B2	Filtern nach Genre
	B3	Filtern nach Einband
Statusverwaltung		
	C1	Statuszuweisung
	C2	Statusänderung
Statistikfunktion		
	D1	Statistische Auswertung
Benutzerfreundlichkeit	E1	Navigation
	E2	Intuitive Bedienung

4 Klassendiagramm



5 Storyboard

5.1 Fall 1: Neues Buch erfassen

Der Nutzer klickt auf „Buch hinzufügen“

alle	Status	Genre	Einband
aktuell am Lesen (3)			
Harry Potter 2	J.K. Rowling	Fantasy	Hardcover
Harry Potter 3	J.K. Rowling	Fantasy	Hardcover
Arsene Lupin	Maurice Leblanc	Krimi	Taschenbuch
gelesen (2)			
Die Verwandlung	Franz Kafka	Erzählung	Taschenbuch
Terror	Ferdinand von Schirach	Theaterstück	Taschenbuch
Wunschliste (2)			
The Great Gatsby	F. Scott Fitzgerald	Roman	Taschenbuch
The box in the Striped ...	John Boyne	Roman	Taschenbuch
Neues Buch hinzufügen			
Statistik			

Nutzer erfasst neues Buch

Neues Buch hinzufügen	
Titel:	<input type="text"/>
Autor:	<input type="text"/>
Genre:	<input type="text"/>
Status:	<input type="text" value="wählen"/> ▼
Einband:	<input type="text" value="wählen"/> ▼
Hinzufügen	
Abbrechen	

Nutzer klickt auf «hinzufügen»

Neues Buch hinzufügen

Titel:	Harry Potter 4
Autor:	J. K. Rowling
Genre:	Fantasy
Status:	aktuell am lesen ▾
Einband:	Hardcover ▾

Hinzufügen

Abbrechen

Neues Buch wurde erfasst

Buchverwaltung

alle	Status	Genre	Einband
------	--------	-------	---------

aktuell am Lesen (4)

Harry Potter 2	J.K. Rowling	Fantasy	Hardcover
Harry Potter 3	J.K. Rowling	Fantasy	Hardcover
Arsene Lupin	Maurice Leblanc	Krimi	Taschenbuch
Harry Potter 4	J.K. Rowling	Fantasy	Hardcover

gelesen (2)

Die Verwandlung	Franz Kafka	Erzählung	Taschenbuch
Terror	Ferdinand von Schirach	Theaterstück	Taschenbuch

Wunschliste (2)

The Great Gatsby	F. Scott Fitzgerald	Roman	Taschenbuch
The box in the Striped ...	John Boyne	Roman	Taschenbuch

Neues Buch hinzufügen

Statistik

5.2 Fall 2: Buch bearbeiten

Nutzer klickt auf Buch

Buchverwaltung			
alle	Status	Genre	Einband
aktuell am Lesen (4)			
Harry Potter 2	J.K. Rowling	Fantasy	Hardcover
Harry Potter 3	J.K. Rowling	Fantasy	Hardcover
Arsene Lupin	Maurice Leblanc	Krimi	Taschenbuch
Harry Potter 4	J.K. Rowling	Fantasy	Hardcover
gelesen (2)			
Die Verwandlung	Franz Kafka	Erzählung	Taschenbuch
Terror	Ferdinand von Schirach	Theaterstück	Taschenbuch
Wunschliste (2)			
The Great Gatsby	F. Scott Fitzgerald	Roman	Taschenbuch
The box in the Striped ...	John Boyne	Roman	Taschenbuch
Neues Buch hinzufügen			
Statistik			

Nutzer bearbeitet «Titel» vom Buch

Buchdetail / bearbeiten	
Titel:	Harry Potter 4
Autor:	J. K. Rowling
Genre:	Fantasy
Status:	aktuell am lesen ▾
Einband:	Hardcover ▾
Änderungen Speichern	
Abbrechen	

Nutzer bearbeitet «Status» vom Buch

The screenshot shows a web form titled 'Buchdetail / bearbeiten'. It contains several input fields for book details: 'Titel' (Harry Potter 4), 'Autor' (J. K. Rowling), and 'Genre' (Fantasy). Below these, there is a 'Status' dropdown menu which is currently open, showing three options: 'aktuell am lesen', 'gelesen' (which is highlighted in green), and 'Wunschliste'. There is also an 'Einband' field. At the bottom of the form, there are two buttons: 'Änderungen Speichern' and 'Abbrechen'.

Nutzer speichert Änderungen

The screenshot shows the same 'Buchdetail / bearbeiten' form, but now the 'Status' dropdown menu is closed and 'gelesen' is selected. The 'Einband' dropdown menu is also open, showing 'Hardcover' as the selected option. The 'Änderungen Speichern' button at the bottom is highlighted in orange, indicating it is the next step in the process. The 'Abbrechen' button remains visible below it.

Nutzer sieht bearbeitetes Buch

Buchverwaltung

alle	Status	Genre	Einband
aktuell am Lesen (3)			
Harry Potter 2	J.K. Rowling	Fantasy	Hardcover
Harry Potter 3	J.K. Rowling	Fantasy	Hardcover
Arsene Lupin	Maurice Leblanc	Krimi	Taschenbuch
gelesen (3)			
Die Verwandlung	Franz Kafka	Erzählung	Taschenbuch
Terror	Ferdinand von Schirach	Theaterstück	Taschenbuch
Harry Potter und der Feu..	J.K. Rowling	Fantasy	Hardcover
Wunschliste (2)			
The Great Gatsby	F. Scott Fitzgerald	Roman	Taschenbuch
The box in the Striped ..	John Boyne	Roman	Taschenbuch

Neues Buch hinzufügen

Statistik

5.3 Fall 3: Statistik ansehen

Nutzer klickt auf «Statistik»

Buchverwaltung			
alle	Status	Genre	Einband
aktuell am Lesen			
Harry Potter 2	J.K. Rowling	Fantasy	Hardcover
Harry Potter 3	J.K. Rowling	Fantasy	Hardcover
Arsene Lupin	Maurice Leblanc	Krimi	Taschenbuch
gelesen			
Die Verwandlung	Franz Kafka	Erzählung	Taschenbuch
Terror	Ferdinand von Schirach	Theaterstück	Taschenbuch
Harry Potter und der Feu...	J.K. Rowling	Fantasy	Hardcover
Wunschliste			
The Great Gatsby	F. Scott Fitzgerald	Roman	Taschenbuch
The boy in the Striped ...	John Boyne	Roman	Taschenbuch
Neues Buch hinzufügen			
Statistik			

Nutzer sieht seine Fortschritte

Meine Lese-Statistik	
Gelesen:	3
aktuell am Lesen:	3
Wunschliste:	2
Zurück	

6 Screen-Mockups

①

Buchverwaltung

alle
status
Genre
Tags

Aktuell am Lesen (3)

