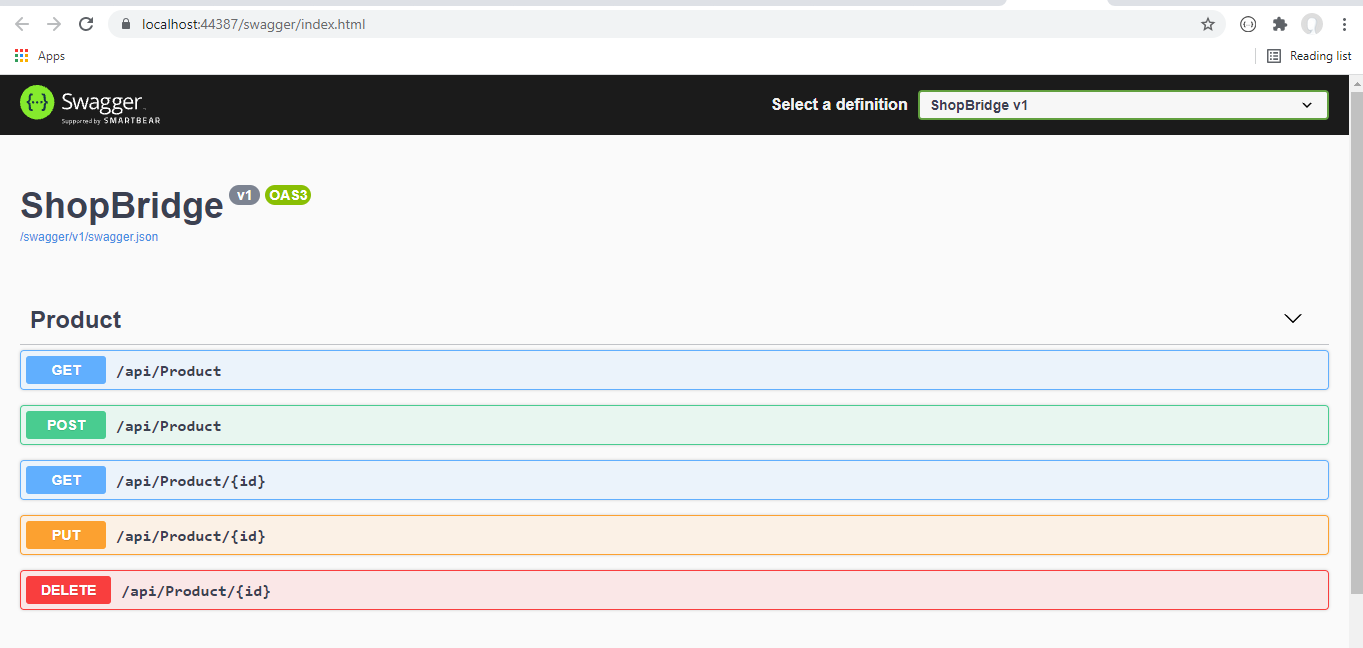
**ShopBridge**

ShopBridge is a Web API solution which consists of the following endpoints:



.Net 5.0 is used for building the Web API solution

Code First approach is used for building the database using EF Core

The Product consists of the following properties:

name

description

price

**1. Get Product:**

The Get Product api returns all the products present in the db.

Searching, sorting and filtering is also implemented in this api.

**> Sorting:**

Sorting can be done using:

**price\_desc:** sorts the results in the descending order of the price of the prduct

**price\_asc:** sorts the results in the ascending order of the price of the product

**name\_desc:** sorts the results in the descending order of the name of the product

**name\_asc:** sorts the results in the ascending order of the name of the product

**description\_desc:** sorts the results in the descending order of the description of the product

**description\_asc:** sorts the results in the asscending order of the description of the product

*By default it sorts the results in the ascending order of the name of the product.*

**> Searching:**

Searches the result on the basis of the name of the product

user has to provide name of the product as input.

