**Invoicing System**

Invoicing System has 5 components:

1. Category
2. Product
3. Customer
4. Invoice
5. Cart

Default data has been given for the main models as below:

**Category**:

Id=1, Name="Electronics", Description="Best Electronics"

Id=2, Name="Kitchen", Description="Best Kitchen products"

Id=3, Name="Books", Description="Best Books"

Id=4, Name="Health", Description="Best Fitness products"

Id=5, Name="Sports", Description="Best Sports"  
  
**Products**:

Id = 1,

Name = "Monitor",

Description = "A spacious and durable Monitor. FHD.",

Price = 19.99m,

Quantity = 50,

CategoryId = 1

Id = 2,

Name = "Toaster",

Description = "A sleek toaster for perfectly toasted bread.",

Price = 29.99m,

Quantity = 30,

CategoryId = 2

Id = 3,

Name = "Novel",

Description = "An engaging novel that keeps you hooked till the last page.",

Price = 12.99m,

Quantity = 40,

CategoryId = 3

Id = 4,

Name = "Dumbbells",

Description = "A set of dumbbells for strength training workouts.",

Price = 39.99m,

Quantity = 20,

CategoryId = 4

Id = 5,

Name = "Basketball",

Description = "A durable basketball for outdoor and indoor play.",

Price = 24.99m,

Quantity = 15,

CategoryId = 5

**Customer**:

Id = 1, Name="John", Address="Mangalore", ContactNumber="9876543191", Email="abc@example.com"

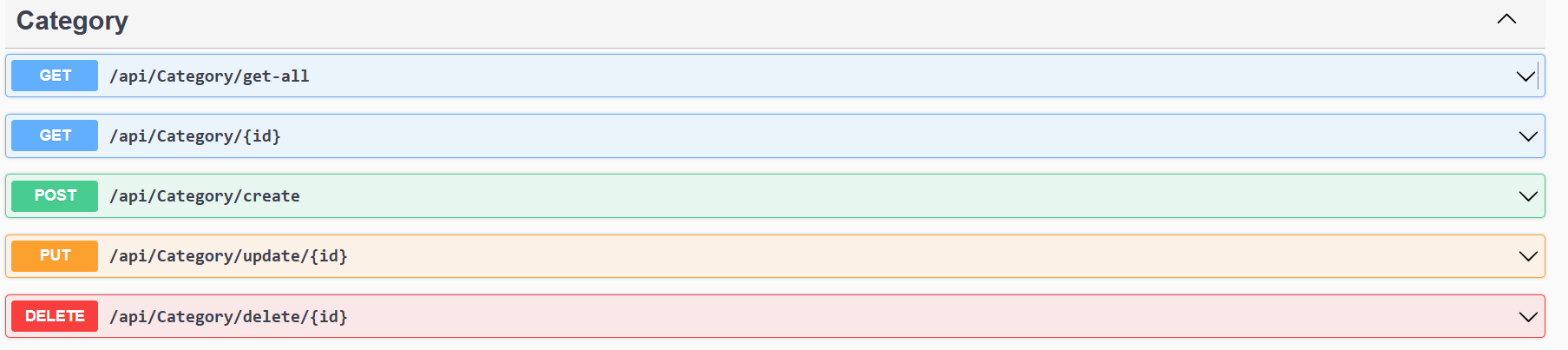
Id = 2, Name="Ben", Address="Chennai", ContactNumber="9874563210", Email="abc1@example.com"

Id = 3, Name="Carl", Address="Kolkota", ContactNumber="987463211", Email="abc2@example.com"

