NET 5 Web API CRUD with Angular and Entity Framework

we will create a [**.NET 5**](https://www.sharepointcafe.net/2021/02/whats-new-in-dotnet-5.html) Web API and Angular application to perform CRUD operation with the help of Entity Framework Core.

To demonstrate CRUD operation with .NET 5 Web API, we will use sample Product data.

What is .NET Web API?

[**.Net Web API**](https://www.sharepointcafe.net/2016/09/what-is-web-api-when-should-we-use-in.html) is a framework for building, consuming HTTP based service. The advantage of Web API is that it can be consumed by a wide range of clients like a web browser and mobile applications.

Pre-requisites

* .NET 5 Framework
* Visual Studio 2019
* Angular 7 and above
* SQL Server 2008 and above

Database Design

**Create Table**

CREATE TABLE [dbo].[Product\_Master](

[Id] [int] IDENTITY(1,1) NOT NULL,

[ProductName] [varchar](250) NULL,

[ProductDescription] [varchar](250) NULL,

[ProductCost] [decimal](18, 0) NULL,

[Stock] [int] NULL,

CONSTRAINT [PK\_Product\_Master] PRIMARY KEY CLUSTERED

(

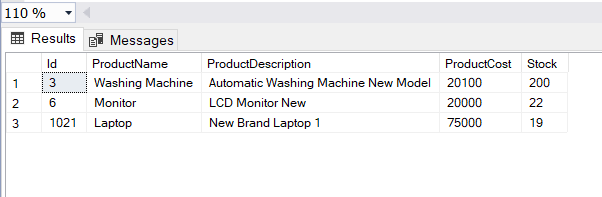
[Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

Add few records



What is .NET 5?

Dot Net 5 or [.NET 5](https://www.sharepointcafe.net/2021/02/whats-new-in-dotnet-5.html) is one of the major release after .NET Core 3.1. It was released in November, 2020.

Microsoft, dropped the word “Core” from the name to indicate that this is the future of .NET. It supports more platforms and apps than [**.NET Core**](https://www.sharepointcafe.net/dot-net-core-tutorial) and .NET Framework.

Create .NET 5 API Project in Visual Studio 2019

Let’s create a .NET 5 Web API project in Visual Studio 2019. There is no big issue in creating project in Visual Studio 2019. Follow below steps –

* Open Visual Studio 2019.
* Click on Create a new Project.
* Select ASP.Net Core Web API from Project template. Click Next
* Give a name to your project. Click Next.
* Select .NET 5.0 as Target framework.

[**How to create .NET 5 Web API in Visual Studio 2019?**](https://www.sharepointcafe.net/2021/03/build-net5-web-api-in-visual-studio-2019.html)

Add a Model class in your project.

**Model Class (Product.cs)**

public class Product

{

[Key]

public int Id { get; set; }

public string ProductName { get; set; }

public string ProductDescription { get; set; }

public decimal ProductCost { get; set; }

public int Stock { get; set; }

}

Install packages related to Entity Framework

You need to install few packages. You can do this either via Packager manager console or from Nuget Packager Manager Wizard.

* Microsoft.EntityFrameworkCore
* Microsoft.EntityFrameworkCore.SqlServer (Install-Package Microsoft.EntityFrameworkCore.SqlServer)
* Microsoft.EntityFrameworkCore.Tools (Install-Package Microsoft.EntityFrameworkCore.Tools)

**Add Data Context Class**

MyDataContext.cs File

public class MyDataContext:DbContext

{

public MyDataContext(DbContextOptions<MyDataContext> options) : base(options)

{

}

public DbSet<Product> Products { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Product>().ToTable("Product\_Master");

}

}

Add Database Connection String

In appsettings.json file, add connection string

"ConnectionStrings": {

"DefaultConnection": "Data Source=localhost\\sqlexpress;Initial Catalog=ProductDataBase;Trusted\_Connection=True;"

}

Get Database Connection String in Startup.cs file

Add below line in ConfigureServices method in startup.cs file

services.AddDbContext<MyDataContext>(options =>

options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

Allow Origins and Header

Add below line in Configure method in startup.cs file. This will help us to avoid ‘CORS’ related issue in browser.

app.UseCors(options => options.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod());