**> Pagination:**

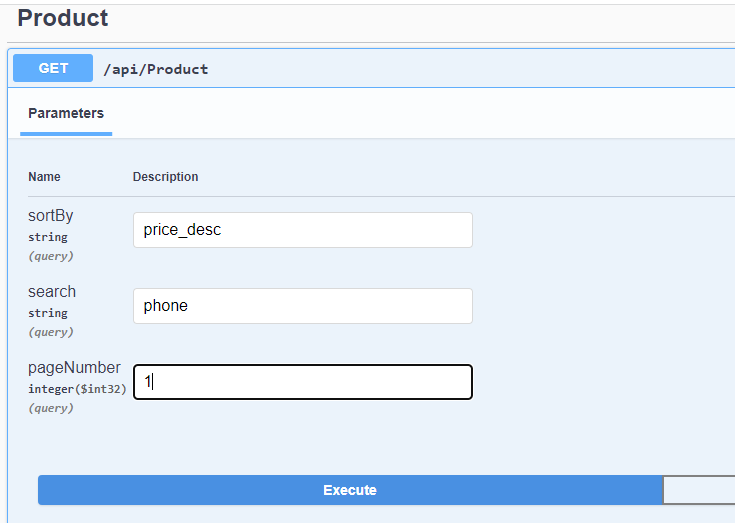
*The default page Size is set to 5*

*the default pageIndex is set to 1*

user has to provide the pageIndex/pageNumber as input

Returns 200 OK with the list of the product details.

Returns NotFound/Bad Request accordingly in case of errors



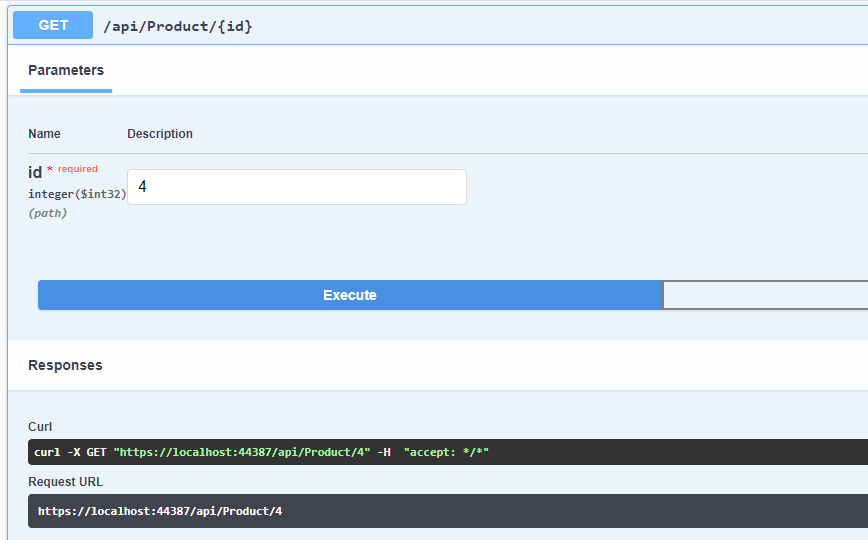
**2. GetProductById:**

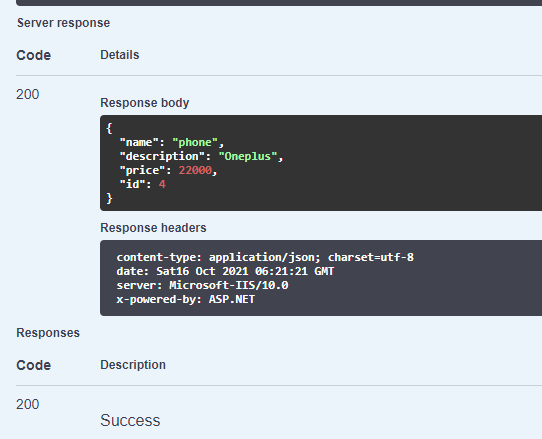
/api/Product/{id}

input: id of the product

Returns 200 OK with the product details in the response body

Returns NotFound/Bad Request accordingly in case of errors



**Post Product:**

Adds the product in the database

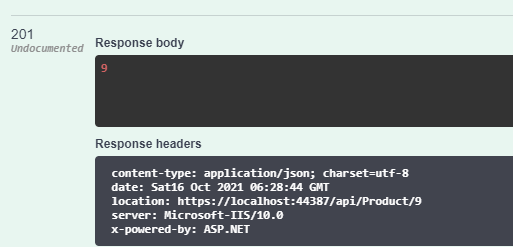
​/api​/Product

> provide the product detais in the request body

Returns a 201 with id of the product as the response body if the product is updated successfully

