DigishopeX E-Commerce REST API

Overview

This application provides a comprehensive set of RESTful web services that can handle various operations of an E-commerce website. There are three types of users - Admin, Customer, and Guest-user, each with their own set of privileges and functionalities. Admins have the highest level of access and can perform a wide range of operations, such as registering as an admin, creating product categories, adding products to categories, manipulating a list of products, registering suppliers and shippers, and changing their active status, viewing and tracking orders, obtaining sales analysis, and viewing bestselling products. These features give admins complete control over the E-commerce website, allowing them to manage products, suppliers, and customers efficiently. Customers can register themselves on the platform and view available categories and products, sort and filter products according to their preferences, add, update, and delete products from their cart, place orders, cancel orders, and request returns. Customers can also track their order status, rate products, and access other functionalities that are specific to them. Guest-users, who have not registered on the platform, also have access to some limited features such as viewing available products and sorting and filtering products. The application is secured with Spring Security, which provides robust authentication and authorization mechanisms to ensure that users are only able to access the functionalities that they are authorized to use. CORS feature is also incorporated so that the web services can be accessed through any type of client, making it easier for users to interact with the application. Overall, this RESTful web service provides extensive functionalities to perform various CRUD operations required in an E-commerce application.

Security

Type	Name	Scopes
apiKey	basicAuth	
apiKey	bearerToken	

Paths

PUT `/customer/update` Update customer details

customer can update his profile details - name, mobile number, password

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/customer/products/editratings/{pid}/{ratings}` Edit ratings for a product

Customer can edit ratings of a product by provding a valid productid. Customer can edit a product if and only if he has rated the product earlier. Ratings should be in the range of 0 - 5, both inclusive

Parameters

Type	Name	Description	Schema
path	**ratings** *required*		integer (int32)
path	**pid** *required*	pid represents productid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/customer/cart/increasequantity/{pid}` Increase quantity of a product in the cart

Customer can increase quantity of a product by 1 which already exists in the customer cart by providing a valid productid. As the product quantity will increase in the cart, quantity of the product in the stock will decrease

Parameters

Type	Name	Description	Schema
path	**pid** *required*	pid represents productid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/customer/cart/decreasequantity/{pid}` Decrease quantity of a product in the cart

Customer can decrease quantity of a product by 1 which already exists in the customer cart by providing a valid productid. As the product quantity will decrease in the cart, quantity of the product in the stock will increase

Parameters

Type	Name	Description	Schema
path	**pid** *required*	pid represents productid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/admin/update` Update admin details

Admin can update his profile details - name, mobile number, password

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/admin/products/update` Update a product

Admin can a update product details by providing valid productid

Responses

Code	Description	Links
200	OK *Content* /	No Links

PUT `/admin/categories/update` Update an existing category details

Admin can upadate an existing categorydetails

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/customer/products/rate/{pid}/{ratings}` Rate a product

Customer can rate a product by providing valid productid. Ratings should be in the range of 0 - 5, both inclusive

Parameters

Type	Name	Description	Schema
path	**ratings** *required*		integer (int32)
path	**pid** *required*	pid represents productid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/customer/orders/purchase` Place an order

Customer can place an order by making a payment. It returns a ResponseEntity containing an OrderDTO object, which represents the details of the order. It contains the following fields: orderID, customerID, details: a list of OrderDetail objects that represent the details of the products ordered. total_order_amount: a double that represents the total cost of the order. orderDateTime: a LocalDateTime object that represents the date and time when the order is placed. shipDateTime: a LocalDateTime object that represents the date and time when the order will be shipped. Order will be shipped after one day, time is auto-generated between 10.00 a.m. - 9.00 p.m. measured in 24 hours clock. deliveryDateTime: a LocalDateTime object that represents the date and time when the order will be delivered. Order will be delivered after 5 days, time is auto-generated between 10.00

a.m. - 9.00 p.m. measured in 24 hours clock. shipper: Shipper will be assigned dynamically based on the active status of the shipper. suppliers: Supplier for each product of different categories will be assigned dynamically based on the active status and the category of products they do supply. The OrderDetail object is a child object of OrderDTO and contains the following fields: orderDetailID, quantity: an integer that represents the quantity of the product ordered and some other details. Also the order status will be updated to 'PENDING'

