

# PRODUCT API

## User Manual



Author

**Prashanth Kumar**

## Contents

Product API .....	2
Technology Stack .....	2
Setup & Execution.....	2
BDD Testing.....	3
API Operations .....	4
GET: /api/Product .....	4
POST: /api/Product .....	5
DELETE: /api/Product.....	6
GET: /api/Product/{productId} .....	7

## Product API

This section of the document describes the Product API's, how to run part and its various operations.

### Technology Stack

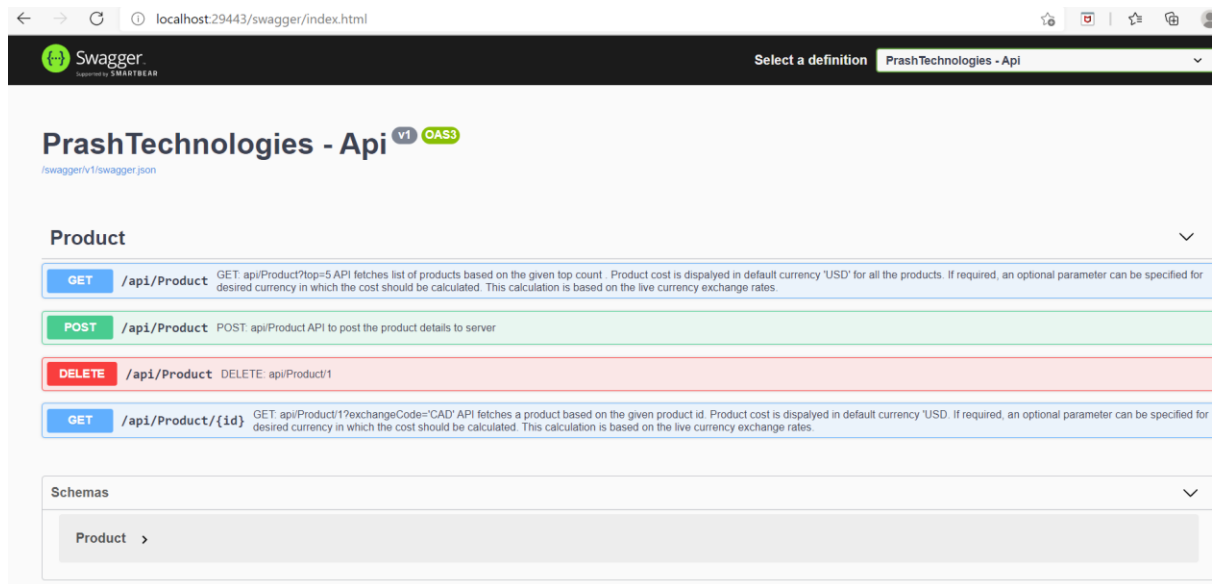
1. Code Editor: Visual Studio 2019 Community
2. Template: ASP.NET Core Web API
3. API Documentation: Swagger
4. Database: SQL Express
5. ORM tool: Dapper
6. Dependency Injection: ASP.NET Core component
7. Framework: .NET Core 3.1
8. Programming Language: C#
9. Other Libraries:
  - Newtonsoft.JSON
10. Unit Test Framework: XUnit
11. BDD Framework: SpecFlow
12. Third party services: <http://api.currencylayer.com/live>

### Setup & Execution

- Download the code from public repository -  
<https://github.com/raoprashanth007/PrashTechnologies.git>
- Open the folder PrashTechnologies.Api\DBScripts and run the scripts in your local database
  - dbo.Products.sql
  - dbo.Stats.sql
- Locate the file appsettings.json (PrashTechnologies.Api\appsettings.json) and modify the ConnectionStrings section to reflect your database instance

```
"ConnectionStrings": {  
  "DefaultConnection": "Data Source=localhost\\SQLEXPRESS;Initial Catalog=master;Integrated Security=True;MultipleActiveResultSets=True"  
}
```

- Build the solution 'PrashTechnologies.Api' and run it
- The API documentation can be accessed from the URL: [Swagger UI](#)
- API Endpoint: <localhost:29443/api/product>



## BDD Testing

SpecFlow and xunit frameworks are used for behavior testing of the API's. As of now, only one test case is written for POST functionality. This can be extended further.

Spec:

```

1 Feature: Product
2     POST Product API (POST: /api/Product)
3
4 @AddProduct
5 Scenario: Add Product
6     Given I creat a new product(<Name>,<CurrentCost>,<Description>)
7     And ModelState is correct
8     Then the system sgould return<StatusCode>
9
10 Examples:
11     | Name | CurrentCost | Description |
12     | Product-Spec | 250.00 | new product |
  
```

Result:

Test	Duration	Traits	Error Message
PrashTechnologies.Api.Tests (1)	1.8 sec		
PrashTechnologies.Api.Tests (1)	1.8 sec		
ProductFeature (1)	1.8 sec		
Add Product(name: "Product-Spec", currentCost: "250.00", description: "new product", exampleTags: [])	1.8 sec	Category [...]	