Add a Controller class in your project. (ProductController.cs)

public class ProductController : ControllerBase

{

private readonly MyDataContext \_context;

public ProductController(MyDataContext context)

{

\_context = context;

}

[HttpGet]

[Route("List")]

public async Task<ActionResult<IEnumerable<Product>>> Get()

{

return await \_context.Products.ToListAsync();

}

[HttpGet]

[Route("Details")]

public async Task<ActionResult<Product>> Get(int id)

{

var data = await \_context.Products.FindAsync(id);

return data;

}

[HttpPost]

//public async void Post(Product product)

[Route("CreateRecord")]

public async Task<ActionResult<Product>> POST(Product product)

{

\_context.Products.Add(product);

await \_context.SaveChangesAsync();

return CreatedAtAction(nameof(Get), new { id = product.Id }, product);

}

// PUT api/<ProductController>/5

[HttpPut("{id}")]

public void Put(int id, [FromBody] string value)

{

}

[HttpPost]

[Route("DeleteProduct")]

public async Task<ActionResult<IEnumerable<Product>>> Delete(int id)

{

var product = await \_context.Products.FindAsync(id);

if (product == null)

{

return NotFound();

}

\_context.Products.Remove(product);

await \_context.SaveChangesAsync();

return await \_context.Products.ToListAsync();

}

[HttpPost]

[Route("UpdateProduct")]

public async Task<ActionResult<IEnumerable<Product>>> Update(int id, Product product)

{

if (id != product.Id)

{

return BadRequest();

}

var productData = await \_context.Products.FindAsync(id);

if (productData == null)

{

return NotFound();

}

productData.ProductCost = product.ProductCost;

productData.ProductDescription = product.ProductDescription;

productData.ProductName = product.ProductName;

productData.Stock = product.Stock;

await \_context.SaveChangesAsync();

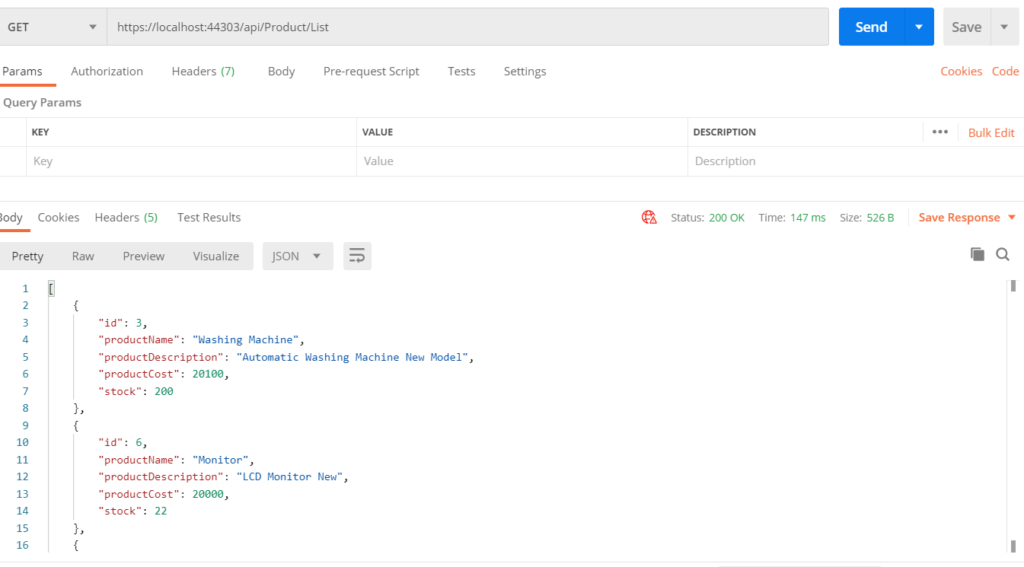
return await \_context.Products.ToListAsync();

}

}

Test .NET 5 Web API with POSTMAN Tool

Once you have created .NET 5 Web API, you need to test it before you integrate with an Angular application.