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/customer/create` Register as a customer

An user can register himself as a customer

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/customer/cart/addproduct` Add product in the cart

Customer can add a product to his cart by providing a valid productid. As the product will be added in cart, quantity of the product in the stock will decrease

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/admin/suppliers/register` Register a supplier

Admin can register a supplier by providing all necessary supplier details. Each supplier supplies the products of a particular category

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/admin/shippers/register` Register a shipper

Admin can register a shipper by providing all necessary shipper details

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/admin/products/create` Create a product

Admin can create a product under a particular category

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/admin/products/add` Increase product quantity

Admin can increase product quantity in stock by providing valid productid and quantity

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/admin/create` Register as an admin

An user can register himself as an admin

Responses

Code	Description	Links
200	OK *Content* /	No Links

POST `/admin/categories/create` Create a product category

Admin can create a non-existing product category

Responses

Code	Description	Links
200	OK *Content* /	No Links

PATCH `/admin/suppliers/changestatus/{sid}` Change active status of a supplier

Admin can change the active status of a supplier

Parameters

Type	Name	Description	Schema
path	**sid** *required*	sid represents supplierid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

PATCH `/admin/shippers/changestatus/{sid}` Change active status of a shipper

Admin can change the active status of a shipper

Parameters

Type	Name	Description	Schema
path	**sid** *required*	sid represents shipperid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/sortbyratings/desc/{catid}` Sort products in descending order of product ratings for a category

Customer can sort all products in descending order of product ratings for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/sortbyratings/asc/{catid}` Sort products in ascending order of product ratings for a category

Customer can sort all products in ascending order of product ratings for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/sortbyprice/desc/{catid}` Sort products in descending order of product price for a category

Customer can sort all products in descending order of product price for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/sortbyprice/asc/{catid}` Sort products in ascending order of product price for a category

Customer can sort all products in ascending order of product price for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/sortbyname/desc/{catid}` Sort products in descending order of product name for a category

Customer can sort all products in descending order of product name for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/sortbyname/asc/{catid}` Sort products in ascending order of product name for a category

Customer can sort all products in ascending order of product name for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/getbycategoryname/{name}` Get all products by category name

Customer can retrive all products of a category by providing valid category name

Parameters

Type	Name	Description	Schema
path	**name** *required*	name represents category name	string

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/getbycategoryid/{cid}` Get all products by categoryId

Customer can retrive all products of a category by providing valid categoryId

Parameters

Type	Name	Description	Schema
path	**cid** *required*	cid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/getall` Get all products for all categories

Customer can retrive all products for all categories

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/get/{pid}` Get a product by productId

Customer can retrive a product by providing valid productId

Parameters

Type	Name	Description	Schema
path	**pid** *required*	pid represents productid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/filterbyratings/{cid}/{min}/{max}` Filter products by a rating range for a category

Customer can filter products by a rating range for a category by providing valid categoryid, lower bound of price range and upper bound of price range

Parameters

Type	Name	Description	Schema
path	**min** *required*	min represents lower bound of ratings range	integer (int32)

Type	Name	Description	Schema
path	**max** *required*	max represents upper bound of ratings range	integer (int32)
path	**cid** *required*	cid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/products/filterbyprice/{cid}/{min}/{max}` Filter products by a price range for a category

Customer can filter products by a price range for a category by providing valid categoryid, lower bound of price range and upper bound of price range

Parameters

Type	Name	Description	Schema
path	**min** *required*	min represents lower bound of price range	number (double)
path	**max** *required*	max represents upper bound of price range	number (double)
path	**cid** *required*	cid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/orders/track/{oid}` Track order

Customer can check order status by providing correct orderid

Parameters

Type	Name	Description	Schema
path	**oid** *required*	oid represents orderid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/orders/getall` Get all orders

Customer can get all the available orders in the database

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/orders/get-by-orderid/{oid}` Get order by orderid

Customer can get an order by providing a correct orderid

Parameters

Type	Name	Description	Schema
path	**oid** *required*	oid represents orderid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/categories/getall` Get all existing categories

Customer can get all the existing categories

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/customer/cart/getallproducts/{email}` Get all products from the cart

