**URLs to access Services**

1. **Authentication**

URL - localhost:9999/login/

Method : POST

Request :

Body - {

"username":"adminone",

"password":"adminone"

}

Headers – Content-Type = application/json

Response :

Authorization Token used by all services

The token for admin can only be used for admin operations –

Ex: "username":"adminone",

"password":"adminone"

The token for customer can only be used for customer operations –

Ex: "username":"userone",

"password":" userone”

The token for delivery person can only be used for delivery person operations –

Ex: "username":"deliveryone",

"password":" deliveryone "

1. **Admin**
2. Add Item:

This method creates a file in database/inventory folder in project with file name datafile

URL - localhost:9999/admin/add

Method : POST

Request:

Body- {

"itemId":1,

"itemName":"Saree",

"itemPrice":10

}

Headers – Content-Type = application/json

Authorization Token

Response:

String – telling whether operation is successful or not

1. Delete Item:

URL - localhost:9999/admin/delete

Method : DELETE

Request:

Body- {

"itemId":1,

"itemName":"Saree",

"itemPrice":10

}

Headers – Content-Type = application/json

Authorization Token

Response:

String – telling whether operation is successful or not

1. **Customer**
2. Search :

URL - localhost:9999/customer/search

Method : GET

Request:

Body-

Saree (Name of the item to be searched)

Headers – Content-Type = text/plain

Authorization Token

Response:

Set of items

1. Add item to Cart :

This method creates a file in database/cart folder in project with file name as the name of the customer provided in request

URL - localhost:9999/customer/addtocart

Method : POST

Request:

Body- {

"itemId":1,

"itemName":"Saree",

"itemPrice":10

}

Headers – Content-Type = application/json

Authorization Token

Params: customerName = customerone

Note – here customername is the customer for whom you want to

add the item to the cart

Response:

String – telling whether operation is successful or not

1. Remove item from Cart :

URL - localhost:9999/customer/removefromcart

Method : DELETE

Request:

Body- {

"itemId":1,

"itemName":"Saree",

"itemPrice":10

}

Headers – Content-Type = application/json

Authorization Token

Params: customerName = customerone

Note – here customername is the customer for whom you want to

delete the item from the cart

Response:

String – telling whether operation is successful or not

1. Checkout Cart and Place Order:

This method creates a file in database/order folder in project with file name as the name of the customer provided in request

This method also clears cart after successful payment and order

URL - localhost:9999/customer/checkout

Method : GET

Request:

Authorization Token

Params:

customerName = customerone , paymentStatus = “Successful” or

“Unsuccessful”

Note – here customername is the customer for whom you want to

checkout the cart and for now sending payment status in request till

UI is not created. Method is set as GET because only String is

obtained from this method. Internally it calls POST method of order

Service

Response:

String – telling whether operation is successful or not

1. **Delivery**
2. Set status:

URL - localhost:9999/delivery/status

Method : PUT

Request:

Authorization Token

Params: itemId = 1, itemName = a, itemPrice = 10, customerName

= customerone, deliveryStatus = Delivered or Undelivered

Note – here customername is the customer for whom you want

to set status

Response:

String – telling whether operation is successful or not

1. View Order

URL - localhost:9999/delivery/vieworders

Method : GET

Request:

Authorization Token

Params: customerName = customerone

Note – here customername is the customer whose detail you

want to see.

Response:

Set of item with order status for given customer