

Product Manager API Rest

Table of Contents

1.2. API REST	2
1.2.1. Create Order	2
1.2.1.1. Test Case 1	3
1.2.1.2. Test Case 2	3
1.2.1.3. Test Case 3	4
1.2.2. Get orders by Customer	4
1.2.2.1. Test Case 1	6
1.2.2.2. Test Case 2	6
1.3. Diagrams	7
1.3.1. ER	7
1.3.2. Component	8
1.2. Web Page	9
1.2.1. Table order	9

1.2.API REST

1.2.1. Create Order

Title	Create order		
URL	/v1/order		
Method	The request type POST	Version	1.0
URL Params	None		
Data Params	URL http://ec2-52-14-66-207.us-east-2.compute.amazonaws.com:8080/productManager-api-rest/v1/order Content { "order": { "deliveryAddress": "15 Queens Park Road, W32 YY, UK", "customer": { "customerId": 1 } }, "products": [{ "productId": 1 }, { "productId": 2 }] }		
Success Response	The order was created. Code: 200 Content: { "description": "OK", "code": 200 }		
Error Response	- Validate the maximum products Example: Code: 400 BAD_REQUEST Content: { "description": "The amount of products must be less or equal to 5 ", "code": 400 } -Not available products by Customer Example: Code: 400 BAD_REQUEST Content: { "description": "The product with id 4 is not available for 'Manny Bharna ' ", "code": 400 }		

1.2.1.1. Test Case 1

The screenshot shows a REST client interface with a POST request to `http://ec2-52-14-66-207.us-east-2.compute.amazonaws.com:8080/productManager-api-rest/v1/order`. The request body is a JSON object with the following structure:

```
1 {
2   "order": {
3     "deliveryAddress": "15 Queens Park Road, W32 YYY, UK",
4     "customer": {
5       "customerId": 1
6     }
7   },
8   "products": [
9     {
10      "productId": 1
11    },
12    {
13      "productId": 2
14    },
15    {
16      "productId": 2
17    },
18    {
19      "productId": 3
20    }
21  ]
22 }
23 }
```

The response status is **200 OK** and the time taken is **306 ms**. The response body is a JSON object:

```
1 {
2   "description": "OK",
3   "code": 200
4 }
```

1.2.1.2. Test Case 2

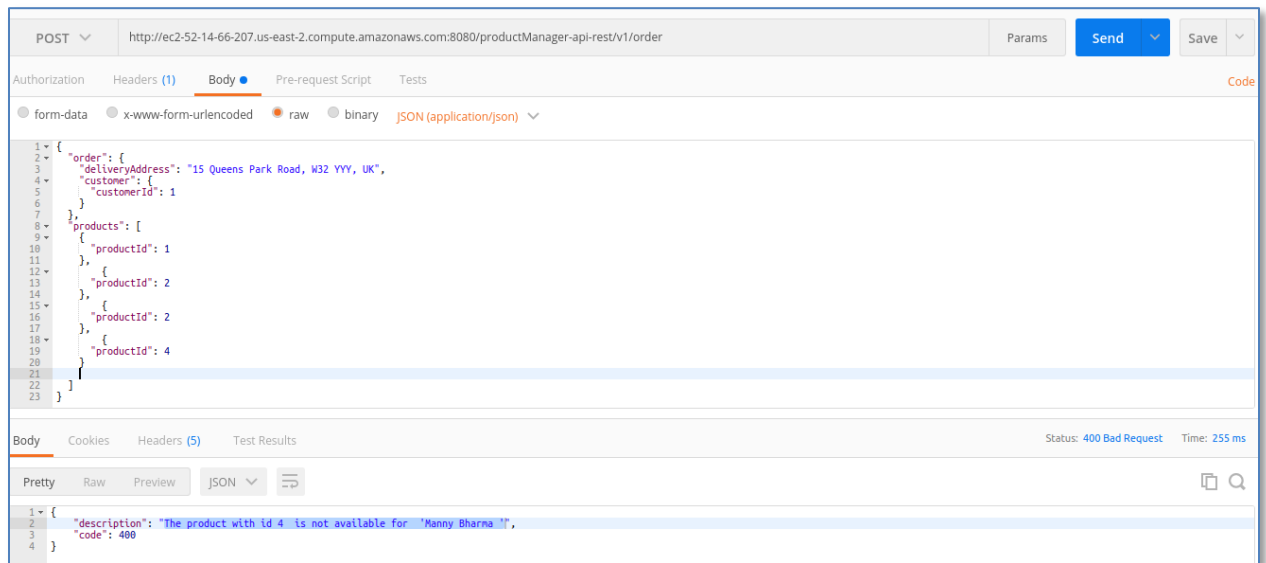
The screenshot shows a REST client interface with a POST request to the same endpoint as Test Case 1. The request body is a JSON object with the following structure:

```
5   "customer": {
6     "customerId": 1
7   },
8   "products": [
9     {
10      "productId": 1
11    },
12    {
13      "productId": 2
14    },
15    {
16      "productId": 2
17    },
18    {
19      "productId": 3
20    },
21    {
22      "productId": 3
23    },
24    {
25      "productId": 3
26    },
27    {
28      "productId": 3
29    }
30  ]
31 }
```