Customer can get all the products from the cart by providing his email id

Parameters

Type	Name	Description	Schema
path	**email** *required*		string

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/authentication/login` Admin or Customer login

An admin or customer can login by providing valid credentials. Server will send a JWT to the client in the response header which can be used for subsequent api calls

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/sortbyratings/desc/{catid}` Sort products in descending order of product ratings for a category

Guest user can sort all products in descending order of product ratings for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/sortbyratings/asc/{catid}` Sort products in ascending order of product ratings for a category

Guest user can sort all products in ascending order of product ratings for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/sortbyprice/desc/{catid}` Sort products in descending order of product price for a category

Guest user can sort all products in descending order of product price for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/sortbyprice/asc/{catid}` Sort products in ascending order of product price for a category

Guest user can sort all products in ascending order of product price for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/sortbyname/desc/{catid}` Sort products in descending order of product name for a category

Guest user can sort all products in descending order of product name for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/sortbyname/asc/{catid}` Sort products in ascending order of product name for a category

Guest user can sort all products in ascending order of product name for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/getbycategoryname/{name}` Get all products by category name

Guest user can retrive all products of a category by providing valid category name

Parameters

Type	Name	Description	Schema
path	**name** *required*	name represents category name	string

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/getbycategoryid/{catid}` Get all products by categoryId

Guest user can retrive all products of a category by providing valid categoryId

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/getall` Get all products for all categories

Guest user can retrive all products for all categories

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/get/{pid}` Get a product by productId

Guest user can retrive a product by providing valid productId

Parameters

Type	Name	Description	Schema
path	**pid** *required*	pid represents productid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/filterbyratings/{cid}/{min}/{max}` Filter products by a rating range for a category

Guest user can filter products by a rating range for a category by providing valid categoryid, lower bound of price range and upper bound of price range

Parameters

Type	Name	Description	Schema
path	**min** *required*	min represents lower bound of ratings range	integer (int32)
path	**max** *required*	max represents upper bound of ratings range	integer (int32)
path	**cid** *required*	cid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/products/filterbyprice/{cid}/{min}/{max}` Filter products by a price range for a category

Guest user can filter products by a price range for a category by providing valid categoryid, lower bound of price range and upper bound of price range

Parameters

Type	Name	Description	Schema
path	**min** *required*	min represents lower bound of price range	number (double)
path	**max** *required*	max represents upper bound of price range	number (double)
path	**cid** *required*	cid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/anonymous/categories/getall` Get all existing categories

Guest user can get all the existing categories

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/sortbyratings/desc/{catid}` Sort products in descending order of product ratings for a category

Admin can sort all products in descending order of product ratings for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/sortbyratings/asc/{catid}` Sort products in ascending order of product ratings for a category

Admin can sort all products in ascending order of product ratings for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/sortbyprice/desc/{catid}` Sort products in descending order of product price for a category

Admin can sort all products in descending order of product price for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/sortbyprice/asc/{catid}` Sort products in ascending order of product price for a category

Admin can sort all products in ascending order of product price for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/sortbyname/desc/{catid}` Sort products in descending order of product name for a category

Admin can sort all products in descending order of product name for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/sortbyname/asc/{catid}` Sort products in ascending order of product name for a category

Admin can sort all products in ascending order of product name for a category by providing valid categoryid

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/getbycategoryname/{name}` Get all products by category name

Admin can retrive all products of a category by providing valid category name

Parameters

Type	Name	Description	Schema
path	**name** *required*	name represents categoryName	string

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/getbycategoryid/{catid}` Get all products by categoryId

Admin can retrive all products of a category by providing valid categoryId

Parameters

Type	Name	Description	Schema
path	**catid** *required*	catid represents categoryId	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/getall` Get all products for all categories

Admin can retrive all products for all categories

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/get/{pid}` Get a product by productId

Admin can retrive a product by providing valid productId

Parameters

Type	Name	Description	Schema
path	**pid** *required*	pid represents productId	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/filterbyratings/{cid}/{min}/{max}` Filter products by a rating range for a category

Admin can filter products by a rating range for a category by providing valid categoryid, lower bound of price range and upper bound of price range