Harry Potter 2	J.K. Rowling	Fantasy	Buchprojekt
Harry Potter 3	J.K. Rowling	Fantasy	aus Spass

gelesen (23)

Globi & die Banditen	Robert Lips	Kinder	Babysitting
----------------------	-------------	--------	-------------

Wunschliste (11)

Fantastic Beast	David Yates	Fantasy	interessant
-----------------	-------------	---------	-------------

Neues Buch hinzufügen

Statistik

Neues Buch hinzufügen

Titel:

Autor:

Genre:

Status: ☐ ← gelesen, aktuell am Lesen, Wunschliste

Tags:

oder
speichern

Buchdetail / bearbeiten

Titel: Harry Potter 2

Autor: J.K. Rowling

Genre: Fantasy

Status: aktuell am lesen

Tags: Buchvortrag halten

Änderung Speichern

Abbrechen

Meine Lese-Statistik

Gelesen:	25
aktuell am lesen:	3
Wunschliste:	11

Zurück

7 REST-Schnittstellen

7.1 Übersicht

Die API ermöglicht das Verwalten von Büchern, Autoren und Buchdetails. Alle Daten werden im JSON-Format gesendet und empfangen.

Basis-URL:

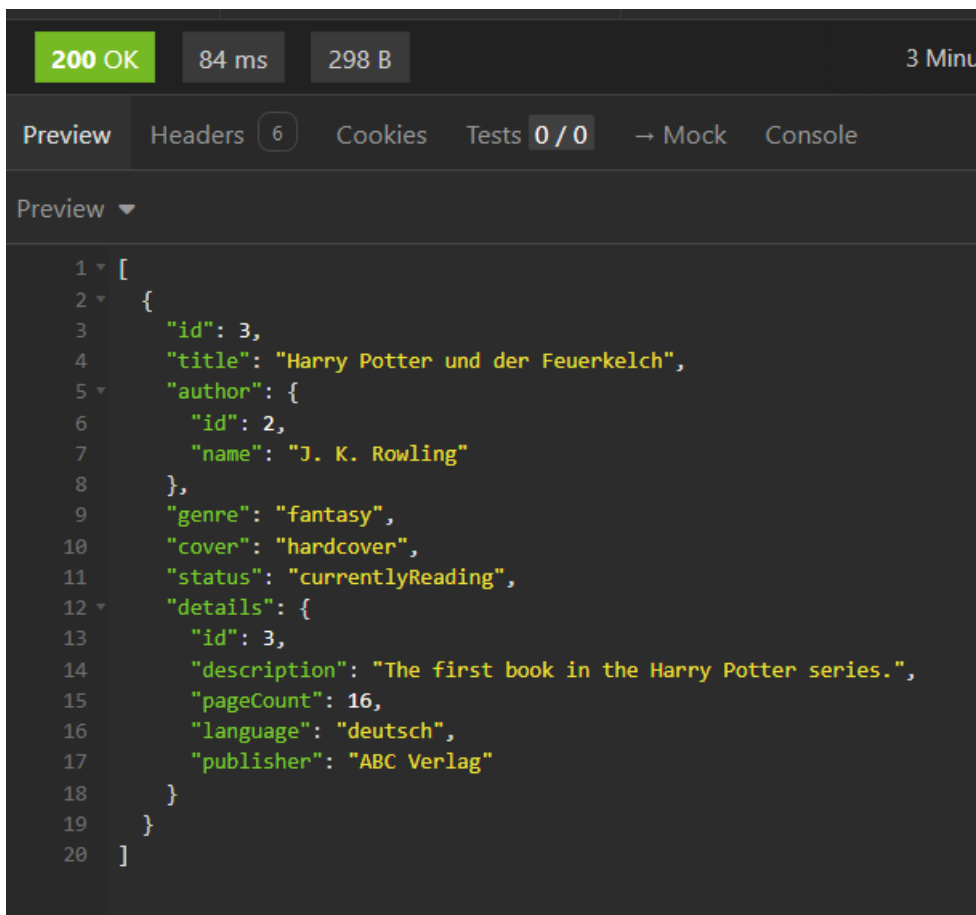
<http://localhost:8080/book>

7.2 REST-Schnittstellen - Bücher

7.2.1 Alle Bücher abrufen

URL	http://localhost:8080/book
Methode	GET
Beschreibung:	Gibt eine Liste aller Bücher zurück.

Ausgabe:

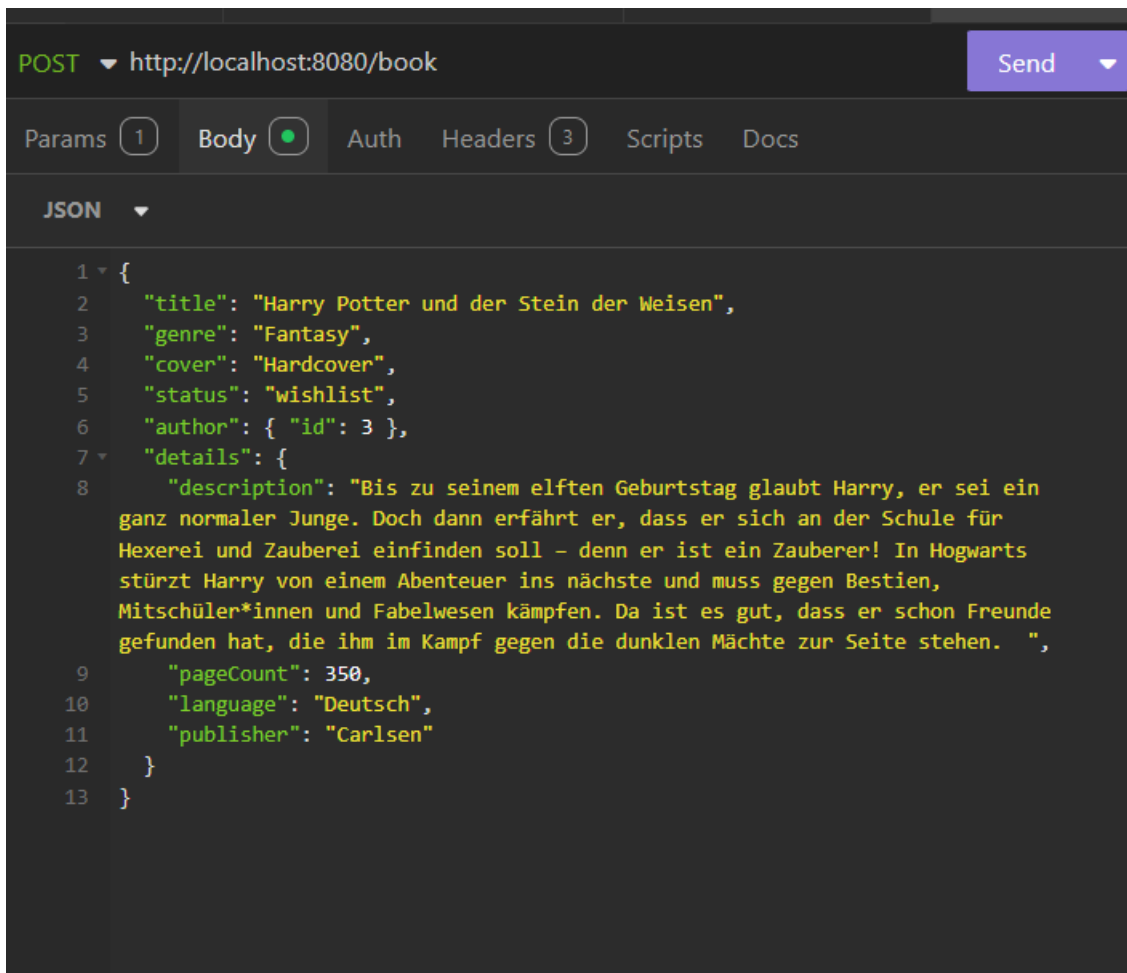


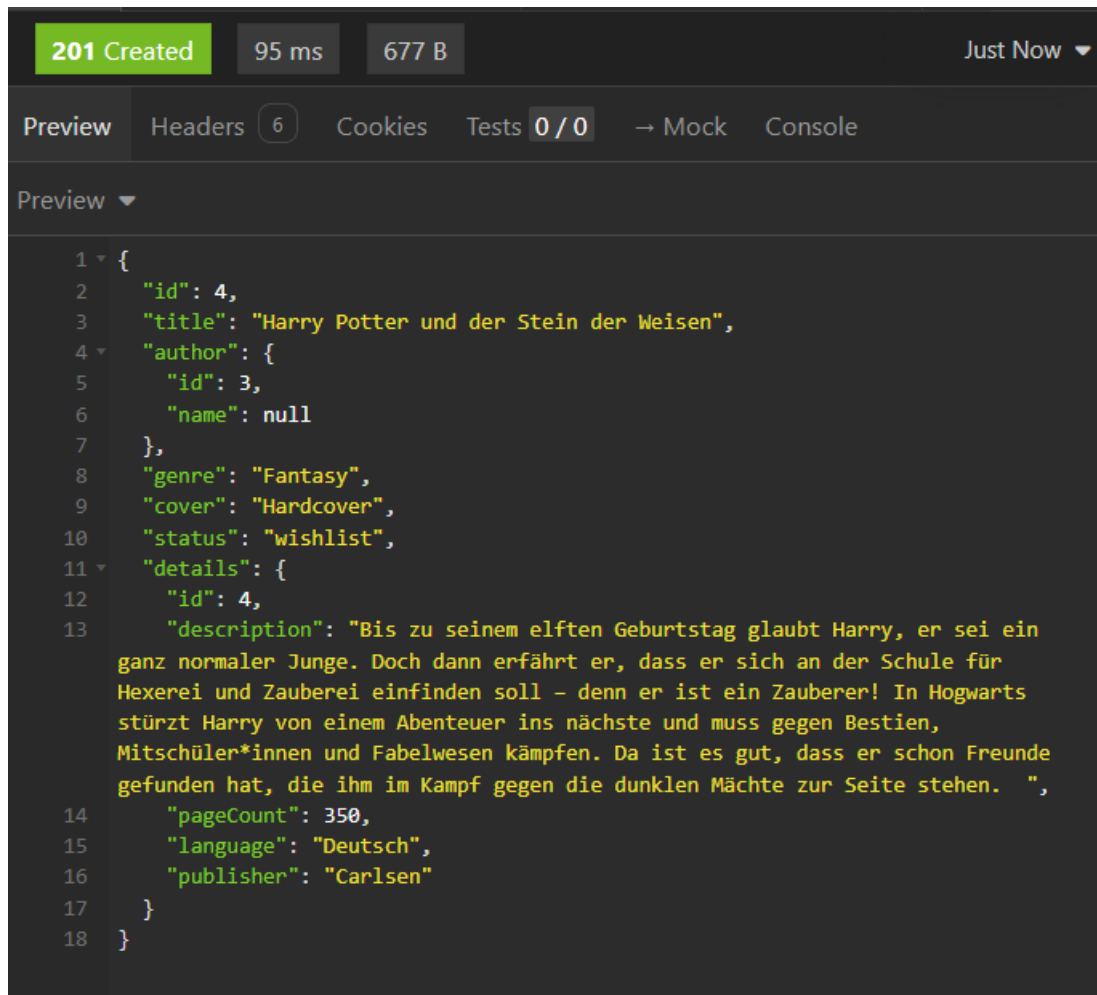
```
200 OK 84 ms 298 B 3 Minu
Preview Headers 6 Cookies Tests 0/0 → Mock Console
Preview ▾
1 [
2   {
3     "id": 3,
4     "title": "Harry Potter und der Feuerkelch",
5     "author": {
6       "id": 2,
7       "name": "J. K. Rowling"
8     },
9     "genre": "fantasy",
10    "cover": "hardcover",
11    "status": "currentlyReading",
12    "details": {
13      "id": 3,
14      "description": "The first book in the Harry Potter series.",
15      "pageCount": 16,
16      "language": "deutsch",
17      "publisher": "ABC Verlag"
18    }
19  }
20 ]
```


7.2.2 Neues Buch anlegen

URL	http://localhost:8080/book
Methode	POST
Beschreibung	Legt ein neues Buch an.

Request Body:



Ausgabe:

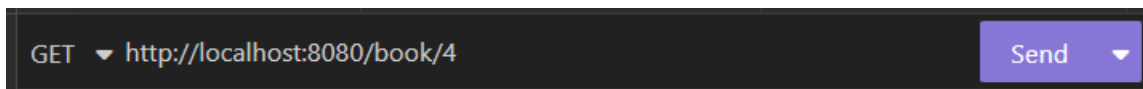
The screenshot shows a REST client interface with a top bar displaying '201 Created', '95 ms', and '677 B'. Below this is a navigation bar with 'Preview' (selected), 'Headers' (6), 'Cookies', 'Tests' (0/0), '→ Mock', and 'Console'. The main area is titled 'Preview' and contains a JSON response for a book. The JSON is formatted with line numbers 1 through 18 on the left. The response includes an 'id', 'title', 'author' (with an 'id' and 'name'), 'genre', 'cover', 'status', 'details' (with an 'id' and a long 'description'), 'pageCount', 'language', and 'publisher'.

```
1 {  
2   "id": 4,  
3   "title": "Harry Potter und der Stein der Weisen",  
4   "author": {  
5     "id": 3,  
6     "name": null  
7   },  
8   "genre": "Fantasy",  
9   "cover": "Hardcover",  
10  "status": "wishlist",  
11  "details": {  
12    "id": 4,  
13    "description": "Bis zu seinem elften Geburtstag glaubt Harry, er sei ein  
ganz normaler Junge. Doch dann erfährt er, dass er sich an der Schule für  
Hexerei und Zauberei einfinden soll – denn er ist ein Zauberer! In Hogwarts  
stürzt Harry von einem Abenteuer ins nächste und muss gegen Bestien,  
Mitschüler*innen und Fabelwesen kämpfen. Da ist es gut, dass er schon Freunde  
gefunden hat, die ihm im Kampf gegen die dunklen Mächte zur Seite stehen. ",  
14    "pageCount": 350,  
15    "language": "Deutsch",  
16    "publisher": "Carlsen"  
17  }  
18 }
```