Returns NotFound/Bad Request accordingly in case of errors





**Put Product:**

Updates the product in the database

/api/Product/{id}

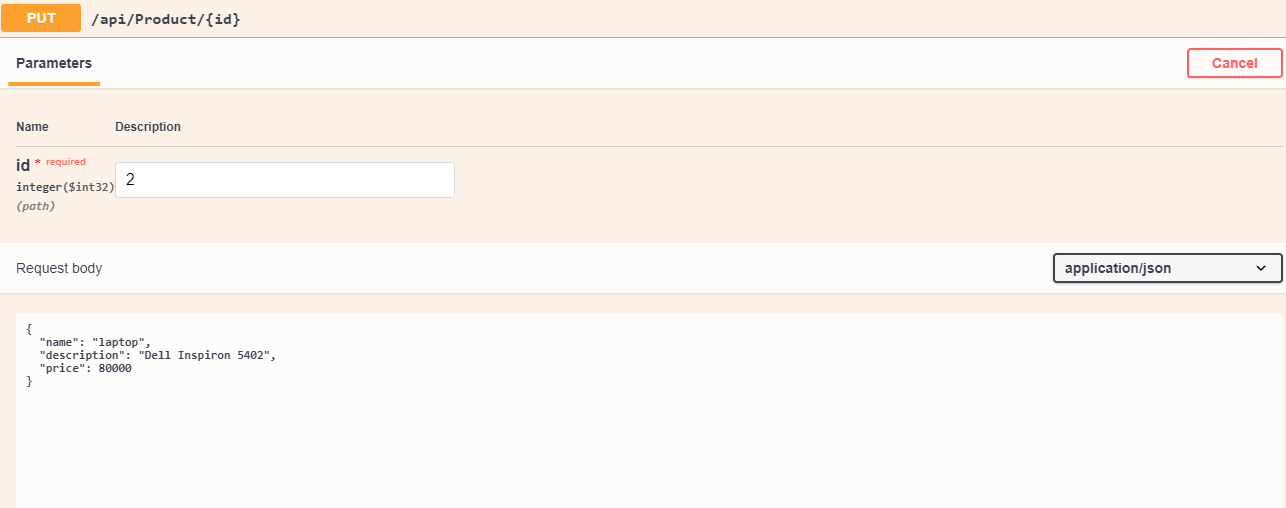
input parameters:

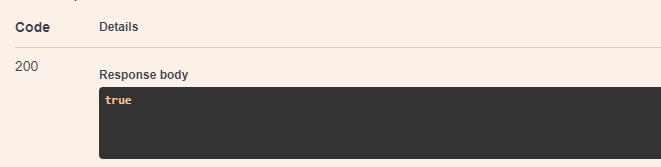
> id: id of the product to be updated

> provide the product detais in the request body

Returns a 200 OK with true as the response body if the product is updated successfully

