

Informatics Institute of Technology

Department of Computing

BSc (Hons) Computer Science

Test Cases Report

**Module Name : 5COSC022W.2 Client-Server Architectures**

**Student name : Miyuru bashitha dayananda**

**IIT Student ID :   20233150**

**UOW Student ID : W2082283**

# Introduction

This report provides a comprehensive overview of the test cases designed and executed for the RESTful Bookstore API. The primary objective of this report is to ensure that all implemented endpoints behave as expected and meet the functional requirements defined for the system.

The Bookstore API was developed using JAX-RS with JSON as the format for data exchange. To test the API endpoints, Postman was utilized, allowing for both manual and automated test executions.

The structure of this report emphasizes clarity and detail, with all test cases documented in tabular format. Each table presents the endpoint tested, the input data or parameters, the expected outcome, and the actual results observed during execution. This documentation serves both as a quality assurance reference and as a validation of the API's functionality.

# API Endpoint Test Case Tables

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Endpoint  http:// localhost : 8080  /BookStoreAPI  /api | Descri  ption | HTTP Method | Request Body (JSON)) | Expe  cted HTTP Status  Code | Expected Response Body (JSON) | Actual Result (Pass/  Fail) |
|  |  |  | AuthorResource |  |  |  |
| GET /Author  /get  /{ id} | Get author details by id | GET | - | 200 | {          "name": "miyuruedfddf",          "surname": "bashitha",          "age": 23,          "authorID": "author502970"      } | pass |
| GET /Author  /get  /{id} | Get invalid author details | GET | - | 404 | {      "error": "Author not found"  } | pass |
| GET /Author  /get  /{id} | Try to get a author details without any authors in system | GET | - | 404 | "error": "No authors have been added to the database."  } | pass |
| GET  /Authors/getBooks/author172517 | Get all books that belong to a author by auther id | GET | - | 200 | [      {          "title": "java",          "authorID": "author172517",          "bookID": "book803153",          "publicationYear": 2018,          "price": 45.5,          "stockQuantity": 10,          "isbn": "97979797"      },      {          "title": "dart",          "authorID": "author172517",          "bookID": "book934360",          "publicationYear": 2018,          "price": 45.5,          "stockQuantity": 10,          "isbn": "97979797123"      }  ] | pass |
| GET  /Authors  /getBooks  /1234 | Get all books that belong to a author by invalide auther id | GET | - | 404 | {      "error": "Author not found"  } | pass |
| POST  /Author  /add | Add author | POST | {    "name": "miyuru",    "surname": "bashitha",    "age": 23  } | 201 | {      "message": "Author successfully added."  } | pass |
| POST  /Author  /add | Add author  With invalide or empty details | POST | {    "name": "miyuru",      "age": 23  } | 400 | {      "error": "Surname is required."  } | pass |
| POST  /Author  /add | Try to add author without details | POST |  | 400 | {      "error": "Author details are missing."  } | pass |
| POST  /Author  /add | Try to add same author details more than once | POST | {    "name": "miyuru",    "surname": "bashitha",    "age": 23  } | 400 | {      "error": "Author with this name is already registered."  } | pass |
| DELETE/ Author  /delete  /id | Delete author by id | DELETE |  | 200 | {      "message": "Author removed successfully."  } | pass |
| DELETE/ Author  /delete  /id | Delete a non-existing author | DELETE |  | 404 | {      "error": "Author not found."  } | pass |
| PUT  /Author  /id | Update a author details | PUT | {    "name": "kevin",    "surname": "sin",    "age": 50  } | 200 | {      "message": "Author updated successfully."  } | pass |
| PUT  /Author  /update  /id | Update a non-existing author | PUT | "name": "kevin",    "surname": "sin",    "age": 50 | 404 | {      "error": "Author not found."  } | pass |
| GET  /Author  /getAll | Get all author details | GET | - | 200 | [      {          "name": "kevin",          "surname": "yu",          "age": 50      },      {          "name": "miyuru",          "surname": "bashitha",          "age": 23      }  ] | pass |
| GET  /Author  /getAll | Get all authors detilas when there is no ones saved | GET |  | 404 | {      "error": "No authors have been added to the database."  } | pass |
|  |  |  | CustomerResource |  |  |  |
| POST  /customer  /add | Add new customer | POST | {    "name": "miyuru bashitha",    "email": "miyuru@example.com",    "password": "mb1234"  } | 201 | {      "message": "Customer successfully added."  } |  |
| POST  /customer  /add | Add same customer again | POST | {    "name": "miyuru bashitha",    "email": "miyuru@example.com",    "password": "mb1234"  } | 400 | {      "error": "A customer with this email already exists."  } | pass |
| POST  /customer  /add | Add customer  With  Wrong email format | POST | {    "name": "miyuru bashitha",    "email": "invalideemailformat",    "password": "mb1234" | 400 | {      "error": "Email format is invalid."  } | pass |