Parameters

Type	Name	Description	Schema
path	**min** *required*	min represents lower bound of ratings range, it should be >= 0	integer (int32)
path	**max** *required*	max represents upper bound of ratings range, it should be ⇐ 5	integer (int32)
path	**cid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/products/filterbyprice/{cid}/{min}/{max}` Filter products by a price range for a category

Admin can filter products by a price range for a category by providing valid categoryid, lower bound of price range and upper bound of price range

Parameters

Type	Name	Description	Schema
path	**min** *required*	min represents lower bound of price range, It should be >= 1	number (double)
path	**max** *required*	max represents upper bound of price range	number (double)
path	**cid** *required*	catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/orders/track/{oid}` Track order

Admin can check order status by providing correct orderid

Parameters

Type	Name	Description	Schema
path	**oid** *required*	oid represents orderid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/orders/getall` Get all orders

Admin can get all the available orders in the database

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/orders/get-by-orderid/{oid}` Get order by orderid

Admin can get an order by providing a correct orderid

Parameters

Type	Name	Description	Schema
path	**oid** *required*	oid represents orderid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/orders/get-by-email` Get all orders of a customer by customer email

Admin can get all the orders of a customer by provding a correct cutsomer email

Parameters

Type	Name	Description	Schema
query	**email** *required*		string

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/orders/get-by-customerid/{cid}` Get all orders of a customer by customerid

Admin can get all the orders of a customer by provding a correct cutsomerid

Parameters

Type	Name	Description	Schema
path	**cid** *required*	cid represents customerid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/dashboard/analysis/year` Get sales analysis of an year

With this API operation admin can get SalesAnalysisDTO object that contains information about the sales analysis of the given year. The SalesAnalysisDTO object has several properties: "TotalSalesAmount" - the total sales amount of all orders in the given year "noOfOrders" - the total number of orders placed in the given year "avgOrderValue" - the average value of orders in the given year "totalRevenue" - the total revenue generated from sales in the given year "topSellingProducts" - a list of ProductDTOV3 objects that represent the top selling products in the given year "revenueBreakdown" - a list of RevenueDTO objects that represent the revenue breakdown by category in the given year "conversionRate" - the conversion rate of the website in the given year. The ProductDTOV3 object has several properties: "productID" - the ID of the product "noOfProductsSold" - the number of products sold in the given year "totalSales" - the total sales of the product in the given year "categoryName" - the category name of the product. The RevenueDTO object has several properties: "categoryName" - the category name of the revenue breakdown "categorywiseTotalRevenue" - the total revenue generated from sales in the given year for that category "revenuePercentage" - the percentage of revenue generated from sales in the given year for that category compared to the total revenue generated from sales in the given year.

Parameters

Type	Name	Description	Schema
query	**yr** *required*	yr represents target year. This year should be a past year or the present one	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/dashboard/analysis/today` Get todays' sales analysis

With this API operation admin can get the sales analysis for the current day which returns a DaywiseSalesData object. The DaywiseSalesData class represents a daily sales analysis, which provides important information about the sales of a particular day. This information includes: serialNo: An auto-generated unique identifier for the sales analysis entry. totalSalesAmount: The total sales amount for the day. noOfOrders: The number of orders placed on that day. avgOrderValue: The average order value for the day. totalRevenue: The total revenue generated for the day. conversionRate: The conversion rate for the day. date: The date for which the sales analysis is being performed. topSellingProducts: A list of the top-selling products on that day, represented by the ProductDTOV3 class. revenueBreakdown: A list of the revenue breakdown by category on that day, represented by the RevenueDTO class. The ProductDTOV3 class represents the details of a product that has been sold on a particular day. It includes: serialNo: An auto-generated unique identifier for the product entry. productID: The unique identifier of the product. noOfProductsSold: The number of units sold for the product. totalSales: The total sales generated for the product. categoryName: The category name of the product. The RevenueDTO class represents the revenue breakdown by category on a particular day. It includes: serialNo: An auto-generated unique identifier for the revenue entry. categoryName: The category name for which the revenue is being calculated. categorywiseTotalRevenue: The total revenue generated for the category. revenuePercentage: The percentage of revenue generated by the category out of the total revenue for the day.

