

Advanced Server Side Programming (Lab)

Assessment **REST Web Service** **Library Management System**

Name : Brian E Shilo

Reg No. : 21MIC0131

Lab Slot : L21 + L22

Course code : CSI3023

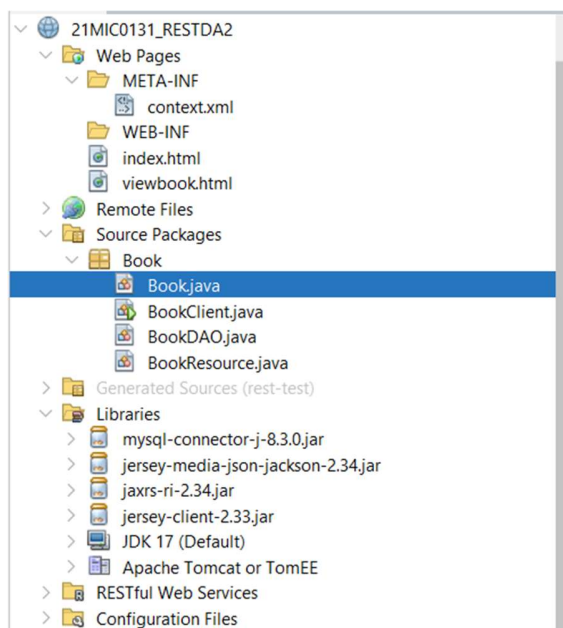
Problem Statement :“To develop Library management system”

Develop the REST application for the following operations:

- i) Create the web method to receive the BOOK information such as ISBN number, title, price, publisher from the client through the form element
- ii) Store the same into a database
- iii) Retrieve a book information of title="Advanced Server side programming" from a database.
- iv) Create the Java REST client to consume the web services as JSON type

IMPLEMENTATION

Project Structure



Made using Netbeans IDE20 and Server integration(Apache TomCat)

Programs - Codes

Client side

1. index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Book Information Form</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>

<h2>Enter Book Information</h2>

<form id="bookForm">
  <label for="isbn">ISBN:</label><br>
  <input type="text" id="isbn" name="isbn"><br>

  <label for="title">Title:</label><br>
  <input type="text" id="title" name="title"><br>

  <label for="price">Price:</label><br>
  <input type="text" id="price" name="price"><br>

  <label for="publisher">Publisher:</label><br>
  <input type="text" id="publisher" name="publisher"><br>

  <input type="submit" value="Add Book">
  <button type="button" id="viewBook">View Book Info</button>
```

</form>

<script>

```
$(document).ready(function() {  
    $('#bookForm').submit(function(event) {  
        event.preventDefault();
```

```
  
        var book = {  
            isbn: $('#isbn').val(),  
            title: $('#title').val(),  
            price: $('#price').val(),  
            publisher: $('#publisher').val()  
        };
```

```
  
        $.ajax({  
            url: 'http://localhost:8080/21MIC0131_RESTDA2/api/books',  
            type: 'POST',  
            contentType: 'application/json',  
            data: JSON.stringify(book),  
            dataType: 'json',  
            success: function(response) {  
                alert('Book information submitted successfully!');  
                $('#bookForm')[0].reset();  
            },  
            error: function(error) {  
                alert('Error submitting book information: ' + error.responseText);  
            }  
        });  
    });
```

```
$('#viewBook').click(function() {  
    var isbn = $('#isbn').val();  
    window.location.href = 'viewbook.html?isbn=' + isbn;  
});  
});  
</script>  
</body>  
</html>
```

2. BookClient.java

```
package Book;  
  
import javax.ws.rs.client.Client;  
import javax.ws.rs.client.ClientBuilder;  
import javax.ws.rs.client.Entity;  
import javax.ws.rs.client.WebTarget;  
import javax.ws.rs.core.MediaType;  
import javax.ws.rs.core.Response;  
  
public class BookClient {  
  
    private static final String BASE_URL =  
        "http://localhost:8080/21MIC0131_RESTDA2/api/books";  
  
    public static void main(String[] args) {  
        Client client = ClientBuilder.newClient();  
        WebTarget target = client.target(BASE_URL);
```