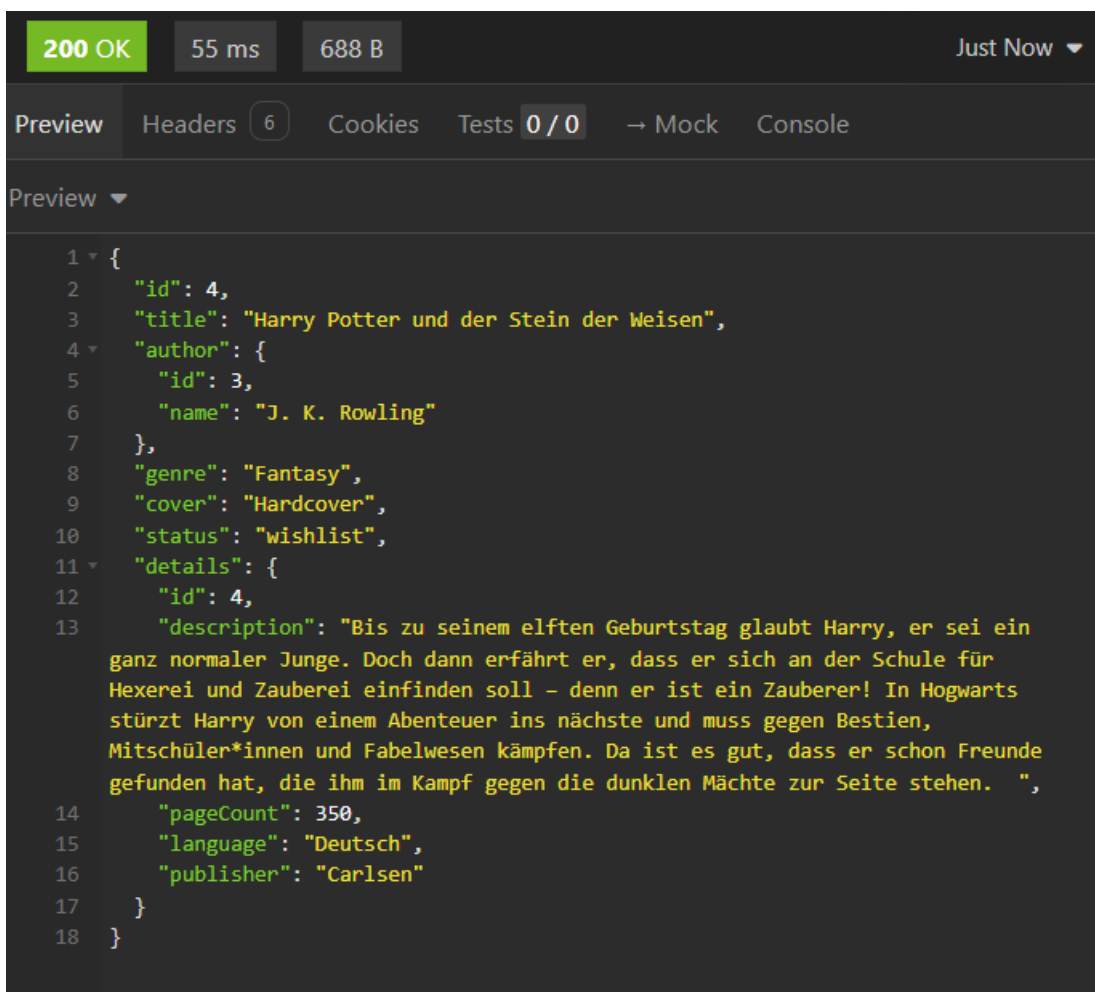
7.2.3 Einzelnes Buch abrufen

URL	http://localhost:8080/book/{id}
Methode	GET
Beschreibung	Gibt das Buch mit der angegebenen ID zurück.

Pfad-Parameter



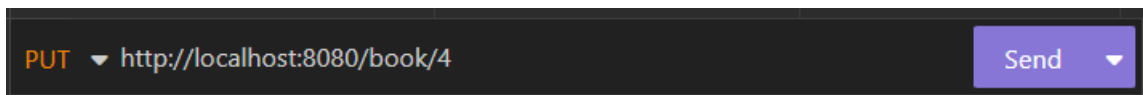
Ausgabe:



