



Service Layer

Saron Berhane

July 16, 2023

Overview:

The backend service for the Book review and recommendation web application will be built using the Express library from NodeJS. It will be a simple REST API that the front end will call to create and manipulate data.

Specifications:

- **GET /api/books**

This will retrieve a list of books.

- Request: No additional request data.
- Response: JSON array of book objects.

- **GET /api/books/{bookId}**

This will retrieve details of a specific book.

- Request: No additional request data.
- Response: JSON object containing book details.

- **POST /api/books**

This will create a new book.

- Request:

```
{
  "title": "New Life",
  "author": "Saron Berhane",
  "genre": "Romance",
  "publication_year": 153,
```

- ```

 "description": "A classic novel about moving on."
 }

```
- Response: JSON object containing the created book details.

- **PUT /api/books/{bookId}**

This will Update the details of a specific book.

- Request:
 

```

{
 "title": "New Title"
}

```

- **DELETE /api/books/{bookId}**

This will be to delete a specific book.

- Request: No additional request data.
- Response: No response data.

- **GET /api/users/{userId}/favorites**

This will be to retrieve the list of favorite books for a user.

- Request: No additional request data.
- Response: JSON array of book objects (user's favorite books).

- **POST /api/users/{userId}/favorites**

This will be used to add a books to a user's favorites.

- Request:
 

```

{
 "bookId": "61f1s5a9e3a7e7a8d5d2b"
}

```
- Response: JSON object indicating the successful addition of the book to favorites.

- **DELETE /api/users/{userId}/favorites/{bookId}**

This will be for removing a book from a user's favorites.

- Request: No additional request data.
- Response: No response data.

- **GET /api/posts**

This is for retrieving a list of posts.

- Request: No additional request data.
- Response: JSON array of post objects.

- **GET /api/posts/{postId}**

This is for retrieving details of a specific post.

- Request: No additional request data.
- Response: JSON object containing post details.

- **POST /api/posts**

This is for creating a new post.

- Request:
 

```

{
 "title": "New Post",

```

```
 "content": "This is a new post about books.",
 "author": "saronberhane@gmail.com"
 }
```

- Response: JSON object containing the created post details.

- **PUT /api/posts/{postId}**

This is for updates on the details of a specific post.

- Request:

```
{
 "title": "Updated Title"
}
```

- Response: JSON object containing the updated post details.

- **DELETE /api/posts/{postId}**

This is for deleting a specific post.

- Request: No additional request data.
- Response: No response data.

- **POST /api/login**

This is for authenticating a user and generating a session token.

- Request:

```
{
 "email": " saronberhane@gmail.com ",
 "password": "pass123"
}
```

- Response: JSON object containing the session token for authentication.

- **POST /api/signup**

This is to register a new user.

- Request:

```
{
 "name": "Saron Berhane",
 "email": " saronberhane@gmail.com ",
 "password": "pass123"
}
```

- Response: JSON object indicating successful user registration.

- **GET /api/reviews/{bookId}**

This is to retrieve reviews for a specific book.

- Request: No additional request data.
- Response: JSON array of review objects for the specified book.

- **POST /api/reviews/{bookId}**

This is to create a new review for a specific book.

- Request:

```
{
 "user": " saronberhane@gmail.com ",
 "rating": 3.5,
 "comment": "Great book!"
}
```

}

- Response: JSON object indicating the successful creation of the review.

- **GET /api/inquiries**

This is to retrieve a list of inquiries.

- Request: No additional request data.
- Response: JSON array of inquiry objects.

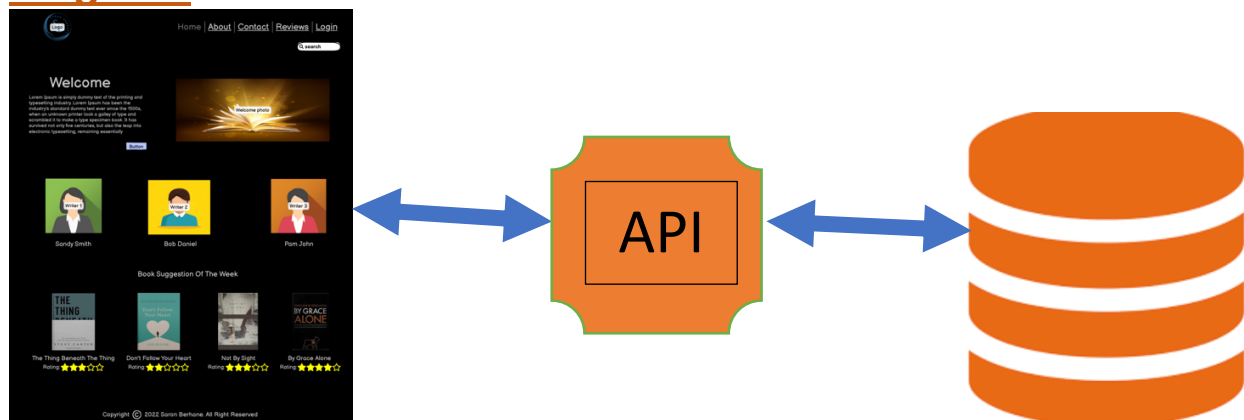
- **POST /api/inquiries**

This is to Create a new inquiry.

- Request:

```
{
 "user": "saronberhane@gmail.com ",
 "subject": "Book recommendation",
 "message": "Can you suggest some good novels?"
}
```
- Response: JSON object indicating the successful creation of the inquiry.

## Diagram:



The graphic depicts the flow of communication from user interface pages to service endpoints. The title of each page represents it, and arrows indicate the direction of requests sent to certain endpoints. This diagram confirms that the designed services and endpoints meet the minimal viable product (MVP) criteria and aids in visualizing the communication paths between the user interface and backend services.

The service layer architecture provides a clear and complete means for the user interface to communicate with the database by adhering to RESTful principles, correctly leveraging HTTP verbs, and providing logical and consistent pathways for endpoints. The

accompanying sample requests and answers, including handling erroneous requests, show the endpoints' anticipated behavior.