```

// Create a new book
Book newBook = new Book();
newBook.setIsbn("1234567890");
newBook.setTitle("Advanced Server side programming");
newBook.setPrice(50.0);
newBook.setPublisher("Publisher Name");

Response response = target.request(MediaType.APPLICATION_JSON)
    .post(Entity.entity(newBook, MediaType.APPLICATION_JSON));

if (response.getStatus() == 200) {
    System.out.println("Book information submitted successfully!");

    // Retrieve a book by title
    WebTarget getByTitleTarget = target.path("/getByTitle");
    Book retrievedBook =
getByTitleTarget.request(MediaType.APPLICATION_JSON)
    .post(Entity.entity("Advanced Server side
programming", MediaType.APPLICATION_JSON), Book.class);

    System.out.println("Retrieved Book: " + retrievedBook.getTitle());
} else {
    System.out.println("Error submitting book information");
}
}
}

```

3.viewbook.html

```
<!DOCTYPE html>
<html>
<head>
  <title>View Book Information</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>

<h2>Book Information</h2>

<div id="bookInfo">
  <p><strong>ISBN:</strong> <span id="isbn">No book found</span></p>
  <p><strong>Title:</strong> <span id="title">No book found</span></p>
  <p><strong>Price:</strong> <span id="price">No book found</span></p>
  <p><strong>Publisher:</strong> <span id="publisher">No book
found</span></p>
</div>

<script>
  $(document).ready(function() {
    var urlParams = new URLSearchParams(window.location.search);
    var isbn = urlParams.get('isbn');

    $.ajax({
      url: 'http://localhost:8080/21MIC0131_RESTDA2/api/books/getByTitle',
      type: 'POST',
      contentType: 'application/json',
      data: JSON.stringify(isbn),
```

```

        dataType: 'json',
        success: function(book) {
            if (book) {
                $('#isbn').text(book.isbn);
                $('#title').text(book.title);
                $('#price').text(book.price);
                $('#publisher').text(book.publisher);
            }
        },
        error: function(error) {
            alert('Error fetching book information: ' + error.responseText);
        }
    });
});
</script>

</body>
</html>

```

Server Side

1. Book.java

```
package Book;
```

```
import javax.xml.bind.annotation.XmlRootElement;
```

```
@XmlRootElement
```

```
public class Book {
```

```
    private String isbn;
```



```
private String title;  
private double price;  
private String publisher;
```

```
// Default Constructor
```

```
public Book() {  
}
```

```
// Parameterized Constructor
```

```
public Book(String isbn, String title, double price, String publisher) {  
    this.isbn = isbn;  
    this.title = title;  
    this.price = price;  
    this.publisher = publisher;  
}
```

```
// Getters and Setters
```

```
public String getIsbn() {  
    return isbn;  
}
```

```
public void setIsbn(String isbn) {  
    this.isbn = isbn;  
}
```

```
public String getTitle() {  
    return title;  
}
```

```
public void setTitle(String title) {
```

```
    this.title = title;
}
```

```
public double getPrice() {
    return price;
}
```

```
public void setPrice(double price) {
    this.price = price;
}
```

```
public String getPublisher() {
    return publisher;
}
```

```
public void setPublisher(String publisher) {
    this.publisher = publisher;
}
```

```
@Override
public String toString() {
    return "Book{" + "isbn=" + isbn + ", title=" + title + ", price=" + price + ",
publisher=" + publisher + '}';
}
}
```

2.BookResource.java

```
package Book;
```

```
import javax.ws.rs.POST;  
import javax.ws.rs.Path;  
import javax.ws.rs.Consumes;  
import javax.ws.rs.Produces;  
import javax.ws.rs.core.MediaType;
```