The response status is **400 Bad Request** and the time taken is **308 ms**. The response body is a JSON object:

```
1 {
2   "description": "The amount of products must be less or equal to 5",
3   "code": 400
4 }
```

1.2.1.3. Test Case 3



1.2.2. Get orders by Customer

Title	Get orders by Customer and dates		
URL	/v1/customer/:id/order		
Method	The request type GET	Version	1.0
URL Params	/v1/customer/{customerId}/order? startDate=[startDate]& endDate=[startDate] Example: /v1/customer/1/order?startDate=2017-09-26&endDate=2017-10-26		
Data Params	{customerId}: Id customer [startDate] : From date format "yyyy-MM-dd" [endDate] : End date format "yyyy-MM-dd" Example: http://ec2-52-14-66-207.us-east-2.compute.amazonaws.com:8080/productManager-api-rest/v1/customer/1/order?startDate=2017-09-27&endDate=2017-10-27		
Success Response	The order was found. Code: 200 Content: <pre>{ "customerId": 1, "email": "manny@Bharma.com", "name": "Manny Bharna", "availableProducts": [{ "productId": 1, "name": "Product A", "price": 100.05 }] }</pre>		

	<pre>{ "productId": 2, "name": "Product B", "price": 200.1 }, { "productId": 3, "name": "Product C", "price": 300.05 }], "orders": [{ "orderId": 1, "deliveryAddress": "15 Queens Park Road, W32 YYY, UK", "orderTime": "2017-10-26", "orderDetails": [{ "orderDetailId": 1, "price": 200.1, "productDescription": "2 X Product A" }, { "orderDetailId": 2, "price": 200.1, "productDescription": "1 X Product B" }, { "orderDetailId": 3, "price": 300.05, "productDescription": "1 X Product C" }], "totalPrice": 700.25 }, { "orderId": 2, "deliveryAddress": "15 Queens Park Road, W32 YYY, UK", "orderTime": "2017-10-26", "orderDetails": [{ "orderDetailId": 4, "price": 100.05, "productDescription": "1 X Product A" }, { "orderDetailId": 5, "price": 200.1, "productDescription": "1 X Product B" }, { "orderDetailId": 6, "price": 600.1, "productDescription": "2 X Product C" }], "totalPrice": 900.25 }] }</pre>
Error Response	<p>- Order not found</p> <p>Example:</p> <p>Code: 404 Not Found</p> <p>Content:</p> <pre>{ "customerId": 0, "email": null, "name": null,</pre>

```
"availableProducts": null,  
"orders": []  
}
```

1.2.2.1. Test Case 1

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://ec2-52-14-66-207.us-east-2.compute.amazonaws.com:8080/productManager-api-rest/v1/customer/1/order?startDate=2017-09-27&endDate=2017-10-27`
- Authorization:** No Auth
- Status:** 200 OK
- Body (JSON):**

```
{  
  "customerId": 1,  
  "email": "nanny@bharna.com",  
  "name": "Nanny Bharna",  
  "availableProducts": [],  
  "orders": [  
    {  
      "orderId": 6,  
      "deliveryAddress": "15 Queens Park Road, W32 YYY, UK",  
      "orderTime": "2017-10-27",  
      "orderDetails": [  
        {  
          "orderDetailId": 11,  
          "price": 100.05,  
          "productDescription": "1 X Product A"  
        },  
        {  
          "orderDetailId": 12,  
          "price": 400.2,  
          "productDescription": "2 X Product B"  
        },  
        {  
          "orderDetailId": 13,  
          "price": 300.05,  
          "productDescription": "1 X Product C"  
        }  
      ],  
      "totalPrice": 800.3  
    }  
  ]  
}
```