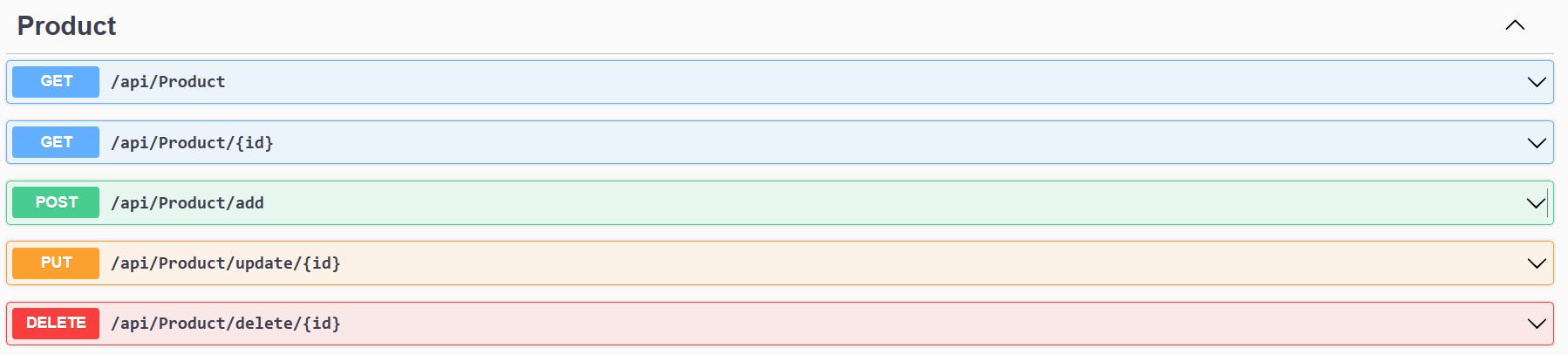
Id = 4, Name="Lexi", Address="London", ContactNumber="9874561233", Email="abc3@example.com"

Id = 5, Name="Marc", Address="Sydney", ContactNumber="98745612345", Email=[abc4@example.com](mailto:abc4@example.com)  
  
  
Below are the API endpoints:

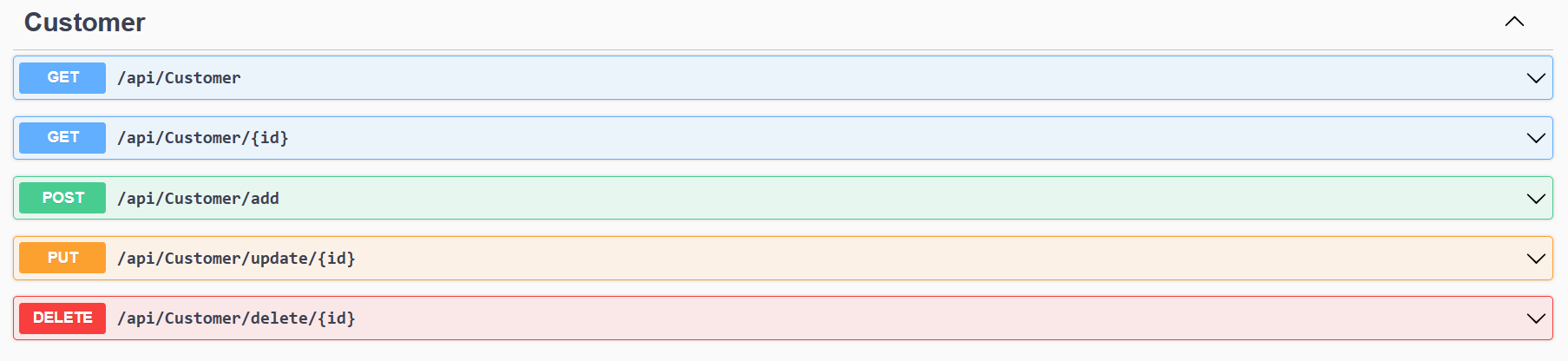
**Category**:



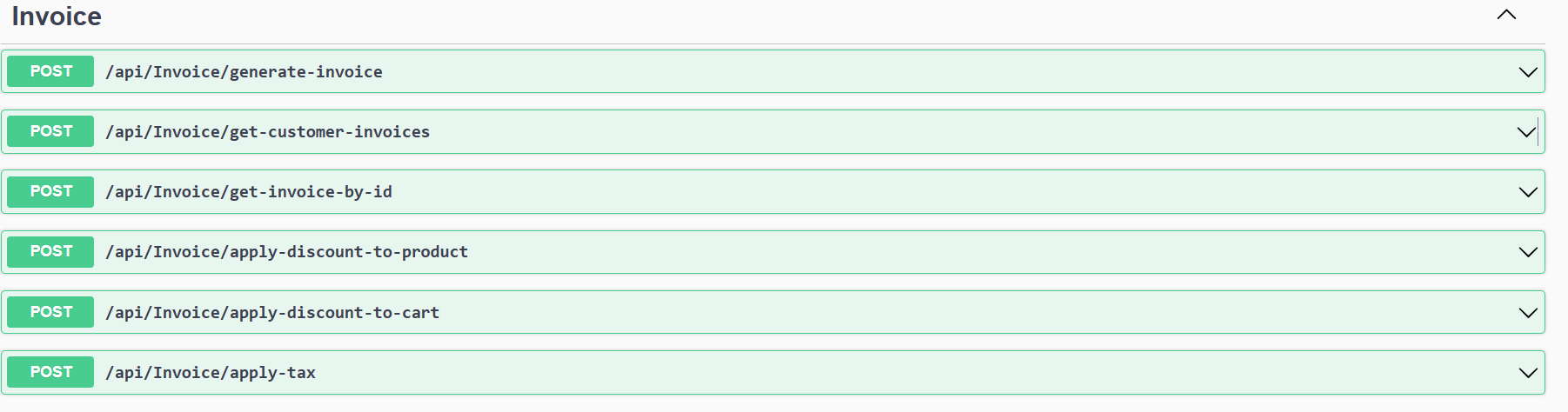
**Product**:



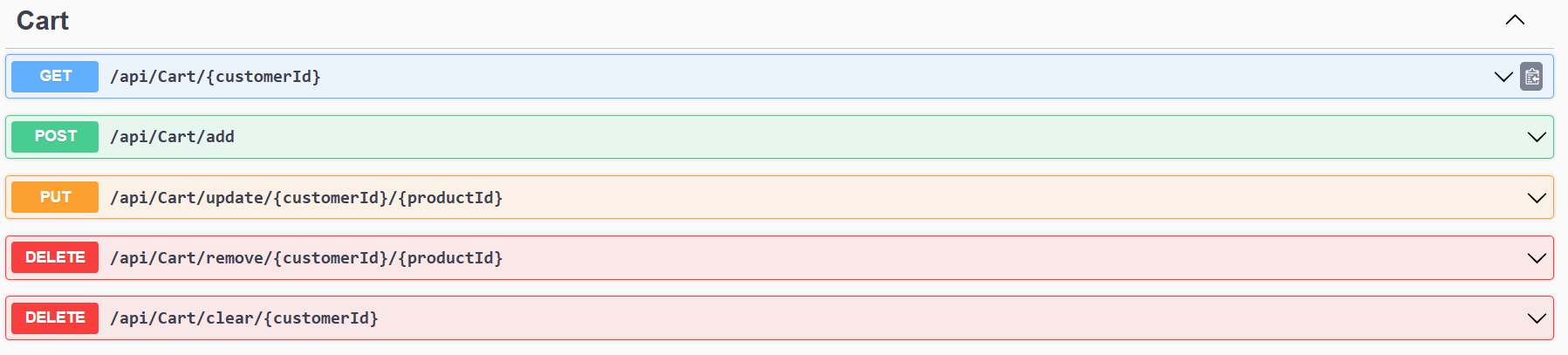
**Customer**:



**Invoice:**

****

**Cart:**

****

**Details on Endpoints:**

**Category Endpoint details:**

* **Add Category [api/Category/create]:** To add a category, user needs to pass the category name and description in the request body.
* **Update Category** [**api/Category/update/id**]: To update category, user needs to pass the id of category and the data to be updated.
* **Delete Category** [**api/Category/delete/id**]: To delete category, user needs to pass the id of category.
* **Get All Categories** [**api/Category/get-all**]: This will list all the categories.
* **Get Category by ID** [**api/Category/id**]: This will retrieve the category details based on provided categoryid.