```
@Path("/books")
```

```
public class BookResource {
```

```
    private final BookDAO bookDAO = new BookDAO();
```

```
    @POST
```

```
    @Consumes(MediaType.APPLICATION_JSON)
```

```
    public void addBook(Book book) {
```

```
        bookDAO.addBook(book);
```

```
    }
```

```
    @POST
```

```
    @Path("/getByTitle")
```

```
    @Produces(MediaType.APPLICATION_JSON)
```

```
    public Book getBookByTitle(String title) {
```

```
        return bookDAO.getBookByTitle(title);
```

```
    }
```

```
}
```

Database Side

1. BookDAO.java

```
package Book;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class BookDAO {

    private static final String JDBC_URL =
"jdbc:mysql://localhost:3306/21MIC0131_RESTDA2";
    private static final String JDBC_USER = "root";
    private static final String JDBC_PASSWORD = "password"; // Replace with
your MySQL password

    public void addBook(Book book) {
        String query = "INSERT INTO books (isbn, title, price, publisher) VALUES
(?, ?, ?, ?)";

        try (Connection conn = DriverManager.getConnection(JDBC_URL,
JDBC_USER, JDBC_PASSWORD);
            PreparedStatement pstmt = conn.prepareStatement(query)) {

            pstmt.setString(1, book.getIsbn());
            pstmt.setString(2, book.getTitle());
            pstmt.setDouble(3, book.getPrice());
            pstmt.setString(4, book.getPublisher());
```

```

        pstmt.executeUpdate();

    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public Book getBookByTitle(String title) {
    String query = "SELECT * FROM books WHERE title = ?";
    Book book = null;

    try (Connection conn = DriverManager.getConnection(JDBC_URL,
        JDBC_USER, JDBC_PASSWORD);
        PreparedStatement pstmt = conn.prepareStatement(query)) {

        pstmt.setString(1, title);

        ResultSet rs = pstmt.executeQuery();

        if (rs.next()) {
            book = new Book();
            book.setIsbn(rs.getString("isbn"));
            book.setTitle(rs.getString("title"));
            book.setPrice(rs.getDouble("price"));
            book.setPublisher(rs.getString("publisher"));
        }

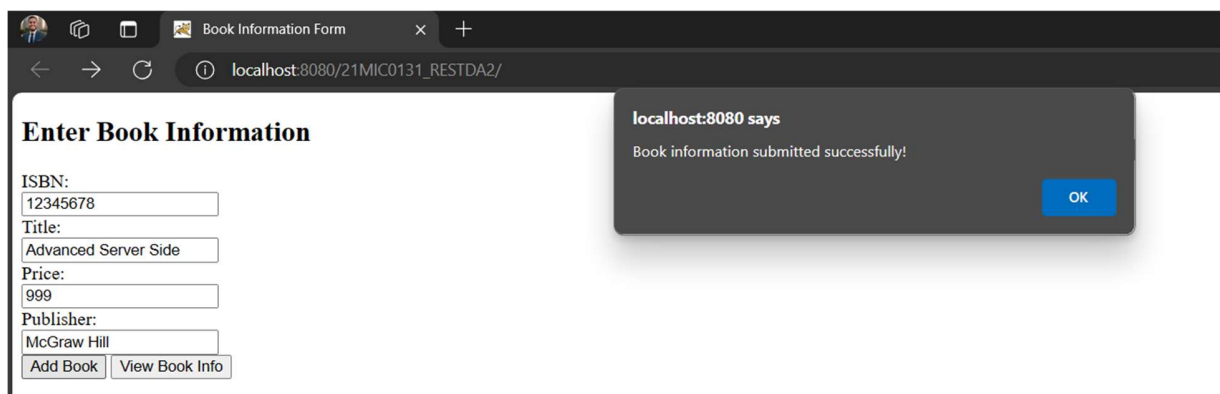
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```
    return book;
}
}
```

Screenshots

-Added book info successfully



- View info of book by entering ISBN number-