Returns NotFound/Bad Request accordingly in case of errors





**Delete Product:**

Deletes the product in the database

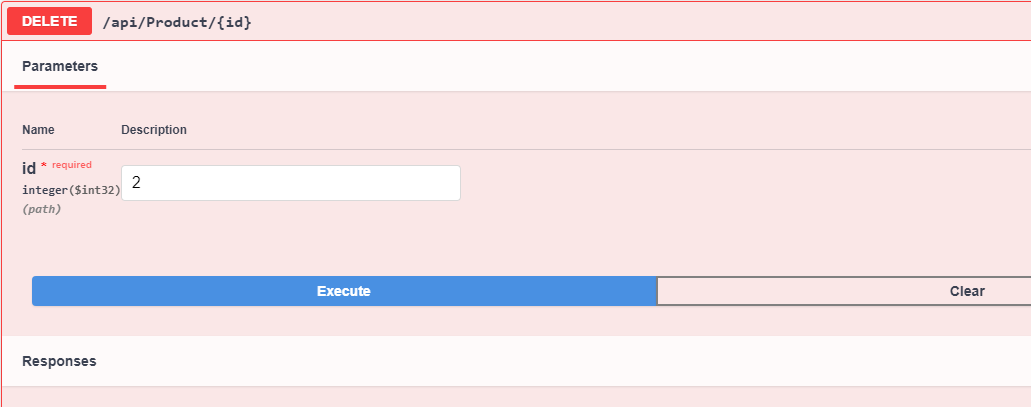
/api/Product/{id}

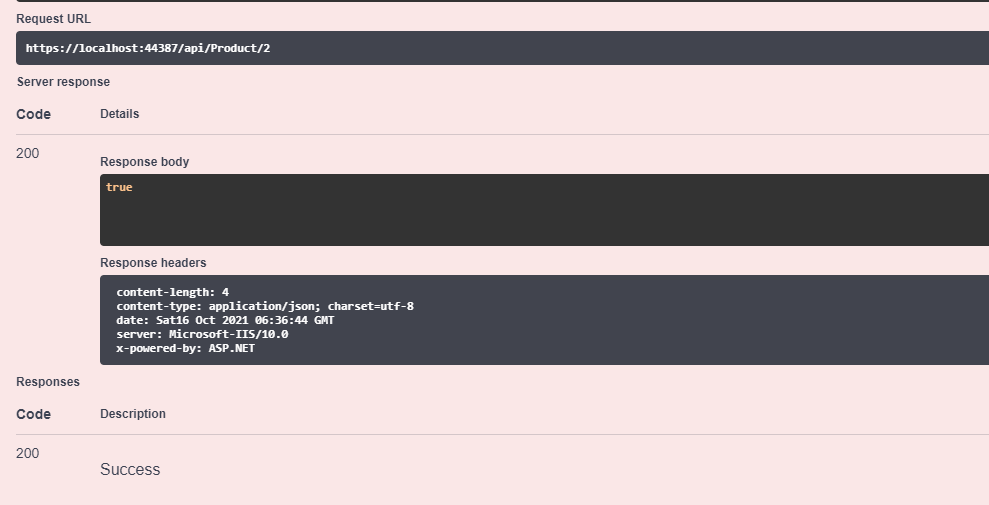
input parameters:

> id: id of the product to be updated

Returns a 200 OK with true as the response body if the product is updated successfully

Returns NotFound/Bad Request accordingly in case of errors



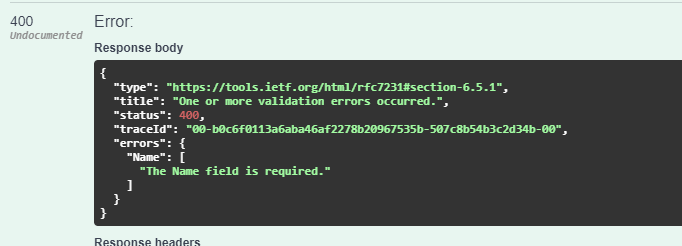


**Additional Features:**

**Validations, Exception handling and logging is implemented at all the places in the application**

**Note: Name and Price are the required fields while description is an optional field**

**Proper error will be thrown if the required fields are not provided.**

****

**{**

**"type": "**[**https://tools.ietf.org/html/rfc7231#section-6.5.1**](https://tools.ietf.org/html/rfc7231#section-6.5.1)**",**

**"title": "One or more validation errors occurred.",**

**"status": 400,**

**"traceId": "00-b0c6f0113a6aba46af2278b20967535b-507c8b54b3c2d34b-00",**

**"errors": {**

**"Name": [**

**"The Name field is required."**

**]**

**}**

**}**

**Note:** Make sure to change the connection string accordingly in the appSettings.json file

"ConnectionStrings": {

"ShopBridgeStoreDb": "Server=.;Database=ShopBridgeStore;Integrated Security= True"

}