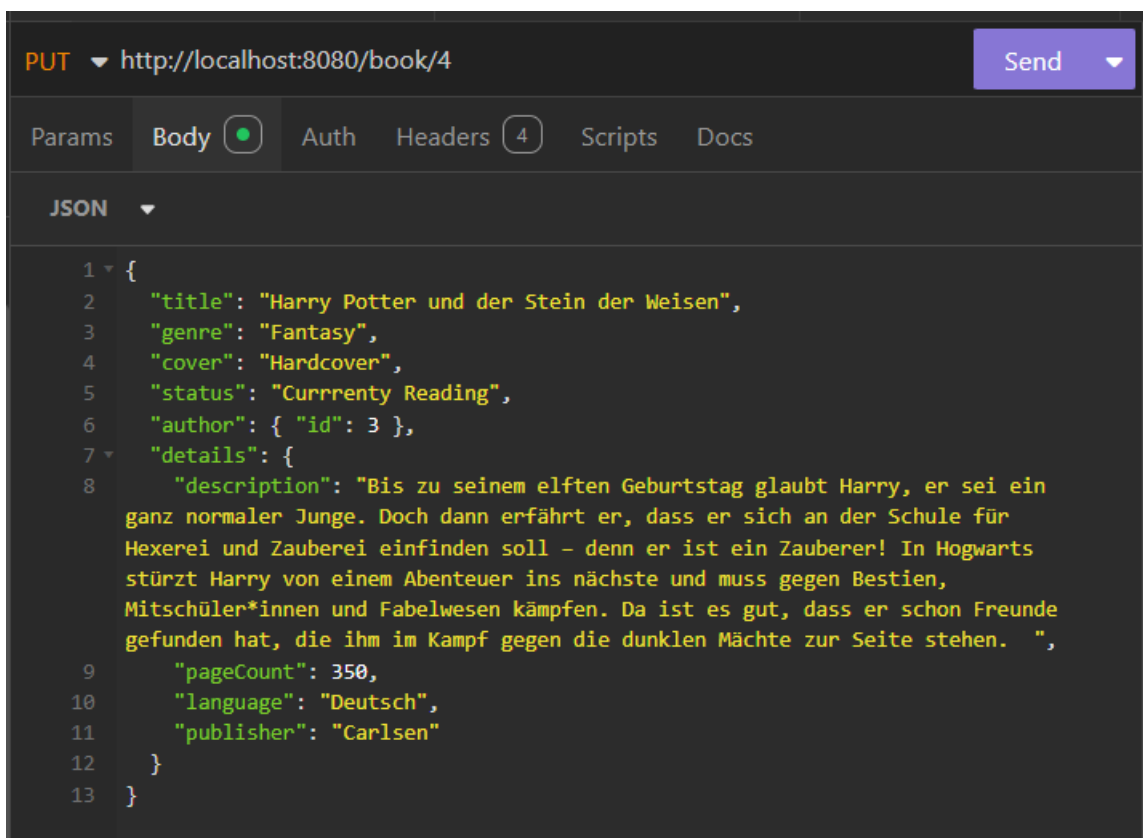
7.2.4 Buch aktualisieren

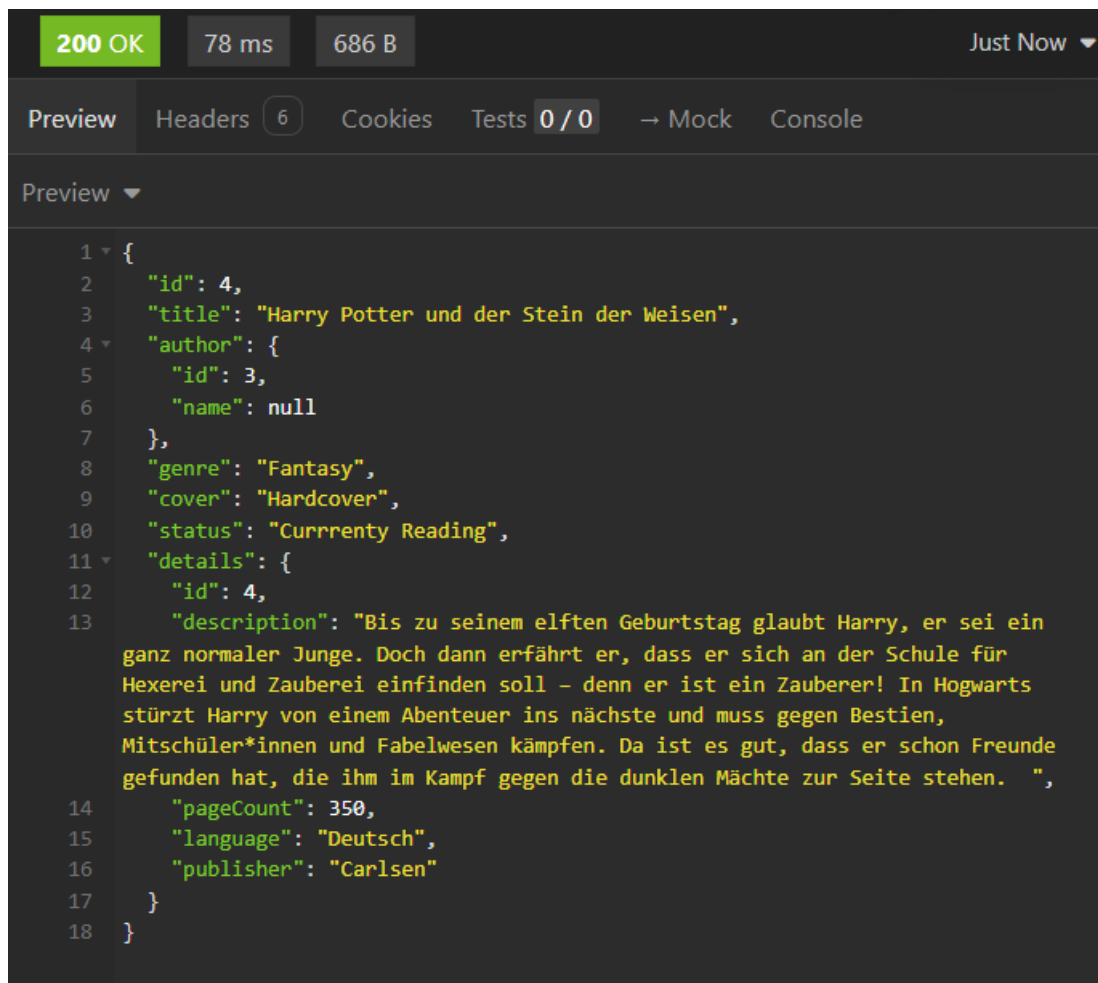
URL	http://localhost:8080/book/{id}
Methode	PUT
Beschreibung	Aktualisiert ein bestehendes Buch.

Pfad-Parameter



Request-Body:



Ausgabe:

200 OK 78 ms 686 B Just Now ▾

Preview Headers 6 Cookies Tests 0 / 0 → Mock Console

Preview ▾

```
1 {  
2   "id": 4,  
3   "title": "Harry Potter und der Stein der Weisen",  
4   "author": {  
5     "id": 3,  
6     "name": null  
7   },  
8   "genre": "Fantasy",  
9   "cover": "Hardcover",  
10  "status": "Currenty Reading",  
11  "details": {  
12    "id": 4,  
13    "description": "Bis zu seinem elften Geburtstag glaubt Harry, er sei ein  
    ganz normaler Junge. Doch dann erfährt er, dass er sich an der Schule für  
    Hexerei und Zauberei einfinden soll – denn er ist ein Zauberer! In Hogwarts  
    stürzt Harry von einem Abenteuer ins nächste und muss gegen Bestien,  
    Mitschüler*innen und Fabelwesen kämpfen. Da ist es gut, dass er schon Freunde  
    gefunden hat, die ihm im Kampf gegen die dunklen Mächte zur Seite stehen. ",  
14    "pageCount": 350,  
15    "language": "Deutsch",  
16    "publisher": "Carlsen"  
17  }  
18 }
```

7.2.5 Buch löschen

URL	http://localhost:8080/book/{id}
Methode	DELETE
Beschreibung	Löscht das Buch mit der angegebenen ID.

Pfad-Parameter

DELETE
▼
http://localhost:8080/book/3
Send
▼

Antwort:

200 OK
1.27 s
27 B
Just Now ▼

Preview
Headers 6
Cookies
Tests 0 / 0
→ Mock
Console

Preview ▼

```

1  Buch erfolgreich gelöscht.