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/dashboard/analysis/lastweek` Get sales analysis of the last week

With this API operation admin can get SalesAnalysisDTO object that contains information about the sales analysis of the last week. The SalesAnalysisDTO object has several properties: "TotalSalesAmount" - the total sales amount of all orders in the last week "noOfOrders" - the total number of orders placed in the last week "avgOrderValue" - the average value of orders in the last week "totalRevenue" - the total revenue generated from sales in the last week "topSellingProducts" - a list of ProductDTOV3 objects that represent the top selling products in the last week "revenueBreakdown" - a list of RevenueDTO objects that represent the revenue breakdown by category in the last week "conversionRate" - the conversion rate of the website in the last week The ProductDTOV3 object has several properties: "productID" - the ID of the product "noOfProductsSold" - the number of products sold in the last week "totalSales" - the total sales of the product in the last week "categoryName" - the category name of the product The RevenueDTO object has several properties: "categoryName" - the category name of the revenue breakdown "categorywiseTotalRevenue" - the total revenue generated from sales in the last week for that category "revenuePercentage" - the percentage of revenue generated from sales in the last week for that category compared to the total revenue generated from sales in the last week.

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/dashboard/analysis/lastmonth` Get sales analysis of the last month

With this API operation admin can get SalesAnalysisDTO object that contains information about the sales analysis of the last month. The SalesAnalysisDTO object has several properties: "TotalSalesAmount" - the total sales amount of all orders in the last month "noOfOrders" - the total number of orders placed in the last month "avgOrderValue" - the average value of orders in the last month "totalRevenue" - the total revenue generated from sales in the last month "topSellingProducts" - a list of ProductDTOV3 objects that represent the top selling products in the last month "revenueBreakdown" - a list of RevenueDTO objects that represent the revenue breakdown by category in the last month "conversionRate" - the conversion rate of the website in the last month. The ProductDTOV3 object has several properties: "productID" - the ID of the product "noOfProductsSold" - the number of products sold in the last month "totalSales" - the total sales of the product in the last month "categoryName" - the category name of the product. The RevenueDTO object has several properties: "categoryName" - the category name of the revenue breakdown "categorywiseTotalRevenue" - the total revenue generated from sales in the last month for that category "revenuePercentage" - the percentage of revenue generated from sales in the last month for that category compared to the total revenue generated from sales in the last month.

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/dashboard/analysis/date` Get sales analysis of a particular date

With this API operation admin can get the sales analysis for the provided date which returns a DaywiseSalesData object. The DaywiseSalesData class represents a daily sales analysis, which provides important information about the sales of a particular day. This information includes: serialNo: An auto-generated unique identifier for the sales analysis entry. totalSalesAmount: The total sales amount for the day. noOfOrders: The number of orders placed on that day. avgOrderValue: The average order value for the day. totalRevenue: The total revenue generated for the day. conversionRate: The conversion rate for the day. date: The date for which the sales analysis is being performed. topSellingProducts: A list of the top-selling products on that day, represented by the ProductDTOV3 class. revenueBreakdown: A list of the revenue breakdown by category on that day, represented by the RevenueDTO class. The ProductDTOV3 class represents the details of a product that has been sold on a particular day. It includes: serialNo: An auto-generated unique identifier for the product entry. productID: The unique identifier of the product. noOfProductsSold: The number of units sold for the product. totalSales: The total sales generated for the product. categoryName: The category name of the product. The RevenueDTO class represents the revenue breakdown by category on a particular day. It includes: serialNo: An auto-generated unique identifier for the revenue entry. categoryName: The category name for which the revenue is being calculated. categorywiseTotalRevenue: The total revenue generated for the category. revenuePercentage: The percentage of revenue generated by the category out of the total revenue for the day.

Parameters

Type	Name	Description	Schema
query	**dt** *required*	dt represents the Target Date	string (date)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/dashboard/analysis/bestselling-products-by-ratings` Get bestselling products by ratings in a certain duration

With this API endpoint admin can retrieve a list of bestselling products based on their ratings within a certain time period. The endpoint accepts two parameters, syr and eyr, which represent the start and end dates of the duration to be analyzed(eyr > syr). The response will contain a list of ProductDTOV4 objects, each of which represents a bestselling product during the specified duration. The ProductDTOV4 class has the following fields: productID: an integer representing the unique identifier of the product. noOfProductsSold: an integer representing the total number of units of the product sold during the specified duration. totalSales: a double representing the total sales generated by the product during the specified duration. categoryName: a string representing the name of the category to which the product belongs. avgRatings: a double representing the average rating received by the product during the specified duration.