| POST  /customer  /add | Add customer with invalid  password length | POST | {    "name": "miyuru bashitha",    "email": "miyuru@example.com",    "password": "no"} | 400 | {      "error": "Password must be at least 6 characters long."  } | pass |
| POST  /customer  /add | Add customer without name | POST | {    "email": "miyuru@example.com",    "password": "mb1234"  } | 400 | {      "error": "Name is required."  } | pass |
| GET  /customer  /id | Get customer details by id | GET |  | 200 | {      "name": "miyuru Updated",      "email": "update@example.com",      "password": "new123456",      "customerID": "customer741578",      "cart": {          "items": [],          "customerEmail": "update@example.com",          "valueOfTheCart": 0.0      }  }  ",      " | pass |
| GET  /customer  /id | Try to get customer that not saved in the system | GET |  | 404 | {      "error": "Customer not found with the given name."  } | pass |
| GET  /customer  /getAll | Get All customer | GET |  | 200 | [      {          "name": "miyuru bashitha",          "email": "miyuru@example.com",          "password": "mb1234",          "cart": {              "items": [],              "customerEmail": "miyuru@example.com",              "valueOfTheCart": 0.0          }      },      {          "name": "miyuru c",          "email": "c@example.com",          "password": "cc1234",          "cart": {              "items": [],              "customerEmail": "c@example.com",              "valueOfTheCart": 0.0          }      }  ] | pass |
| DELETE  customer/delete/id | Delete  A customer  Using id | DELETE |  | 200 | {      "message": "Customer removed successfully."  } | pass |
| DELETE  customer/delete/id | Try to delete non-saved  customer | DELETE |  | 404 | {      "error": "Customer not found."  } | pass |
| PUT  / customer  /update  /id | Try to update customer with invalide password  format | PUT | {    "name": "miyuru Updated",    "email": "update@example.com",    "password": "no"  } | 400 | {      "error": "Password must be at least 6 characters long."  } | pass |
| PUT  /customer  /update  /id | Update customer details with all valid details | PUT | {    "name": "miyuru Updated",    "email": "update@example.com",    "password": "new123456"} | 200 | {      "message": "Customer updated successfully."  } | pass |
|  |  |  | BookResource |  |  |  |
| POST  /books  /add | Add books | POST | {    "title": "Effective Java",    "authorID": "31443",    "isbn": "978123456767890",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 10  } | 200 | {      "message": "Book added successfully."  } | pass |
| POST  /books  /add | Add two books with same isbn | POST | {    "title": "Effective Java",    "authorID": "31443",    "isbn": "978123456767890",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 10  } | 400 | {      "error": "A book with this ISBN already exists."  } | pass |
| POST  /books  /add | Add a book with a 0 stock | POST | {    "title": "Effective Java",    "authorID": "31443",    "isbn": "978123456767876901111",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 0  } | 404 | {      "error": "At least 1 book needs to be added to stock."  } | pass |
| POST  /books  /add | Adding book with missing details | POST | {      "authorID": "31443",    "isbn": "978123456767876901111",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 20  } | 404 | {      "error": "Title is required."  } | pass |
| POST  /books  /add | Add book without a saved author | POST | {    "title": "Effective Java",    "authorID": "3144323",    "isbn": "978123456767876901111",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 12  } | 404 | {      "error": "Author not found.Can not add a book without author."  } | pass |
| DELETE  /books/delete/book916576 | Delete unsaved book from system | DELETE |  | 404 | "error": "Book not found." | pass |
| DELETE  /books/delete/ book305065 | Delete a saved book from the system | DELETE |  | 200 | {      "message": "Book deleted successfully"  } | pass |
| DELETE  /books/delete/ book305065 | Try to delete withoud saving any book | DELETE |  | 404 | {      "error": "Book not found."  } | pass |
| PUT  /books/update/book072467 | Update book | PUT | {    "title": "book",    "authorID": "author480255",    "isbn": "978123456767890",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 10  } | 200 | {      "message": "Book updated successfully"  } | pass |
| PUT  /books/update/book072467 | Update book without 0 stock | PUT | {    "title": "book",    "authorID": "author480255",    "isbn": "978123456767890",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 0  } | 400 | {      "error": "At least 1 book needs to be added to stock."  } | pass |
| PUT  PUT  /books/update/book072467 | Update book without saved author | PUT | {    "title": "book",    "authorID": "author480258765",    "isbn": "978123456767890",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 87  } | 404 | {      "error": "Author not found.Can not add a book without author."  } | pass |