1.2.2.2. Test Case 2

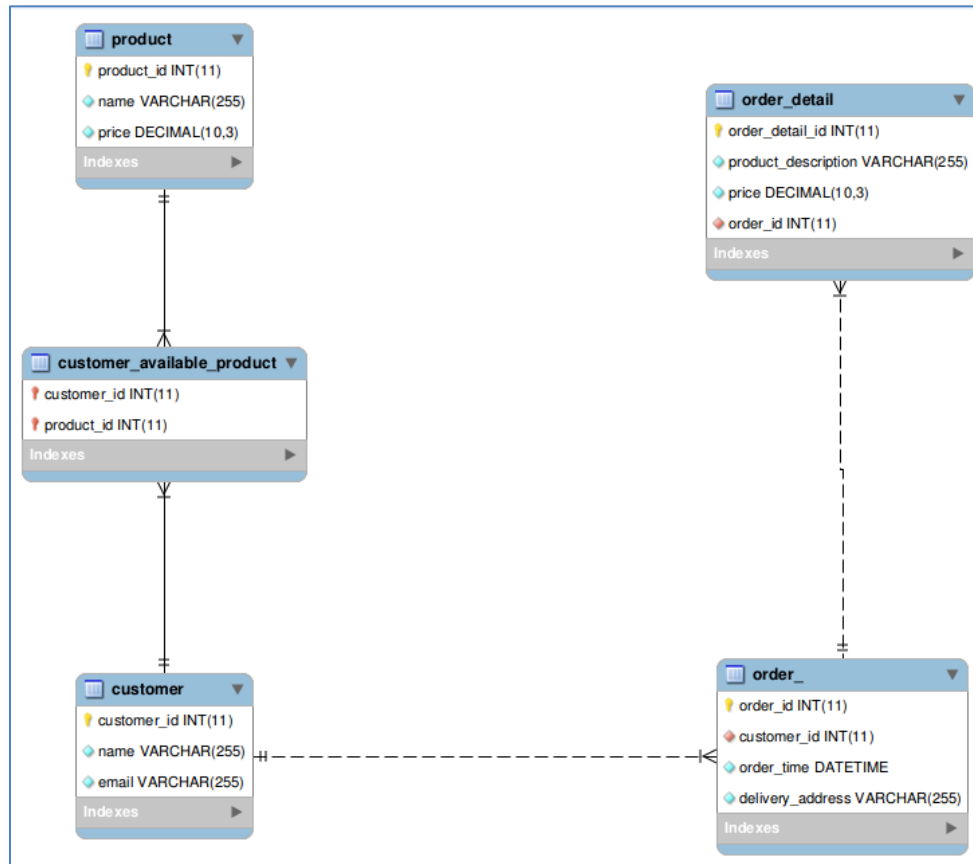
The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://ec2-52-14-66-207.us-east-2.compute.amazonaws.com:8080/productManager-api-rest/v1/customer/4/order?startDate=2017-09-27&endDate=2017-10-27`
- Authorization:** No Auth
- Status:** 404 Not Found
- Body (JSON):**

