

## ZSS Challenge Documentation

Developer – Alfred Mukonda

Email - [aamukonda@gmail.com](mailto:aamukonda@gmail.com)

Phone - 0777697075

Dear recruiting executive

The solution to this challenge was **NOT** done to **best practice**. The Developer deliberately developed in that way to respond to the main Objectives in Question. I am available to discuss best programming practices as far as transaction processing is concerned.

### **Business Logic**

1. Add new BOOK
2. Add new Category
3. Search Books By Category
4. Search Books By Title
5. Search Books By Description
6. Purchase Book
  - a. Add your book to CART
  - b. Pay Book

This Solution was developed using MySQL Database , Kindly Add to your Database Connections to

**Applications.Properties** FILE in /demoMysqltutorial/src/main/resources

Kindly Call me for any help in integrating with your Database

## Create books

[http://localhost:8080/api/v1/Book\\_Add](http://localhost:8080/api/v1/Book_Add)

Method	POST
--------	------

NB

Books Belong to valid Categories

Create Category First and assign a **valid Id from Category** to a book

Sample Request	Sample Response
<pre>{   "title": "Pre-School Day 1",   "description": "Pre School Book",   "price": 20.25,   "purchase_status": "Available",   "created_by": "Alfred",   "created_date": "2022-01-17T00:00:00.000+00:00",   "category": 0 }</pre>	<pre>{   "id": 80,   "title": "Pre-School Day 1",   "description": "Pre School Book",   "price": 20.25,   "purchase_status": "Available",   "created_by": "Alfred",   "created_date": "2022-01-17T00:00:00.000+00:00",   "category": 0,   "purchasereference": null,   "updated": null,   "responseCode": null,   "responseDescription": null,   "debitReference": null }</pre>

# Create Category

[http://localhost:8080/api/v1/Category\\_Add](http://localhost:8080/api/v1/Category_Add)

Method	POST
--------	------

Sample Request	Sample Response
<pre>{    "category_code": "Drama001",   "categorytitle": "Drama Queens",   "category_description": "Drama Books for Level 1 Students",   "created_by": "Alfred",   "created_date": "2022-03-08T11:28:39.4+02:00" }</pre>	<pre>{    "id": 82,   "category_code": "Drama001",   "categorytitle": "Drama Queens",   "category_description": "Drama Books for   "created_by": "Alfred",   "created_date": "2022-03-08T09:28:39.400"</pre>

## Search Book By Title

[http://localhost:8080/api/v1/Books\\_Search\\_By\\_Title](http://localhost:8080/api/v1/Books_Search_By_Title)

Method	POST
--------	------

NB – Search phrase can be even 1 letter

Sample Request	Sample Response
<pre>{   "title": "School" }</pre>	<pre>[   {     "id": 83,     "title": "Pre-School Day 1",     "description": "Pre School Book",     "price": 20.25,     "purchase_status": "Available",     "created_by": "Alfred",     "created_date": "2022-01-17T00:00:00.000+00:00",     "category": 82,     "purchasereference": null,     "updated": null,     "responseCode": null,     "responseDescription": null,     "debitReference": null   } ]</pre>

## Search Book By Description

[http://localhost:8080/api/v1/Books\\_Search\\_By\\_Description](http://localhost:8080/api/v1/Books_Search_By_Description)

Method	POST
--------	------

NB – Search phrase can be even 1 letter

Sample Request	Sample Response
<pre>{   "description": "School" }</pre>	<pre>[   {     "id": 83,     "title": "Pre-School Day 1",     "description": "Pre School Book",     "price": 20.25,     "purchase_status": "Available",     "created_by": "Alfred",     "created_date": "2022-01-17T00:00:00.000+00:00",     "category": 82,     "purchasereference": null,     "updated": null,     "responseCode": null,     "responseDescription": null,     "debitReference": null   } ]</pre>

## Retrieve Books in a certain category

[http://localhost:8080/api/v1/Books\\_By\\_Category/82](http://localhost:8080/api/v1/Books_By_Category/82)

Method	GET
--------	-----

NB – Use path variable id (Pass Id from Category Class Model)

Sample Request	Sample Response
	<pre>[   {     "id": 83,     "title": "Pre-School Day 1",     "description": "Pre School Book",     "price": 20.25,     "purchase_status": "Available",     "created_by": "Alfred",     "created_date": "2022-01-17T00:00:00.000+00:00",     "category": 82,     "purchasereference": null,     "updated": null,     "responseCode": null,     "responseDescription": null,     "debitReference": null   } ]</pre>

## Purchase Book

### Step 1 - Add Book to CART

[http://localhost:8080/api/v1/Purchase\\_Book\\_Step01/49](http://localhost:8080/api/v1/Purchase_Book_Step01/49)

Method	PUT
--------	-----

NB – Use path variable id (Pass Id from Books)

Only Existing books are sold.

The Book is put in a Pending state and a Payment

Sample Request	Sample Response
	<pre>{   "id": 84,   "title": "The King Of Harare",   "description": "Harare based drama",   "price": 20.25,   "purchase_status": "Pending Payment",   "created_by": "Alfred",   "created_date": "2022-01-17T00:00:00.000+00:00",   "category": 82,   "purchasereference": "96209c9b-fa5a-4594-94ec-469a033986bb",   "updated": null,   "responseCode": null,   "responseDescription": null,   "debitReference": null }</pre>

## Step 2 - Commit Payment

### Integration with ZSS Payment API

<http://localhost:8080/api/v1/PurchaseBook>

Method	PUT
--------	-----

NB – Consume ZSS Purchase API

## Add the Purchase Reference Created In Step 1 above to the API reference

This will match the Payment to the Book

Sample Request	Sample Response
<pre>{   "type" : "PURCHASE",   "extendedType" : "NONE",   "amount" : 123.33,   "created" : "2022-02-14T11:28:39.4+02:00",   "card" : {     "id" : "1234560000000001",     "expiry" : "2020-01-01"   },   "reference" : "96209c9b-fa5a-4594-94ec-469a033986bb",   "narration" : "Test Narration",   "additionalData" : {     "SampleKey" : "This is a sample value"   } }</pre>	<pre>{   "id": 49,   "title": "Little Star",   "description": "Books for children",   "price": "20.25",   "purchase_status": "Booked - Pending Payment",   "created_by": "Alfred",   "created_date": "2022-01-17T00:00:00.000+00:00",   "category": 0,   "purchasereference": "33d0893a-4c0e-49cf-a373-7e67a64ce456",   "updated": "2022-03-08T09:35:48.000+00:00",   "responseCode": "000",   "responseDescription": "Approved",   "debitReference": "d96ec0e4-27c6-4d25-bdd0-343c0f2bfc90" }</pre>