

Abschlussprojekt - Buchbewertungsanwendung

Geschätzte benötigte Zeit: 2 Stunden

- In diesem Abschlussprojekt werden wir eine serverseitige Online-Buchbewertungsanwendung erstellen und sie mit einem sicheren REST-API-Server integrieren, der die Authentifizierung auf Sitzungsbasis mithilfe von JWT verwendet. Anschließend werden Sie Ihre Anwendung mit Promises-Callbacks oder Async-Await-Funktionen testen.

Ziele:

Nach Abschluss dieses Labors werden Sie in der Lage sein:

1. APIs zu erstellen und CRUD-Operationen auf einem Express-Server unter Verwendung von Sitzungs- und JWT-Authentifizierung durchzuführen.
2. Async/Await oder Promises mit Axios in Node.js zu verwenden.
3. REST-API-Endpunkte zu erstellen und diese mit Postman zu testen.

Einrichten : Anwendung erstellen

1. Öffnen Sie ein Terminalfenster über das Menü im Editor: Terminal > Neues Terminal.
2. Wechseln Sie in Ihren Projektordner, falls Sie sich noch nicht im Projektordner befinden.

```
cd /home/project
```

3. Bitte forken Sie das Git-Repository, das den Starter-Code für dieses Labor enthält:

```
https://github.com/ibm-developer-skills-network/expressBookReviews.git
```

4. Klonen Sie das geforkte Git-Repository, falls es noch nicht existiert.

```
[ ! -d 'expressBookReviews' ] && git clone https://github.com/<your Github username>/expressBookReviews.git
```

Wechseln Sie in das Verzeichnis **expressBookReviews/final_project/**, um mit der Arbeit am Labor zu beginnen.

```
cd expressBookReviews/final_project/
```

6. Listen Sie den Inhalt dieses Verzeichnisses auf, um die Artefakte für dieses Labor zu sehen.

```
ls
```

Verstehen der Serveranwendung

1. Öffnen Sie im Datei-Explorer den Ordner **expressBookReviews/final_project/** und sehen Sie sich `index.js` mit dem folgenden Code an:

```
const express = require('express');
const jwt = require('jsonwebtoken');
const session = require('express-session')
const customer_routes = require('./router/auth_users.js').authenticated;
const genl_routes = require('./router/general.js').general;
const app = express();
app.use(express.json());
app.use("/customer",session({secret:"fingerprint_customer",resave: true, saveUninitialized: true}))
app.use("/customer/auth/*", function auth(req,res,next){
  //Write the authentication mechanism here
});

const PORT =5000;
app.use("/customer", customer_routes);
app.use("/", genl_routes);
app.listen(PORT,()=>console.log("Server is running"));
```

2. Die für dieses Labor erforderlichen Pakete sind als Abhängigkeiten in `packages.json` wie folgt definiert:

```
"dependencies": {
  "express": "^4.18.1",
  "express-session": "^1.17.3",
  "jsonwebtoken": "^8.5.1",
  "nodemon": "^2.0.19"
}
```

Verstehen der Benutzerpfade

Navigiere zum router-Verzeichnis mit den folgenden 3 Dateien:

1. `booksdb.js` - Dies enthält die vorab geladenen Buchinformationen für diese Anwendung.
2. `general.js` - Dies enthält die grundlegenden Implementierungen für die Routen, auf die ein allgemeiner Benutzer zugreifen kann.
3. `auth_users.js` - Dies enthält die grundlegenden Implementierungen für die Routen, auf die ein autorisierter Benutzer zugreifen kann.

Aktualisierung des Codes für den Authentifizierungsmechanismus:

- Navigiere zu `index.js` und aktualisiere den Authentifizierungscode unter `app.use("/customer/auth/*", function auth(req, res, next){`:

Hinweis: Verwende die Sitzungsautorisierungsfunktion (implementiert im Praxisprojekt-Labor), um einen Benutzer basierend auf dem Zugriffstoken zu authentifizieren.

Aktualisieren und Testen der allgemeinen Benutzer-Routen in `general.js`

Hinweis:

- Bitte denken Sie daran, Ihre Arbeit nach Abschluss jeder Aufgabe oder wenn Sie das Labor nicht in einer Sitzung abschließen, in das GitHub-Repository zu pushen. Die Schritte zum Pushen des Repositories zu GitHub finden Sie in Aufgabe 14. Dies hilft Ihnen, Ihren Fortschritt nicht zu verlieren.
- Wenn Sie Schwierigkeiten haben, Postman zu verwenden, können Sie alternativ `curl`-Befehle verwenden, um die API-Endpunkte zu testen, wie im [Modul 3: Praktisches Labor: CRUD-Operationen mit Node.js und Express](#) demonstriert. Bitte fügen Sie Screenshots bei, die die JSON-Ausgabe Ihrer `curl`-Befehle in Ihrer Einreichung zur Peer-Überprüfung zeigen.

