Misr Insurance Task – Dot Net Developer

REST API – Postman Requests & Examples

Base URL: https://localhost:5225 (adjust the port as needed)

Created By :

Mahmoud Khaled Ragab – Software Developer

Notes

- All responses are wrapped in ApiResult<T> when applicable.
- Content-Type for request bodies: application/json.
- Replace IDs with real values from your database.

1) Create Customer

Method & URL

```
POST https://localhost:5225/api/Customers
```

Headers

Header	Value
Content-Type	application/json

Request Body (JSON)

```
{
  "name": "Mahmoud Khaled",
  "email": " mahmoud.khaled@gmail.com",
  "phone": "+201234567890"
}
```

Sample 201 Created Response

```
{
  "isSuccess": true,
  "message": null,
  "data": {
    "id": 101,
    "name": " Mahmoud Khaled ",
    "email": " mahmoud.khaled@gmail.com ",
    "phone": "+201234567890"
  }
}
```

Sample 400 Bad Request (Validation Error)

```
{
  "isSuccess": false,
  "message": "Invalid request",
  "errors": {
    "Name": ["Customer name is required."],
    "Email": ["Invalid email format."]
  }
}
```

Sample 409 Conflict (Email Exists)

```
{
  "isSuccess": false,
  "message": "Email already exists"
}
```

2) Get All Customers

Method & URL

```
GET https://localhost:5225/api/Customers
```

Sample 200 OK Response

```
{
  "isSuccess": true,
  "message": null,
  "data": [
    {
      "id": 1,
      "name": "Mahmoud Khaled",
      "email": "mahmoud.khaled@gmail.com",
      "phone": "+201234567890"
    },
    {
      "id": 2,
      "name": "Jane Smith",
      "email": "jane.smith@example.com",
      "phone": null
    }
  ]
}
```

3) Get Customer By Id

Method & URL

```
GET https://localhost:5225/api/Customers/1
```

Sample 200 OK Response

```
{
  "isSuccess": true,
  "message": null,
  "data": {
    "id": 1,
    "name": "John Doe",
    "email": "john.doe@example.com",
    "phone": "+201234567890"
  }
}
```

Sample 404 Not Found

```
{
  "isSuccess": false,
  "message": "Customer with id 1 not found"
}
```

4) Create Order

Method & URL

```
POST https://localhost:5225/api/Order
```

Headers

Header	Value
Content-Type	application/json

Request Body (JSON)

Assumes unique product IDs and quantity = 1 each (per your simplified implementation).

```
{
  "customerId": 1,
  "productIds": [2, 5, 7]
}
```

Sample 200 OK Response

```
{
  "isSuccess": true,
  "message": null,
  "data": {
    "id": 3001,
    "customerName": "John Doe",
    "status": "Pending",
    "numberOfProducts": 3
  }
}
```

Sample 400 Bad Request

```
{
  "isSuccess": false,
  "message": "Order must contain at least one product"
}
```

Sample 404 Not Found (Customer)

```
{
  "isSuccess": false,
  "message": "Customer with id 99 not found"
}
```

5) Get Order By Id

Method & URL

```
GET https://localhost:5225/api/Order/1
```

Sample 200 OK Response

```
{
  "isSuccess": true,
  "message": null,
  "data": {
    "id": 1,
    "customerName": "John Doe",
    "status": "Pending",
    "numberOfProducts": 3
  }
}
```

Sample 404 Not Found

```
{
  "isSuccess": false,
  "message": "Order with id 1 not found"
}
```

6) Update Order Status to Delivered

Method & URL

```
POST https://localhost:5225/api/Order/UpdateOrderStatus/1
```

Headers

Header	Value
Content-Type	application/json

Request Body

If your endpoint does not require a body, you can omit it. If using a body:

```
{
  "status": "Delivered"
}
```

Sample 200 OK Response

Updates status to Delivered and decrements product stock for items in the order.

```
{
  "isSuccess": true,
  "message": "Order status updated to Delivered",
  "data": {
    "id": 1,
    "customerName": "John Doe",
    "status": "Delivered",
    "numberOfProducts": 3
  }
}
```

Sample 400 Bad Request (Invalid Transition)

```
{
  "isSuccess": false,
  "message": "Only Pending orders can be delivered"
}
```

Sample 404 Not Found

```
{
  "isSuccess": false,
  "message": "Order with id 1 not found"
}
```