## API Operations

The details of various operations have been described in the Swagger documentation. However, the below section highlights the same.

GET: /api/Product

Request:

GET

/api/Product

GET: api/Product?top=5 API fetches list of products based on the given top count. Product cost is displayed in default currency 'USD' for all the products. If required, an optional parameter can be specified for desired currency in which the cost should be calculated. This calculation is based on the live currency exchange rates.

Try it out

Parameters

Name	Description
top	number of rows to be dispalyed
integer(int32)	
(query)	Default value : 5
	<input type="text" value="5"/>
exchangeCode	desired currency. suported currencies are: USD, GBP, CAD, EUR
string	
(query)	<input type="text" value="exchangeCode - desired currency. suported c"/>

Responses

Code	Description	Links
200	Success	No links

Response:

Curl

```
curl -X 'GET' \
  'http://localhost:29443/api/Product?top=5' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:29443/api/Product?top=5
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>[   {     "name": "Product 1",     "code": "cd157b87-1fea-42a5-8791-d367e25a05cc",     "clicks": 49,     "description": "",     "currentCost": 50,     "exchangeCode": "USD"   },   {     "name": "Product 8",     "code": "e8eb78ed-641b-41b1-acfd-b2e834f9ef15",     "clicks": 23,     "description": "",     "currentCost": 60,     "exchangeCode": "USD"   },   {     "name": "Product 10",     "code": "2a60ae78-bbcf-4984-8116-f6a2c6e13cf0",     "clicks": 18,     "description": "",     "currentCost": 60,     "exchangeCode": "USD"   },   {     "name": "Product 9",     "code": "901e6e43x-2605-4800-7c-44600-0-45-7",     "clicks": 10,     "description": "",     "currentCost": 50,     "exchangeCode": "USD"   } ]</pre></div><div>Response headers</div></div>

Download

## POST: /api/Product

## Request:

**POST** /api/Product POST: api/Product API to post the product details to server

Try it out

Parameters

No parameters

Request body

application/json

Product details

Example Value | Schema

```
{
  "name": "string",
  "description": "string",
  "currentCost": 0,
  "exchangeCode": "string"
}
```

Responses

Code	Description	Links
200	Success	No links

## Response:

Curl

```
curl -X 'POST' \
  'http://localhost:29443/api/Product' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Product 88",
    "description": "string",
    "currentCost": 60
  }'
```

Request URL

http://localhost:29443/api/Product

Server response

Code	Details
200	<div>Response body</div> <div>1</div> <div>Download</div> <div>Response headers</div> <div>content-type: application/json; charset=utf-8 date: Sun, 09 May 2021 06:41:22 GMT server: Microsoft-IIS/10.0 transfer-encoding: chunked x-powered-by: ASP.NET</div>

## DELETE: /api/Product

## Request:

**DELETE** /api/Product DELETE: api/Product/1

Try it out

Name	Description
id	integer(\$int32)Product id to be deleted
(query)	

id - product id to be deleted

**Responses**

Code	Description	Links
200	Success	No links

## Response:

Curl

```
curl -X 'DELETE' \
'http://localhost:29443/api/Product?id=1' \
-H 'accept: */*'
```

Request URL

```
http://localhost:29443/api/Product?id=1
```

Server response

Code	Details
200	<div><div>Response body</div><div>1</div><div>Download</div></div> <div><div>Response headers</div><div>content-type: application/json; charset=utf-8 date: Sun, 09 May 2021 06:44:20 GMT server: Microsoft-IIS/10.0 transfer-encoding: chunked x-powered-by: ASP.NET</div></div>

**Responses**

Code	Description	Links
200	Success	No links

GET: /api/Product/{productId}

Request:

GET /api/Product/{id} GET: api/Product/?exchangeCode='CAD' API fetches a product based on the given product id. Product cost is displayed in default currency 'USD'. If required, an optional parameter can be specified for desired currency in which the cost should be calculated. This calculation is based on the live currency exchange rates.

**Parameters** [Try it out](#)

Name	Description
<b>id</b> <span style="color: red;">*</span> <small>required</small> integer(int32) (path)	product id
<b>exchangeCode</b> string (query)	desired currency. supported currencies are: USD, GBP, CAD, EUR

**Responses**

Code	Description	Links
200	Success	No links

Response:

Curl

```
curl -X 'GET' \
  'http://localhost:29443/api/Product/2' \
  -H 'accept: */*'
```

Request URL

http://localhost:29443/api/Product/2

Server response

Code	Details
200	<p>Response body</p> <pre>{   "name": "Product 2",   "code": "528c3f63-a30b-4853-817f-280e076a9534",   "title": null,   "description": "",   "currentCost": 55,   "exchangeCode": "USD" }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Sun, 09 May 2022 06:46:15 GMT server: Microsoft-IIS/10.0 transfer-encoding: chunked x-powered-by: ASP.NET</pre>

**Responses**

Code	Description	Links
200	Success	No links