Aufgabe 1:

- Vervollständigen Sie den Code zum Abrufen der Liste der im Shop verfügbaren Bücher unter `public_users.get('/',function (req, res) {`.

Hinweis: Verwenden Sie die Methode `JSON.stringify`, um die Ausgabe ordentlich anzuzeigen.

- Führen Sie `npm install` aus, um die erforderlichen Module zu installieren, und starten Sie den Server.
- Testen Sie die Ausgabe in Postman.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen `1-getallbooks.png` zur Einreichung unter Aufgabe 1 für die Peer-Überprüfung.

Aufgabe 2:

- Vervollständigen Sie den Code zum Abrufen der Buchdetails basierend auf der ISBN unter `public_users.get('/isbn/:isbn',function (req, res) {.`

Hinweis: Rufen Sie die ISBN aus den Anforderungsparametern ab.

- Testen Sie die Ausgabe in Postman.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen `2-getdetailsISBN.png` zur Einreichung unter Aufgabe 2 für die Peer-Überprüfung.

Aufgabe 3:

- Vervollständigen Sie den Code zum Abrufen der Buchdetails basierend auf dem Autor unter `public_users.get('/author/:author',function (req, res) {.`

Hinweise:

1. Erhalten Sie alle Schlüssel für das `'books'`-Objekt.
2. Iterieren Sie durch das `'books'`-Array und prüfen Sie, ob der Autor mit dem in den Anforderungsparametern angegebenen übereinstimmt.

- Testen Sie die Ausgabe in Postman.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen `3-getbooksbyauthor.png` zur Einreichung unter Aufgabe 3 für die Peer-Überprüfung.

Aufgabe 4:

- Vervollständigen Sie den Code zum Abrufen der Buchdetails basierend auf dem Titel unter `public_users.get('/title/:title',function (req, res) {.`

Hinweis: Dies wird ähnlich wie Übung 3 sein.

- Testen Sie die Ausgabe in Postman.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen `4-getbooksbytitle.png` zur Einreichung unter Aufgabe 4 für die Peer-Überprüfung.

Aufgabe 5:

- Vervollständigen Sie den Code zum Abrufen von Buchrezensionen unter `public_users.get('/review/:isbn',function (req, res) {.`

Hinweis: Holen Sie sich die Buchrezensionen basierend auf der in den Anforderungsparametern angegebenen ISBN.

- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen `5-getbookreview.png` zur Einreichung unter Aufgabe 5 für die Peer-Überprüfung.

Aufgabe 6:

- Vervollständigen Sie den Code zur Registrierung eines neuen Benutzers.

Hinweis: Der Code sollte den 'Benutzernamen' und das 'Passwort', die im Body der Anfrage zur Registrierung bereitgestellt werden, übernehmen. Wenn der Benutzername bereits existiert, muss dies erwähnt werden und es müssen auch andere Fehler angezeigt werden, z. B. wenn Benutzername und/oder Passwort nicht bereitgestellt werden.

- Testen Sie die Ausgabe in Postman.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen `6-register.png` zur Einreichung unter Aufgabe 6 für die Peer-Überprüfung.

Aktualisieren und Testen der authentifizierten Benutzer-Routen in `auth_users.js`.

Aufgabe 7:

- Vervollständigen Sie den Code für das Einloggen als registrierter Benutzer.

Hinweis: Der Code muss einen Kunden basierend auf dem Benutzernamen und Passwort, die in Übung 6 erstellt wurden, validieren und anmelden. Außerdem müssen die Benutzeranmeldeinformationen für die Sitzung als JWT gespeichert werden. Da Sie sich als Kunde anmelden müssen, verwenden Sie beim Testen der Ausgabe in Postman den Endpunkt als "customer/login".

- Testen Sie die Ausgabe in Postman.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen 7-login.png zur Einreichung unter Aufgabe 7 für die Peer Review Aufgabe.

Aufgabe 8:

- Vervollständigen Sie den Code zum Hinzufügen oder Ändern einer Buchbewertung.

Hinweis: Sie müssen eine Bewertung als Anfrageabfrage angeben, und sie muss mit dem Benutzernamen (der in der Sitzung gespeichert ist) gepostet werden. Wenn derselbe Benutzer eine andere Bewertung zur gleichen ISBN postet, sollte die bestehende Bewertung geändert werden. Wenn ein anderer Benutzer sich anmeldet und eine Bewertung zur gleichen ISBN abgibt, wird sie als andere Bewertung unter derselben ISBN hinzugefügt.

- Testen Sie die Ausgabe in Postman.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen 8-reviewadded.png zur Einreichung unter Aufgabe 8 für die Peer Review Aufgabe.