**Product Endpoint details [There must be category details before adding product]:**

* **Add Product [api/Product/add]:** To add a product, user needs to pass the product name, description, price, quantity and categoryId in the request body.
* **Update Product** [**api/ Product /update/id**]: To update product, user needs to pass the id of product and the data to be updated.
* **Delete Product** [**api/ Product /delete/id**]: To delete product, user needs to pass the id of product.
* **Get All Products** [**api/ Product**]: This will list all the products.

**Get Product by ID** [**api/ Product /id**]: This will retrieve the product details based on provided productId.

**Customer Endpoint details:**

* **Add Customer [api/Customer/add]:** To add a customer, user needs to pass the customer’s name, email, address and contact number in the request body.
* **Update Customer** [**api/ Customer /update/id**]: To update customer, user needs to pass the id of customer and the data to be updated.
* **Delete Customer** [**api/ Customer /delete/id**]: To delete customer data, user needs to pass the id of customer record.
* **Get All Customer**s [**api/ Customer**]: This will list all the customers.

**Get Customer by ID** [**api/ Customer /id**]: This will retrieve the customer details based on provided customerid.

**Cart Endpoint details:**

* **View Cart Items of a customer [api/Cart/customerId]:** This will retrieve the cart details of the customer.
* **Add Items to cart [api/Cart/add]:** To add items to cart, user needs to pass customerId, productid and the quantity.
* **Update quantity of an item in the cart [api/Cart/update/customerid/productid]:** This will allow user to update the quantity of an item in the cart.
* **Remove an item from the cart [api/Cart/remove/customerid/productid]:** This will allow user to remove an item from the customer’s cart.
* **Clear the customer’s cart [api/Cart/clear/customerid]:** This will allow user to clear the customers cart details.

**Invoice Endpoint details:***(After the user has submitted the cart details and proceed with payment, this section can be used to get the details of items stored in the cart and generate the invoice.)*

* **Generate Invoice [api/Invoice/generate-invoice]:** This will generate invoice for the items purchased by the customer. User needs to pass the customer id and select the payment method to continue.
* **Get Invoices of a customer [api/Invoice/get-customer-invoice]:** This will generate list of invoices for a customer.
* **Get Invoices by Invoice ID [api/Invoice/get-invoice-by-id]:** This will generate invoice details based on the given invoice ID
* **Apply Discount for an item in the cart [api/Invoice/apply-discount-to-product]:** This will apply discount to a product in the customer’s cart. User needs to pass customerId, productid and the discount Percentage value.
* **Apply Discount for the whole cart of a customer [api/Invoice/apply-discount-to-cart]:** This will apply discount to the customer’s cart. User needs to pass customerId and the discount Percentage value.
* **Apply Tax [api/Invoice/apply-tax]:** This will apply tax to the customer’s subtotal. User needs to pass customerId and the Tax Percentage value.

**Steps to run the application:**

* **Open the solution file “InvoicingSystem.sln”**
* **For using application without the default data stored, comment the code inside the constructors inside the service classes under “Services” folder.  
  Update CategoryService.cs  
  Update ProductService.cs  
  Update CustomerService.cs**
* **Click on the Run button in visual studio code to run the application.**
* **The application data will be persistent until the application is restarted.**
* **Follow below steps:**
  + **Step 1: Add Categories**
  + **Step 2: Add Products**
  + **Step 3: Add Customer Information**
  + **Step 4: Add Products to cart**
  + **Step 5: Apply Discount**
  + **Step 6: Apply Tax**
  + **Step 5: Generate Invoice (Items in the cart will be removed when the invoice is generated. So, customer can proceed with his next purchase)  
      
    Payment Types with code**1-Cash,

2-CreditCard,

3-DebitCard,

4-OnlinePayment