Parameters

Type	Name	Description	Schema
query	**syr** *required*	syr represents start date	string (date)
query	**eyr** *required*	eyr represents end date	string (date)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/dashboard/analysis/bestselling-products-by-category` Get bestselling products for each category in a certain duration

With this API endpoint admin can retrieve the bestselling products for each category within a specified duration. It accepts two parameters, "syr" representing the start date and "eyr" representing the end date, both of which are required. ProductDTOV5 fields: categoryID: an integer representing the ID of the product’s category. categoryName: a string representing the name of the product’s category. productID: an integer representing the ID of the product. productName: a string representing the name of the product. salePrice: a double representing the selling price of the product. costPrice: a double representing the cost price of the product. noOfProducts: an integer representing the number of products sold in the specified duration. revenueGenerated: a double representing the revenue generated by the product within the specified duration.

Parameters

Type	Name	Description	Schema
query	**syr** *required*	syr represents start date	string (date)
query	**eyr** *required*	eyr represents end date	string (date)

Responses

Code	Description	Links
200	OK *Content* /	No Links

GET `/admin/categories/getall` Get all existing categories

Admin can get all the existing categories

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/customer/orders/return/{oid}` Submit a return request

With this API endpoint a customer can submit a return request for a particular order. The method first checks if the customer has made any orders or not and checks if the order status allows for a return request to be made. If the order has been delivered within the last 7 days, the method creates a ReturnRequestedOrders object and sets the fulfillment(After 3 days of return request submitted) and refund dates(After 10 days of return request submitted, time is dynamically generated between between 10.00 a.m - 9.00 p.m. in a 24 hours clock) accordingly. It also saves the returned order details and the ReturnRequestedOrders object in the database. Finally, it updates the order status to "RETURN_IN_PROGRESS". When the refund will be completed then the order status will be "RETURNED_AND_REFUND_IN_PROGESS". The ReturnRequestDTO is returned with the details of the return request, including the order ID, requested date, fulfillment date, refund date, and status. On the fulfilment date all the products will be picked from the customer and stock will be updated with the products(Stock gets updated every 5 minutes). ReturnRequestDTO contains the following fields: orderID: the ID of the order for which the return request is being made requestedON: the date and time when the return request was made fulfillON: the date and time by which the return request should be fulfilled refundON: the date and time by which the refund should be processed status: the status of the return request

Parameters

Type	Name	Description	Schema
path	**oid** *required*	oid represents orderid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/customer/orders/cancel/{oid}` Cancel an order

Customer can cancel an order by providng a valid orderid. Cancelling an order is only possible if the product is not shipped yet. After cancelling the product customer customer will get a response in which a refund datetime will be present. Refund date will be 7 days later of the cancelled date and refund time is a randomly generated time between 10.00 a.m. - 9.00 p.m. mesaured in 24 hours clock. Also the orderStatus will be updated to 'CANCELLED_AND_REFUND_IN_PROGESS'. After the order is being cancelled the products get updated to the stock immediately

Parameters

Type	Name	Description	Schema
path	**oid** *required*	oid represents orderid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/customer/delete/{email}` Delete customer details

Customer can delete his all details. This customer has no longer has any authority to perform any action in the application

Parameters

Type	Name	Description	Schema
path	**email** *required*		string

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/customer/cart/deleteproduct/{pid}` Delete a product from the cart

Customer can delete a product from the cart by providing a valid productid. As the product is deleted form the cart, the product quantity will increase in the stock

Parameters

Type	Name	Description	Schema
path	**pid** *required*	pid represents productid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/authentication/logout` Admin or Customer logout

An admin or customer can logout by providing a valid JWT Once the admin or customer is loggedout the JWT will be blacklisted and can not be used for any api call

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/admin/products/delete/{pid}` Delete a product

Admin can delete a product by providing valid productid

Parameters

Type	Name	Description	Schema
path	**pid** *required*	pid represents productid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/admin/delete` Delete admin details