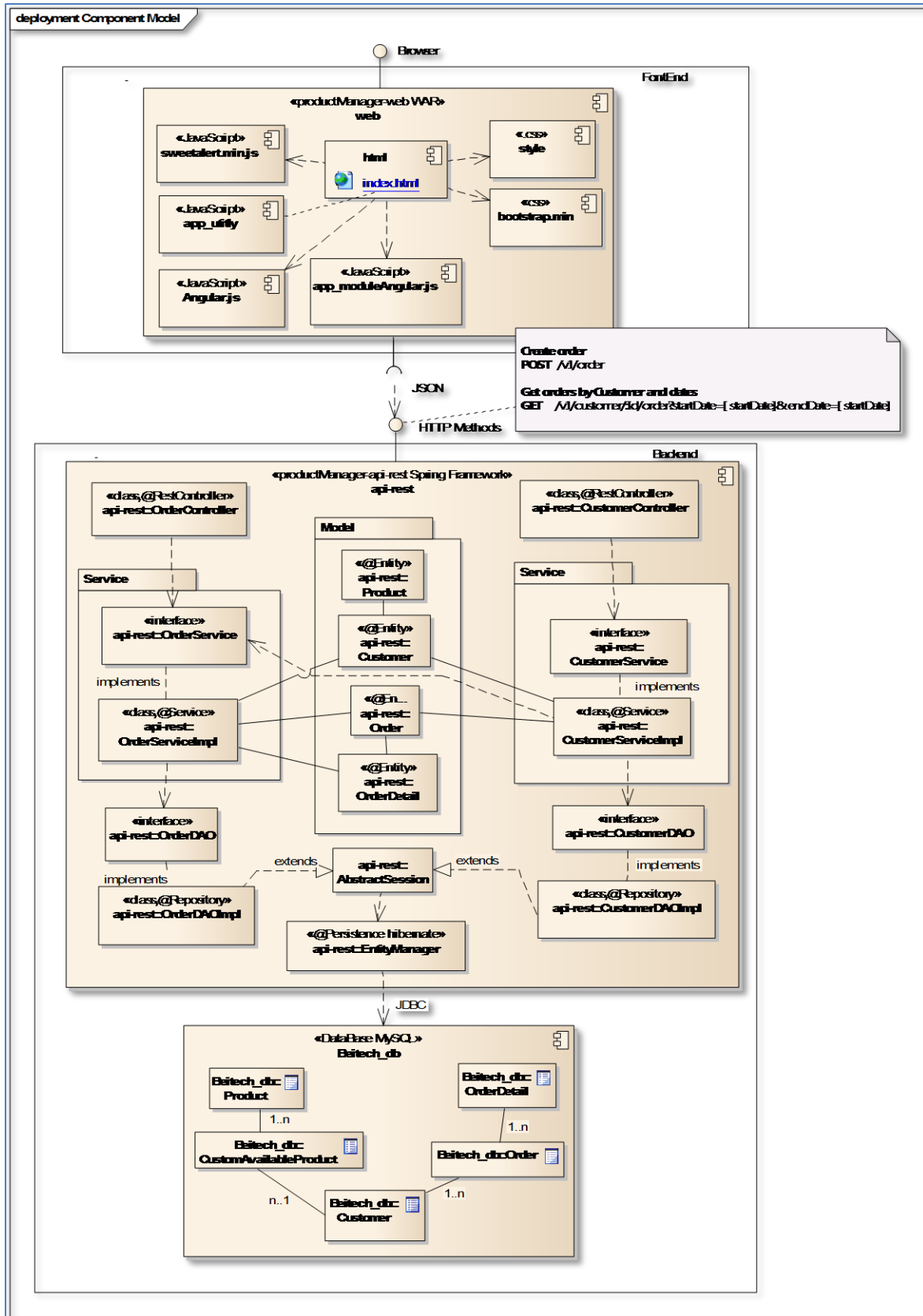
```
{  
  "customerId": 0,  
  "email": null,  
  "name": null,  
  "availableProducts": null,  
  "orders": []  
}
```

1.3. Diagrams

1.3.1. ER




1.3.2. Component




1.2. Web Page

1.2.1. Table order



Product Manager

Customer: --Select one customer--



Oops, 'Customer name is required

'Insufficient Data! Please provide values for Customer name!

OK

Creation Date				Products
2017-10-26	2			1 X Product A : \$ 200.1 1 X Product B : \$ 200.1 1 X Product C : \$ 300.05
2017-10-26	2			1 X Product A : \$ 100.05 1 X Product B : \$ 200.1 1 X Product C : \$ 600.1
2017-10-27	5	\$ 300.15	15 Queens Park Road, W32 YYY, UK	1 X Product A : \$ 100.05 1 X Product B : \$ 200.1
2017-10-27	6	\$ 800.3	15 Queens Park Road, W32 YYY, UK	1 X Product A : \$ 100.05 2 X Product B : \$ 400.2 1 X Product C : \$ 300.05

Product Manager

Customer: Mike Simm



Oops, 'Data not found

'Orders not found!

OK

Creation Date

Products

Product Manager

Customer: Manny Bharna

Search

Order List

2017-09-27 - 2017-10-27

Creation Date	Order Id	Total \$	Delivery Address	Products
2017-10-26	1	\$ 700.25	15 Queens Park Road, W32 YYY, UK	2 X Product A : \$ 200.1 1 X Product B : \$ 200.1 1 X Product C : \$ 300.05
2017-10-26	2	\$ 900.25	15 Queens Park Road, W32 YYY, UK	1 X Product A : \$ 100.05 1 X Product B : \$ 200.1 2 X Product C : \$ 600.1
2017-10-27	5	\$ 300.15	15 Queens Park Road, W32 YYY, UK	1 X Product A : \$ 100.05 1 X Product B : \$ 200.1
2017-10-27	6	\$ 800.3	15 Queens Park Road, W32 YYY, UK	1 X Product A : \$ 100.05 2 X Product B : \$ 400.2 1 X Product C : \$ 300.05