| PUT  books/update/book07246 | Try to update a book with not saved book id | PUT | {    "title": "book",    "authorID": "author480255",    "isbn": "97812345676789230",    "publicationYear": 2018,    "price": 45.5,    "stockQuantity": 34  } | 404 | {      "error": "Book not found."  } | pass |
|  |  |  | CartResource |  |  |  |
| GET /carts/getcart/{customerID} | Find cart by customer ID | GET |  | 200 | {      "customerID": "customer274202",      "items": [          {              "title": "48power of rules",              "authorID": "author762064",              "bookID": "book963156",              "publicationYear": 2018,              "price": 45.5,              "stockQuantity": 8,              "isbn": "97979797"          },          {              "title": "48power of rules",              "authorID": "author762064",              "bookID": "book963156",              "publicationYear": 2018,              "price": 45.5,              "stockQuantity": 8,              "isbn": "97979797"          }      ],      "valueOfTheCart": 91.0  } | pass |
| GET  /carts/getcart/customer 438738 | Try to Get non saved customer  Carts details | GET |  | 404 | {      "error": "no books found in the cart"  } | pass |
| GET  /carts/getcart/customer591300 | Try to get empty cart details |  |  |  | {      "customerID": "customer274202",      "items": [],      "valueOfTheCart": 0.0  } |  |
| POST  /carts/addBook/customer438738/book953177 | Add book to the customer cart | POST |  | 200 | {      "message": "Book added to cart."  } | pass |
| POST  /carts/addBook/customer438738/book9531771 | Add non saved book to the customer | POST |  | 404 | {      "error": "Book not found in database."  } | pass |
| POST  /carts/addBook/customer4387381/book9531771 | Add book to non – saved customer | POST |  | 404 | {      "error": "Customer not found!"  } |  |
| POST  /carts/addBook/customer4387381/book9531771 | Try to add out of stock book | POST |  | 409 | {      "error": "Requested book is out of stock."  } | pass |
| DELETE  /carts/removeBook/customer274202/book963156 | Try to delete a book from cart | DELETE |  | 200 | {      "message": "Book has been removed from the cart."  } | pass |
| DELETE  /carts/removeBook/customer274202/book9631562 | Try to delete a book that not saved in the cart | DELETE |  | 404 | {      "error": "Book not found in cart."  } | pass |
| DELETE  /carts/removeBook/customer2742021/book963156 | Try to remove a book from non-excised customer | DELETE |  | 404 | {      "error": "Customer not found!"  } | pass |
|  |  |  | OrderResource |  |  |  |
| POST  /orders/addOrder/customer628099/123 Main Street, Colombo | Add order using customerID | POST |  | 201 | {      "message": "Order added successfully."  } | pass |
| POST  /orders/addOrder/customer628099/123 Main Street, Colombo 1 | Adding a order with invalid customer id | POST |  | 404 | {      "error": "Customer not found."  } | pass |
| POST  /orders/addOrder/customer505992/1234 | Add a order with empty cart | POST |  | 404 | {      "error": "this customer has no books in cart."  } | pass |
| GET  /orders/getOrder/customer791642 | Get order details with customer id | GET |  | 200 | {      "orderId": "order694848",      "customerID": "customer791642",      "cart": {          "customerID": "customer791642",          "items": [              {                  "title": "java",                  "authorID": "author010133",                  "bookID": "book211279",                  "publicationYear": 2018,                  "price": 45.5,                  "stockQuantity": 7,                  "isbn": "97979797"              }          ],          "valueOfTheCart": 45.5      },      "orderDate": null,      "shippingCost": 2.275,      "shippingAddress": "1234",      "totalAmount": 47.775,      "status": "pending"  } | pass |
| GET  /orders/getOrder/customer7916421 | Try to get order details with invalid customer id | GET |  | 404 | {      "error": "Order not found."  } | pass |
| GET  /orders/getOrder/customer7916421 | Get order details without adding any order | GET |  | 404 | {  No orders have been added to the system  } | pass |
| GET  /orders  /getOrder  /customer791642  /order694848 | Get order details with customer id and order id | GET |  | 200 | "orderId": "order694848",      "customerID": "customer791642",      "cart": {          "customerID": "customer791642",          "items": [              {                  "title": "java",                  "authorID": "author010133",                  "bookID": "book211279",                  "publicationYear": 2018,                  "price": 45.5,                  "stockQuantity": 7,                  "isbn": "97979797"              }          ],          "valueOfTheCart": 45.5      },      "orderDate": null,      "shippingCost": 2.275,      "shippingAddress": "1234", | pass |
| GET  /orders  /getOrder  /customer222222  /order111111 | Try to get order details with invalide customer and order id |  |  | 404 | {      "error": "No orders found for this customerID and orderID."  } | pass |
|  |  |  |  |  |  |  |