Admin can delete his all details. This admin has no longer any authority to perform any action in the application

Responses

Code	Description	Links
200	OK *Content* /	No Links

DELETE `/admin/categories/delete/{catid}` delete an existing category

Admin can delete an existing category by providing the valid categoryid. Point to be noted that all the associated products will not be deleted

Parameters

Type	Name	Description	Schema
path	**catid** *required*	Here catid represents categoryid	integer (int32)

Responses

Code	Description	Links
200	OK *Content* /	No Links

Components

Schemas

Address

Properties

Name	Description	Schema
state *required*		string
city *required*		string
pincode *required*		string

CustomerDTOV2

Properties

Name	Description	Schema
password *optional*		string
customerName *optional*		string
customerMobile *optional*		string
address *optional*		Address

Customer

Properties

Name	Description	Schema
userid *optional*		integer (int32)

Name	Description	Schema
role *required*		enum (ROLE_ADMIN, ROLE_CUSTOMER,ROLE_ANONYMOUS)
email *optional*		string
password *optional*	*write only*	string
customerName *required*	**Maximum Length**: 20 **Minimum Length**: 3	string
customerMobile *required*		string
address *optional*		Address

Category

Properties

Name	Description	Schema
categoryId *optional*		integer (int32)
categoryName *required*		string
imageUrl *required*		string
description *required*		string

Product

Properties

Name	Description	Schema
productId *optional*		integer (int32)
productImage *required*		string
productName *required*		string
description *required*		string

Name	Description	Schema
costPrice *required*	**Minimum**: 1.0	number (double)
price *required*	**Minimum**: 1.0	number (double)
avgRatings *optional*	**Maximum**: 5 **Minimum**: 0	number (double)
noOfPeopleRated *optional*		integer (int32)
stockQuantity *optional*		integer (int32)
category *optional*		Category

AdminDTOV2

Properties

Name	Description	Schema
password *optional*		string
adminName *optional*		string
adminMobile *optional*		string

Admin

Properties

Name	Description	Schema
userid *optional*		integer (int32)
role *required*		enum (ROLE_ADMIN, ROLE_CUSTOMER,ROLE_ANONYMOUS)
email *optional*		string
password *optional*	*write only*	string
adminName *required*	**Maximum Length**: 20 **Minimum Length**: 3	string

Name	Description	Schema
adminMobile *required*		string
reports *optional*		< Report > array

Report

Properties

Name	Description	Schema
reportID *optional*		integer (int32)
description *required*		string
adminName *required*		string
admin *optional*		Admin

ProductDTO

Properties

Name	Description	Schema
productId *optional*		integer (int32)
productImage *optional*		string
productName *optional*		string
description *optional*		string
costPrice *optional*		number (double)
price *optional*		number (double)
quanity *optional*		integer (int32)
categoryName *optional*		string

Payment

Properties

Name	Description	Schema
paymentID *optional*		integer (int32)
paymentType *required*		enum (DEBITCARD,POD,PAYPAL,WALLET,NETBANKING,UPI)

OrderDTO

Properties

Name	Description	Schema
orderID *optional*		integer (int32)
customerID *optional*		integer (int32)
details *optional*		< OrderDetail > array
total_order_amount *optional*		number (double)
orderDateTime *optional*		string (date-time)
shipDateTime *optional*		string (date-time)
deliveryDateTime *optional*		string (date-time)
shipper *optional*		string
suppliers *optional*		string

OrderDetail

Properties

Name	Description	Schema
orderDetailID *optional*		integer (int32)

Name	Description	Schema
quantity *required*		integer (int32)
order *optional*		Orders
product *optional*		Product
supplier *optional*		Supplier

Orders

Properties

Name	Description	Schema
orderID *optional*		integer (int32)
shipDateTime *optional*		string (date-time)
deliveryDateTime *optional*		string (date-time)
total_order_amount *required*		number (double)
returnRequested *optional*		boolean
cancelled *optional*		boolean
shipper *optional*		Shipper
payment *optional*		Payment
customer *optional*		Customer

Shipper

Properties

Name	Description	Schema
shipperID *optional*		integer (int32)
companyName *required*		string

Name	**Description**	**Schema**
mobileNumber *required*		string
active *optional*		boolean

