ORDER INVENTORY

To be developed

Ecommerce order tracking allows you to monitor all your online orders and shipments and communicate the order status to your customers at any point in time. Ecommerce order tracking includes features like shipment tracking, estimated delivery dates, and frequent updates on the order status. Together, these things play a key role in the customer ­­­­­­­­­­­experience, letting buyers know what to expect for their delivery and being notified of any delays.

Order tracking automates the process without creating any extra work for you, putting less burden on your customer support team without compromising quality. By saving time and money, you have more resources to invest in other methods of improving the customer experience.

Shoppers have obviously become accustomed to having a high level of visibility into their orders. Online shopping basically demands it with such a competitive ecommerce landscape. Having an ecommerce order tracking system in place is mandatory if your brand doesn’t want to lose business to Amazon and other major retailers.

1. All Rest API request and response format must be in JSON.
2. 200, 201, 400, 404, 401, 500, 505 – use appropriate http status code in your response object.
3. Handle the Rest exceptions globally using Rest exception handler for all endpoints.
4. Sample Error Responses are shown in below table, participants to identify error paths for all the Endpoints.
5. Test your Rest Api using Junit and Mockito.
6. Participants should not alter database schema under any circumstances except wherever changes are mentioned explicitly.
7. All GET and POST requests for customer are secured

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
| /api/v1/customers \*done | GET | Fetch all customers | Collection of Customers |  |
| /api/v1/customers \*done | POST | Add new Customer object in DB | String: Record Created Successfully |  |
| /api/v1/customers \*done | PUT | Update customer by object | String: Record Updated Successfully |  |
| /api/v1/customers | DELETE | Delete customer by object | String: Record deleted Successfully |  |
| /api/v1/customers/{email} \*done | GET | Search Customers Matching emailid | Collection of Customers |  |
| /api/v1/customers/{name} \*done | GET | Search Customers Matching name wildcard | Collection of Customers |  |
| /api/v1/customers/shipment/status  \* done | GET | Shipment status wise count of customers | Custom Object of Shipment Status and count of customers |  |
| /api/v1/customers/{custid}/order | GET | Fetch orders for the respective customer by id | Custom object of customer with list of Order(s) |  |
| /api/v1/customers/{custid}/shipment | GET | Fetch shipment history for the respective customer by id | Custom object of customer with list of shipment(s) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
| /api/v1/products | GET | Fetch all products | Collection of products |  |
| /api/v1/products | POST | Add new product object in DB | String: Record Created Successfully |  |
| /api/v1/products | PUT | Update products by object | String: Record Updated Successfully |  |
| /api/v1/products /{id} | DELETE | Delete product | String: Record deleted Successfully |  |
| /api/v1/products /{productname} | GET | Search Products Matching product name | Collection of Products |  |
| /api/v1/products /unitprice?min=value&max=value | GET | Fetch List of Products matching between min and max unit price | Collection of Products |  |
| /api/v1/products/sort?field=value | GET | Get list of products order by query field | Collection of Product Object ordered by query field |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
| /api/v1/orders | GET | Fetch all orders | Collection of Orders |  |
| /api/v1/orders | POST | Create new Order object in DB | String: Record Created Successfully |  |
| /api/v1/orders | PUT | Update Order by object | String: Record Updated Successfully |  |
| /api/v1/orders/{id} | DELETE | Delete Order | String: Record deleted Successfully |  |
| /api/v1/orders/status | GET | Count of orders by status | Custom Object for status of order and count |  |
| /api/v1/orders/{store} | GET | Fetch order details matching storename | Object of Order having orderid, orderstatus, storename and webaddress |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
| /api/v1/inventory | GET | Fetch all inventory records | Custom Object of Inventory records |  |
| /api/v1/inventory?storeid=value | GET | Fetch productdata, storedata, orderstatus matching storied | Custom Object Of inventory contains product, store and order details |  |
| /api/v1/inventory/shipment | GET | Fetch inventories and matching shipments | Collection of Inventory records matching shipments |  |
| /api/v1/inventory/{orderid} | GET | Fetch store, product and customer data matching order id | Custom Object to return product, customer and store details matching ordered |  |
| /api/v1/inventory/shipment | GET | Count shipment status wise count of total products sold | Custom Object of Inventory to print shipment status and count of sold products |  |
| /api/v1/inventory/{orderid}/details | GET | Fetch list of Products in a order matching ordered with store details, shipment status and total amount | Custom List of object for inventory details |  |