```

7.3 REST-Schnittstellen - Autoren

7.3.1 Alle Autoren abrufen

URL	http://localhost:8080/author
Methode	GET
Beschreibung	Gibt eine Liste aller Autoren zurück.

Pfad-Parameter

GET ▼ <http://localhost:8080/author> Send ▼

Antwort

200 OK48 ms67 BJust Now ▼

PreviewHeaders6CookiesTests0/0→ MockConsole

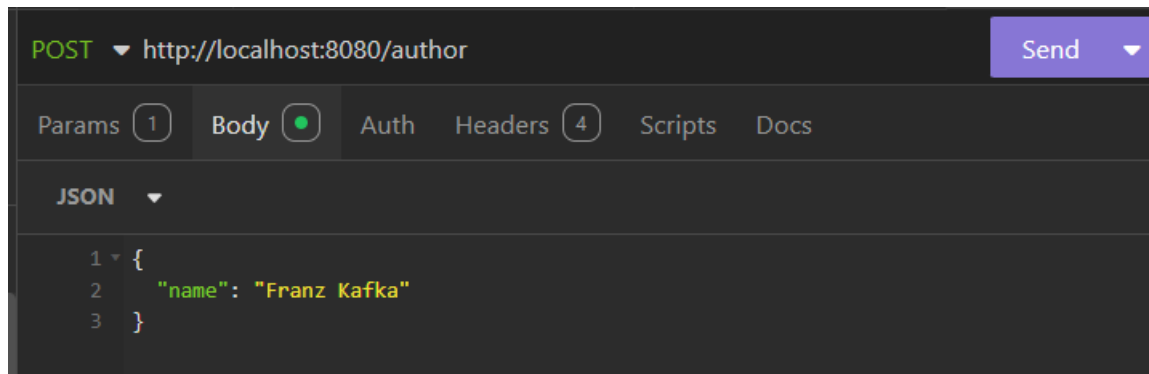
Preview ▼

```
1 [
2   {
3     "id": 3,
4     "name": "J. K. Rowling"
5   },
6   {
7     "id": 4,
8     "name": "Maurice Leblanc"
9   }
10 ]
```

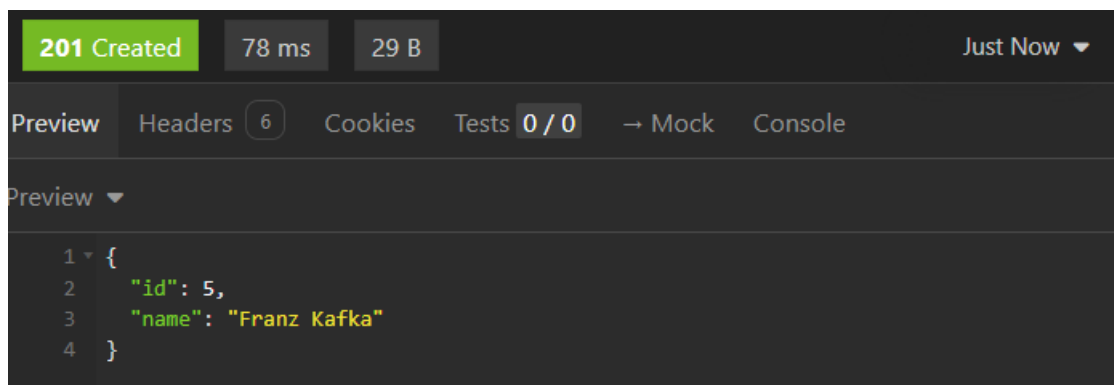
7.3.2 Neuen Autor anlegen

URL	http://localhost:8080/author
Methode	POST
Beschreibung	Legt einen neuen Autor an.

Request-Body



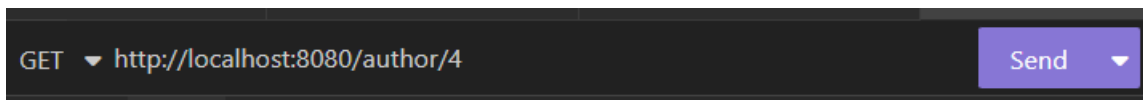
Antwort



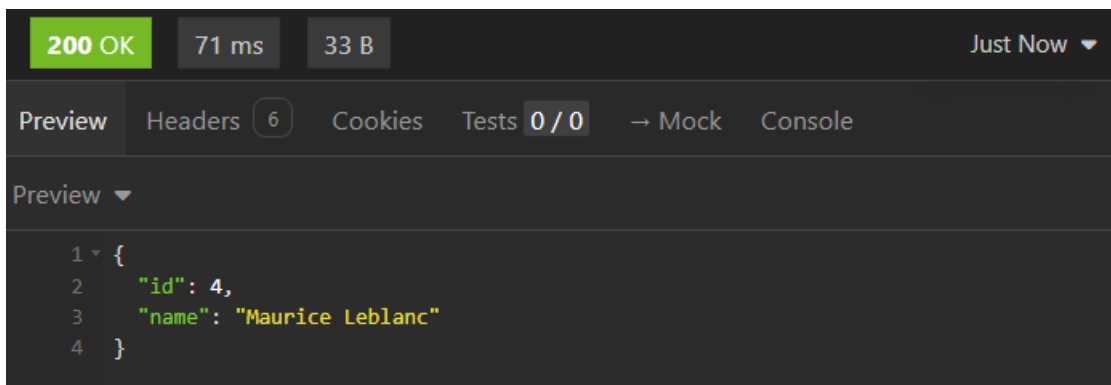
7.3.3 Einzelnen Autor abrufen

URL	http://localhost:8080/author/{id}
Methode	GET
Beschreibung	Gibt den Autor mit der angegebenen ID zurück.

Pfad-Parameter



Antwort:



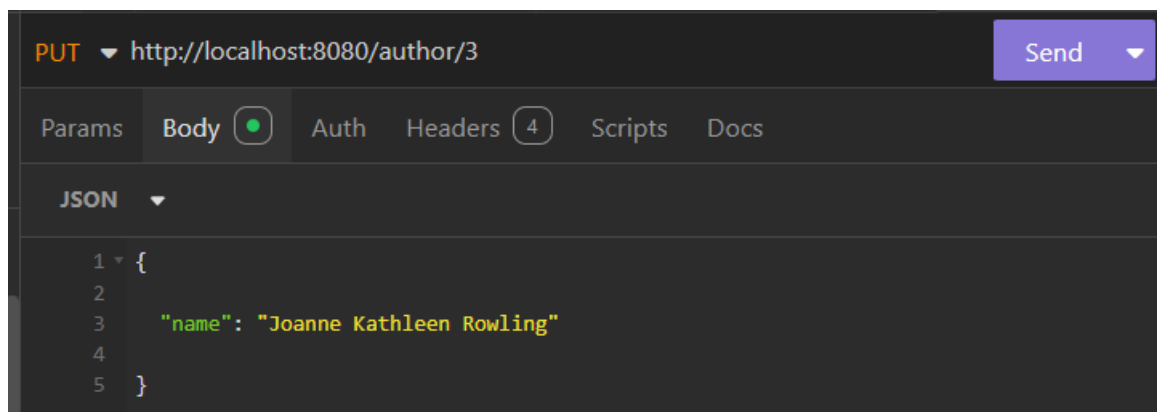
7.3.4 Einzelnen Autor abrufen

URL	http://localhost:8080/author/{id}
Methode	PUT
Beschreibung	Aktualisiert einen bestehenden Autor.