Supplier

Properties

Name	**Description**	**Schema**
supplierID *optional*		integer (int32)
companyName *required*		string
city *optional*		string
state *optional*		string
pincode *optional*		string
mobileNumber *required*		string
categoryName *required*		string
active *optional*		boolean

CustomerDTO

Properties

Name	**Description**	**Schema**
customerName *optional*		string
customerEmail *optional*		string
password *optional*		string
customerMobile *optional*		string
address *optional*		Address
role *optional*		string

CartDTO

Properties

Name	Description	Schema
productId *optional*		integer (int32)
quantity *optional*		integer (int32)

ProductDTOV2

Properties

Name	Description	Schema
productId *optional*		integer (int32)
quantity *optional*		integer (int32)

AdminDTO

Properties

Name	Description	Schema
adminName *optional*		string
adminEmail *optional*		string
adminMobile *optional*		string
password *optional*		string
role *optional*		string

CartDTOV2

Properties

Name	Description	Schema
cartId *optional*		integer (int32)
categoryName *optional*		string
productId *optional*		integer (int32)

Name	Description	Schema
productName *optional*		string
imageUrl *optional*		string
quantity *optional*		integer (int32)
price *optional*		number (double)

CartDTOV3

Properties

Name	Description	Schema
quantity *optional*		integer (int32)
subtotal *optional*		number (double)
list *optional*		< CartDTOV2 > array

ProductDTOV3

Properties

Name	Description	Schema
productID *optional*		integer (int32)
noOfProductsSold *optional*		integer (int32)
totalSales *optional*		number (double)
categoryName *optional*		string

RevenueDTO

Properties

Name	Description	Schema
categoryName *optional*		string

Name	Description	Schema
categorywiseTotalRevenue *optional*		number (double)
revenuePercentage *optional*		string

SalesAnalysisDTO

Properties

Name	Description	Schema
noOfOrders *optional*		integer (int32)
avgOrderValue *optional*		number (double)
totalRevenue *optional*		number (double)
topSellingProducts *optional*		< ProductDTOV3 > array
revenueBreakdown *optional*		< RevenueDTO > array
conversionRate *optional*		string
totalSalesAmount *optional*		number (double)

DaywiseSalesData

Properties

Name	Description	Schema
serialNo *optional*		integer (int32)
totalSalesAmount *optional*		number (double)
noOfOrders *optional*		integer (int32)
avgOrderValue *optional*		number (double)
totalRevenue *optional*		number (double)

Name	Description	Schema
conversionRate *optional*		string
date *optional*		string (date)
topSellingProducts *optional*		< ProductDTOV3 > array
revenueBreakdown *optional*		< RevenueDTO > array

ProductDTOV4

Properties

Name	Description	Schema
productID *optional*		integer (int32)
noOfProductsSold *optional*		integer (int32)
totalSales *optional*		number (double)
categoryName *optional*		string
avgRatings *optional*		number (double)

ProductDTOV5

Properties

Name	Description	Schema
categoryID *optional*		integer (int32)
categoryName *optional*		string
productID *optional*		integer (int32)
productName *optional*		string
salePrice *optional*		number (double)
costPrice *optional*		number (double)

Name	Description	Schema
noOfProducts *optional*		integer (int32)
revenueGenerated *optional*		number (double)

ReturnRequestDTO

Properties

Name	Description	Schema
orderID *optional*		integer (int32)
requestedON *optional*		string (date-time)
fulfillON *optional*		string (date-time)
refundON *optional*		string (date-time)
status *optional*		enum (PENDING,SHIPPED,DELIVERED,CANCELLED_AND_REFUND_IN_PROGESS,RETURN_IN_PROGRESS,RETURNED_AND_REFUND_IN_PROGESS,REFUNDED)

CancelledDTO

Properties

Name	Description	Schema
orderID *optional*		integer (int32)
cancelledON *optional*		string (date-time)
refundON *optional*		string (date-time)

Name	Description	Schema
status *optional*		enum (PENDING,SHIPPED,DELIVERED,CANCELLED_AND_REFUND_IN_PROGESS,RETURN_IN_PROGRESS,RETURNED_AND_REFUND_IN_PROGESS,REFUNDED)