Aufgabe 9:

Vervollständigen Sie den Code zum Löschen einer Buchbewertung unter `regd_users.delete("/auth/review/:isbn", (req, res) => {`

Hinweis: Filtern und löschen Sie die Bewertungen basierend auf dem Benutzernamen der Sitzung, damit ein Benutzer nur seine eigenen Bewertungen und nicht die anderer Benutzer löschen kann.

- Testen Sie die Ausgabe in Postman.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen 9-deletereview.png zur Einreichung unter Aufgabe 9 für die Peer Review Aufgabe.

Damit haben Sie die Codes für die allgemeinen und authentifizierten Benutzer-Routen implementiert und getestet.

Verbesserung des Umfangs der Aufgaben 1-4 mit Promises oder Async-Await

Sie werden jetzt Promise-Callbacks oder Async-Await-Funktionen verwenden, um dieselbe Funktionalität zu erreichen, die wir synchron in den Aufgaben 1-4 behandelt haben.

Hinweis: Bitte machen Sie einen Screenshot Ihrer Code-Implementierung mit entweder `async/await` oder Promises für die Aufgaben 10-13. Dieser Screenshot wird für die Peer-Review-Einreichung verwendet.

Aufgabe 10:

- Fügen Sie den Code hinzu, um die Liste der im Geschäft verfügbaren Bücher abzurufen (in Aufgabe 1 erledigt) mithilfe von Promise-Callbacks oder `async-await` mit Axios.

Hinweis: Verweisen Sie auf [dieses](#) Labor zu Promises und Callbacks.

- Bitte stellen Sie sicher, dass die Datei `general.js` den Code zum Abrufen der Liste der im Geschäft verfügbaren Bücher mithilfe von Promise-Callbacks oder `async-await` mit Axios enthält.

- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen task10.png zur Einreichung unter Aufgabe 10 für die Peer-Review-Aufgabe.

Aufgabe 11:

- Fügen Sie den Code hinzu, um die Buchdetails basierend auf der ISBN abzurufen (in Aufgabe 2 erledigt) mithilfe von Promise-Callbacks oder async-await mit Axios.

Hinweis: Verweisen Sie auf [dieses](#) Labor zu Promises und Callbacks.

- Bitte stellen Sie sicher, dass die Datei general.js den Code zum Abrufen der Buchdetails basierend auf der ISBN mithilfe von Promise-Callbacks oder async-await mit Axios enthält.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen task11.png zur Einreichung unter Aufgabe 11 für die Peer-Review-Aufgabe.

Aufgabe 12:

- Fügen Sie den Code hinzu, um die Buchdetails basierend auf dem Autor abzurufen (in Aufgabe 3 erledigt) mithilfe von Promise-Callbacks oder async-await mit Axios.

Hinweis: Verweisen Sie auf [dieses](#) Labor zu Promises und Callbacks.

- Bitte stellen Sie sicher, dass die Datei general.js den Code zum Abrufen der Buchdetails basierend auf dem Autor mithilfe von Promise-Callbacks oder async-await mit Axios enthält.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen task12.png zur Einreichung unter Aufgabe 12 für die Peer-Review-Aufgabe.

Aufgabe 13:

- Fügen Sie den Code hinzu, um die Buchdetails basierend auf dem Titel abzurufen (in Aufgabe 4 erledigt) mithilfe von Promise-Callbacks oder async-await mit Axios.

Hinweis: Verweisen Sie auf [dieses](#) Labor zu Promises und Callbacks.

- Bitte stellen Sie sicher, dass die Datei general.js den Code zum Abrufen der Buchdetails basierend auf dem Titel mithilfe von Promise-Callbacks oder async-await mit Axios enthält.
- Bitte machen Sie einen Screenshot davon und speichern Sie ihn unter dem Namen task13.png zur Einreichung unter Aufgabe 13 für die Peer-Review-Aufgabe.

Github-Repo-Aktualisierung für die Peer-Review-Einreichung

Aufgabe 14:

- Bitte committen und pushen Sie alle Änderungen in Ihr geforktes Github-Repo.

Hinweis: Bitte beziehen Sie sich auf [dieses](#) Labor, wenn Sie Hilfe beim Committen und Pushen von Änderungen in Ihr Repo benötigen.

- Ihr Github-Repo-Link wird für die Bewertung von Aufgabe 14 in der Peer-Review-Aufgabe verwendet.

Herzlichen Glückwunsch! Sie haben das Abschlussprojekt abgeschlossen!

Zusammenfassung:

In diesem Labor haben Sie eine serverseitige Online-Buchbewertungsanwendung erstellt, diese mit einer sicheren REST-API unter Verwendung von JWT-basierter Sitzungsauthentifizierung integriert und die erstellte Anwendung mit Promises-Callbacks oder Async-Await-Funktionen getestet.

Autor(en)

Lavanya T S

Sapthashree K S

K Sundararajan

© IBM Corporation. Alle Rechte vorbehalten.