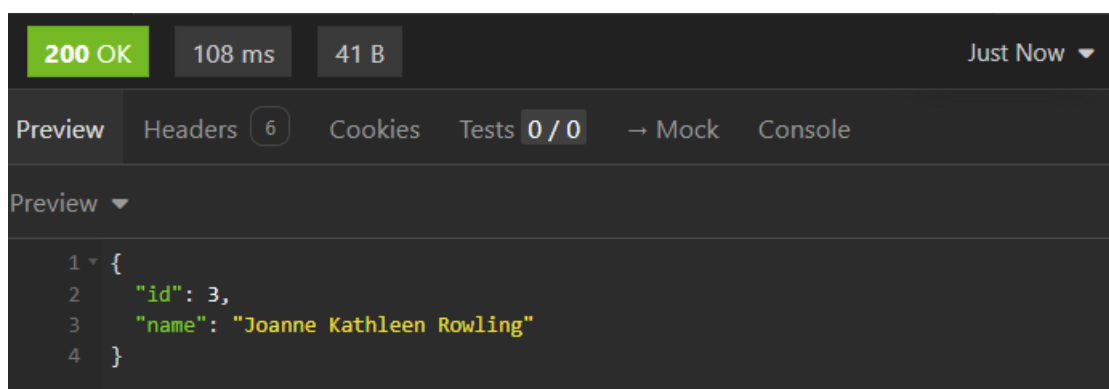
Pfad-Parameter



Request-Body



Antwort:



8 Testprotokoll

8.1 JUnit-Tests - Backend

8.1.1 BookValidationTest

Testfall-ID	TBookValid-001
Testklasse	BookValidationTest.java
Testmethode	testInvalidTitle_BlankOrTooShort()
Beschreibung	Überprüft die Validierung der Book-Entität bei einem ungültigen Titel (zu kurz).
Voraussetzung	<ul style="list-style-type: none">• Ein Validator ist initialisiert.• Ein gültiges Book-Objekt ist erstellt.• Der Titel des Buchs wird im Test geändert.
Testdaten	book.title = "HP 4" (nur 4 Zeichen, unter der angenommenen Mindestlänge von 5)
Erwartetes Ergebnis	"Title cannot be under 5 or exceed 100 characters."
Tatsächliches Ergebnis	"Title cannot be exceed 100 characters."
Status	Bestanden

8.1.2 BookTest

Testfall-ID	TBook-001
Testklasse	BookTest.java
Testmethode	testSetAndGetId()
Beschreibung	Überprüft, ob die ID korrekt gesetzt und abgerufen werden kann.
Voraussetzung	Ein Book-Objekt wurde im @BeforeEach-Setup korrekt initialisiert.
Testdaten	book.setId(4)
Erwartetes Ergebnis	book.getId() gibt 4 zurück.
Tatsächliches Ergebnis	Die Methode gibt den erwarteten Wert zurück.
Status	Bestanden

Testfall-ID	TBook-002
Testklasse	BookTest
Testmethode	testSetAndGetStatus()
Beschreibung	Überprüft, ob der Lesestatus korrekt gesetzt und gelesen werden kann.
Voraussetzung	Ein Book-Objekt wurde im @BeforeEach-Setup korrekt initialisiert.
Testdaten	book.setStatus("Read")
Erwartetes Ergebnis	book.getStatus() gibt "Read" zurück.
Tatsächliches Ergebnis	Die Methode gibt den erwarteten Wert zurück.
Status	Bestanden

8.1.3 AuthorTest

Testfall-ID	TAuthor-001
Testklasse	AuthorTest.java
Testmethode	testAddBook()
Beschreibung	Überprüft das Hinzufügen eines Buchs zu einem Autor sowie die korrekte bidirektionale Verknüpfung zwischen Author und Book.
Voraussetzung	<ul style="list-style-type: none"> • Ein Author-Objekt ist initialisiert. • Ein Book-Objekt (book1) ist initialisiert.
Testdaten	author.addBook(book1)
Erwartetes Ergebnis	<ul style="list-style-type: none"> • Die Bücherliste des Autors enthält book1. • book1.getAuthor() gibt den Autor zurück.
Tatsächliches Ergebnis	<ul style="list-style-type: none"> • author.getBooks().size() == 1 • author.getBooks().contains(book1) == true • book1.getAuthor() == author
Status	Bestanden

Testfall-ID	TAuthor-002
Testklasse	AuthorTest.java
Testmethode	testRemoveBook()
Beschreibung	Überprüft das Entfernen eines Buchs vom Autor und die korrekte Aufhebung der bidirektionalen Verknüpfung.
Voraussetzung	<ul style="list-style-type: none"> • Der Autor wurde mit book1 verknüpft. • author.addBook(book1) wurde im Test aufgerufen.
Testdaten	author.removeBook(book1)
Erwartetes Ergebnis	<ul style="list-style-type: none"> • Die Bücherliste des Autors ist leer. • book1.getAuthor() gibt null zurück.
Tatsächliches Ergebnis	<ul style="list-style-type: none"> • author.getBooks().isEmpty() == true • book1.getAuthor() == null
Status	Bestanden

8.2 JUnit-Tests - Frontend

8.2.1 AuthorFormTest

Testfall-ID	TAuthorForm-001
Testdatei	AuthorForm.test.jsx
Testmethode	test('zeigt Fehler an, wenn der Name zu kurz ist')
Beschreibung	Überprüft die Validierung, wenn der eingegebene Autorenname kürzer als fünf Zeichen ist.
Voraussetzung	<ul style="list-style-type: none">• Komponente AuthorForm wird korrekt gerendert.• useNavigate und authorService.addAuthor sind gemockt.
Testdaten	Eingabe: "JKR" (3 Zeichen)
Erwartetes Ergebnis	Eine Fehlermeldung wird angezeigt: "Bitte Autornamen mit mindestens fünf Zeichen eingeben"
Tatsächliches Ergebnis	Die erwartete Fehlermeldung ist im DOM vorhanden (toBeInTheDocument() schlägt nicht fehl).
Status	Bestanden

