## RESTful APIs

### Exercise 1: Product Collection

Create a RESTful API to manage a collection of products. The API should allow clients to retrieve a list of products, retrieve a single product by ID, filter products by category using route parameters and query string parameters, and create new products.

**Project Structure:**

* Controller: **ProductsController**
* This controller will handle all the HTTP requests related to product management.
* It will include methods to retrieve a list of products, retrieve a single product by ID, filter products by category, and create new products.
* Create a folder named **Data** and have the **ApplicationDbContext.cs file** toset up the database context for the application with the DbSet property **Products**.
* Model: **Product**
* Create a folder named **Models** and create the model for Product.
* Represents a product in the system.
* Properties:
* **Id**: Integer - The unique identifier of the product (auto-increments by 1)
* **Name**: String - The name of the product.
* **Category**: String - The category of the product.
* **Price**: Decimal - The price of the product.
* **Stock**: Integer - The available stock of the product.

**Controller Methods:**

1. GET /api/products:

* Retrieves a list of all products.
* Status: **200**
* Route: **/api/products**
* Method: **public ActionResult<IEnumerable<Product>> GetProducts()**
* Response: List of **Product** objects

2. GET /api/products/{id}

* Retrieves a single product by its ID.
* Status: **200**
* Route: **/api/products/{id}**
* Method: **public ActionResult<Product> GetProductById(int id)**
* Parameters:**id**: Integer (Route parameter)
* Response: **Product** object with the specified ID

3. GET /api/products/filter?category={categoryName}

* Retrieves a list of products filtered by category.
* Route: **/api/products/filter**
* Status: **200**
* Method: **public ActionResult<IEnumerable<Product>> GetProductsByCategory([FromQuery] string category)**
* Query String Parameter:**category**: String
* Response: List of **Product** objects filtered by the specified category

4. POST /api/products

* Creates a new product.
* Route: **/api/products**
* Status: **201**
* Method: **public ActionResult<Product> CreateProduct(Product product)**
* Request Body: **Product** object (JSON)
* Response: **Product** object representing the newly created product

**Example Request:**

1. Retrieve all products:

* Request: GET /api/products
* Response: List of all products in the system.

2. Retrieve a product by ID:

* Request: GET /api/products/1
* Response: Details of the product with ID 1.

3. Retrieve products filtered by category:

* Request: GET /api/products/filter?category=electronics
* Response: List of products belonging to the "electronics" category.

4. Create a new product:

* Request:

POST /api/products

Content-Type: application/json

{

"Name": "New Product",

"Category": "electronics",

"Price": 99.99,

"Stock": 10

}

* Response: Details of the newly created product.

### Exercise 2: Artwork Collection

**Create a RESTful API to manage a collection of artworks. The API should allow clients to retrieve a list of artworks, retrieve a single artwork by ID, filter artworks by artist using route parameters and query string parameters, and create new artworks.**

* Controller: **ArtworksController**
* This controller will handle all the HTTP requests related to artwork management.
* It will include methods to retrieve a list of artworks, retrieve a single artwork by ID, filter artworks by artist, and create new artworks.
* **Data Folder:** Create a folder named **Data** and have the **ApplicationDbContext.cs file** toset up the database context for the application with the DbSet property **Artworks.**
* Model: **Artwork**
* Create a folder named Models and create the model for Artwork.
* Represents an artwork in the system.
* **Properties:**

1. **ArtworkId:** Integer - The unique identifier of the artwork (auto-increments by 1)
2. **Title:** String - The title of the artwork.
3. **Artist:** String - The artist who created the artwork.
4. **Year:** Integer - The year the artwork was created.
5. **Medium:** String - The medium used for the artwork.
6. **Description:** String - A brief description of the artwork

**Controller Methods:**

**1. GET /api/artworks**

* Retrieves a list of all artworks.
* Status: 200
* Route: **/api/artworks**
* Method: **public ActionResult<IEnumerable<Artwork>> GetArtworks()**
* Response: List of Artwork objects

**2. GET /api/artworks/{id}**

* Retrieves a single artwork by its ID.
* Status: 200
* Route: **/api/artworks/{id}**
* Method: **public ActionResult<Artwork> GetArtworkById(int id)**
* Parameters: id: Integer (Route parameter)
* Response: Artwork object with the specified ID

**3. GET /api/artworks/filter?artist={artistName}**

* Retrieves a list of artworks filtered by artist.
* Route: **/api/artworks**
* Status: 200
* Method: **public ActionResult<IEnumerable<Artwork>> GetArtworksByArtist([FromQuery] string artist)**
* Query String Parameter: artist: String
* Response: List of Artwork objects filtered by the specified artist

**4. POST /api/artworks**

* Creates a new artwork.
* Route: **/api/artworks**
* Status: 201
* Method: **public ActionResult<Artwork> CreateArtwork(Artwork artwork)**
* Request Body: Artwork object (JSON)
* Response: Artwork object representing the newly created artwork

**Example Request:**

1.Retrieve all artworks:

* Request: GET /api/artworks
* Response: List of all artworks in the system.

2. Retrieve an artwork by ID:

* Request: GET /api/artworks/1
* Response: Details of the artwork with ID 1.

3.Retrieve artworks filtered by artist:

* Request: GET /api/artworks/filter?artist=VanGogh
* Response: List of artworks created by "Van Gogh".

4.Create a new artwork:

* Request:

POST /api/artworks

Content-Type: application/json

{

"Title": "Starry Night",

"Artist": "Vincent van Gogh",

"Year": 1889,

"Medium": "Oil on canvas",

"Description": "A famous painting by Vincent van Gogh depicting a night sky."

}

* Response: Details of the newly created artwork.