8.2.2 AuthorListTest

Testfall-ID	TAuthorList-001
Testdatei	AuthorList.test.jsx
Testmethode	test('zeigt die Autorenliste nach erfolgreichem Laden an')
Beschreibung	Überprüft, ob nach erfolgreichem Laden der Daten die Autorenliste angezeigt wird.
Voraussetzung	<ul style="list-style-type: none">• global.fetch ist gemockt und gibt eine Liste von zwei Autoren zurück.• Link-Komponente aus react-router-dom wurde gemockt.
Testdaten	mockAuthors = [{ id: 1, name: 'Autor A' }, { id: 2, name: 'Autor B' }]
Erwartetes Ergebnis	<ul style="list-style-type: none">• Überschrift "Autorenliste" ist im DOM sichtbar.• "Autor A" und "Autor B" werden angezeigt.• Zwei "Bearbeiten"-Buttons werden gerendert.
Tatsächliches Ergebnis	Alle erwarteten Elemente wurden korrekt angezeigt.
Status	Bestanden

Testfall-ID	TAuthorList-002
Testdatei	AuthorList.test.jsx
Testmethode	test('zeigt "Keine Autoren gefunden." an, wenn die Liste leer ist')
Beschreibung	Überprüft das Verhalten der Komponente, wenn keine Autoren vom Server zurückgegeben werden.
Voraussetzung	global.fetch ist gemockt und gibt ein leeres Array zurück.
Testdaten	[] (leere Autorenliste)
Erwartetes Ergebnis	Die Meldung "Keine Autoren gefunden." wird angezeigt.
Tatsächliches Ergebnis	Die erwartete Meldung wurde korrekt im DOM angezeigt.
Status	Bestanden

8.2.3 BookEdit.test

Testfall-ID	TBookEdit-001
Testdatei	BookEdit.test.jsx
Testmethode	test('ruft deleteBook auf und navigiert bei erfolgreichem Löschen')
Beschreibung	Testet, ob beim Klicken auf „Buch löschen“ die deleteBook-Funktion aufgerufen und danach zur Buchliste navigiert wird.
Voraussetzung	<ul style="list-style-type: none">• useParams liefert id = 1 zurück.• global.fetch liefert das Buch mit ID 1 sowie eine Autorenliste.• mockNavigate ist als Mock gesetzt.• mockDeleteBook ist als Mockfunktion gesetzt.
Testdaten	<ul style="list-style-type: none">• mockBook mit ID 1, Titel, Autor und Details.• mockAuthors mit zwei Autoren.
Erwartetes Ergebnis	<ul style="list-style-type: none">• Das Eingabefeld für den Titel (mockBook.title) ist sichtbar (indiziert geladenes Buch).• Nach Klick auf „Buch löschen“:• mockDeleteBook wird einmal mit 1 (Buch-ID) aufgerufen.• Navigation erfolgt zu '/bookList'.
Tatsächliches Ergebnis	<ul style="list-style-type: none">• mockDeleteBook wurde korrekt aufgerufen.• Navigation zu '/bookList' wurde korrekt ausgelöst.
Status	Bestanden

8.2.4 BookFilter.test

Testfall-ID	TBookFilter-001
Testdatei	BookFilter.test.js
Testmethode	test('ruft onGenreFilter mit korrektem Wert auf, wenn Genre ausgewählt ist')
Beschreibung	Überprüft, ob die Callback-Funktion onGenreFilter mit dem erwarteten Wert aufgerufen wird, wenn ein Genre ausgewählt wird.
Voraussetzung	<ul style="list-style-type: none"> Die BookFilter-Komponente wird mit drei Mock-Funktionen (onGenreFilter, onStatusFilter, onCoverFilter) gerendert. Alle Mocks sind vor dem Test zurückgesetzt.
Testdaten	genreSelect.value = 'fantasy'
Erwartetes Ergebnis	<ul style="list-style-type: none"> mockOnGenreFilter wird einmal aufgerufen mit 'fantasy'. mockOnStatusFilter und mockOnCoverFilter werden nicht aufgerufen.
Tatsächliches Ergebnis	<ul style="list-style-type: none"> mockOnGenreFilter wurde einmal mit 'fantasy' aufgerufen. Die anderen Callback-Funktionen wurden nicht aufgerufen.
Status	Bestanden

9 Installationsanleitung

9.1 Allgemein

1. Entpacken Sie den ZIP-Ordner „**LB-Projekt-M294_M295_Prasanna_Thanikasalam**» auf Ihrem Computer.

9.2 Backend starten

1. Öffnen Sie **Eclipse** und importieren Sie den backend-Ordner:
 - Wählen Sie im Menü: **File > Import > Existing Maven Projects**
 - Wählen Sie den Ordner „**backend**“ aus und bestätigen Sie mit **Finish**
 - Der Ordner wird im Eclipse als «**lb295 (in backend)**» angezeigt.
2. Navigieren Sie im Projektbaum zu folgendem Pfad:
src/main/java/bookmanagementtool/BookManagementApplication.java
3. Öffnen Sie die Datei **BookManagementApplication.java**, klicken Sie sie an.
4. Drücken Sie oben in der Mitte auf den **grünen „Run“-Button**, um das Backend zu starten.
 - Das Backend startet nun automatisch.

9.3 Frontend starten

1. Öffnen Sie den Ordner:
„**LB-Projekt-M294_M295_Prasanna_Thanikasalam\frontend\src**“ auf Ihrem Computer.

Klicken Sie in die Adresszeile oben im Ordnerfenster und geben Sie **cmd** ein – drücken Sie **Enter**.

- Dadurch öffnet sich das Terminal direkt im richtigen Ordner.
2. Geben Sie im Terminal den folgenden Befehl ein und drücken Sie **Enter**:
 - «npm run dev»
- Das Frontend wird nun gestartet und sollte im Browser die Seite mit dem URL «http://localhost:3000/home» angezeigt werden.

10 Hilfestellungen

Tool/Programm	Verwendungszweck
Visual Studio Code	Entwicklung des Frontends (React, JavaScript, CSS, etc.)
Eclipse	Entwicklung des Backends (Java, Spring Boot)
draw.io	Erstellung von Storyboards und Ablaufdiagrammen
Violet UML Editor	Erstellung von Klassendiagrammen
ChatGPT	Fehleranalyse, Fehlerbehandlung, Sprachkorrektur / Formulierung von Projektdokumentation.
DeepSeek	Fehleranalyse und Fehlerbehandlung
Gemini	Sprachkorrektur / Formulierung von Projektdokumentation, Erstellung von Javadoc

11 Bildquelle

Titelbild: https://de.freepik.com/premium-ki-bild/buecherregal-cartoon_53680431.htm

12 Troubleshooting

Wenn der Autor in der Bücherliste nach dem Erfassen eines Buches als «Unknown Author» angezeigt wird, müssen Sie die Seite aktualisieren, damit die vollständigen Daten geladen werden.