Create Angular Project

Now, create an Angular project with below command.

ng new Net5Demo

Model File

export class Products {

id: number;

productName: string;

productCost: number;

productDescription: string;

stock: number;

}

Service File

Add a service file with below command

ng g s net5-service

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { Observable } from 'rxjs';

import { Products } from './Products';

@Injectable({

providedIn: 'root'

})

export class Net5ServiceService {

url = 'https://localhost:44303/api/Product/';

constructor(private http: HttpClient) { }

getProductList(): Observable<Products[]> {

return this.http.get<Products[]>(this.url + 'List');

}

postProductData(productData: Products): Observable<Products> {

const httpHeaders = { headers:new HttpHeaders({'Content-Type': 'application/json'}) };

return this.http.post<Products>(this.url + 'CreateRecord', productData, httpHeaders);

}

updateProduct(product: Products): Observable<Products> {

const httpHeaders = { headers:new HttpHeaders({'Content-Type': 'application/json'}) };

return this.http.post<Products>(this.url + 'UpdateProduct?id=' + product.id, product, httpHeaders);

}

deleteProductById(id: number): Observable<number> {

return this.http.post<number>(this.url + 'DeleteProduct?id=' + id, null);

}

getProductDetailsById(id: string): Observable<Products> {

return this.http.get<Products>(this.url + 'Details?id=' + id);

}

}

Component file

Let’s create component file with below command. (**net5-apicall is the component file**)

ng g c net5-apicall

<form class="form-horizontal" [formGroup]="productForm">

<h1>Welcome to Angular CRUD with .NET 5</h1>

<div class="form-group">

<label class="control-label col-sm-2" for="pwd">Name of Product :</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="txtProductName" formControlName="productName"

placeholder="Name of Product">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="pwd">Cost of Product :</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="txtProductCost" formControlName="productCost" placeholder="Cost of Product">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="pwd">Product Description :</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="txtProductDescription" formControlName="productDescription"

placeholder="Product Description">

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="pwd"># of Stock Available :</label>

<div class="col-sm-2">

<input type="text" class="form-control" id="txtStock" formControlName="stock" placeholder="Stock Available">

</div>

</div>

<div class="form-group">

<div class="container">

<div class="row">

<div class="col-sm">

<button type="submit" class="btn btn-primary" (click)="PostProduct(productForm.value)">Submit</button>

</div>

<div class="col-sm">

<button type="submit" class="btn btn-primary" (click)="UpdateProduct(productForm.value)">Update</button>

</div>

</div>

</div>

</div>

<div>

<div class="alert alert-info"><b>Product List</b></div>

<div class="table-responsive">

<table class="table">

<tr>

<th>Product Name</th>

<th>Cost</th>

<th># of Stock</th>

<th>Description</th>

<th>Action</th>

</tr>

<tr \*ngFor="let prd of ProductList | async">

<td>{{prd.productName}}</td>

<td>{{prd.productCost}}</td>

<td>{{prd.stock}}</td>

<td>{{prd.productDescription}}</td>

<td><button type="button" matTooltip="Click Edit Button" (click)='ProductDetailsToEdit(prd.id)'>Edit</button>

|

<button type="button" matTooltip="Click Delete Button" (click)="DeleteProduct(prd.id)">Delete</button>

</td>

</tr>

</table>

</div>

</div>

</form>

Component Type Script File

import { HttpClient } from '@angular/common/http';

import { Component, OnInit } from '@angular/core';

import { FormBuilder, Validators } from '@angular/forms';

import { Observable } from 'rxjs';

import { Net5ServiceService } from '../net5-service.service';

import { Products } from '../Products';

@Component({

selector: 'app-net5-apicall',

templateUrl: './net5-apicall.component.html',

styleUrls: ['./net5-apicall.component.css']

})

export class Net5APICallComponent implements OnInit {

ProductList: Observable<Products[]>;

ProductList1: Observable<Products[]>;

productForm: any;

massage = "";

prodCategory = "";

productId = 0;

constructor(private formbulider: FormBuilder, private httpClient: HttpClient, private productService: Net5ServiceService) { }

ngOnInit() {

this.prodCategory = "0";

this.productForm = this.formbulider.group({

productName: ['', [Validators.required]],

productCost: ['', [Validators.required]],

productDescription: ['', [Validators.required]],

stock: ['', [Validators.required]]

});

this.getProductList();

}

getProductList() {

this.ProductList1 = this.productService.getProductList();

this.ProductList = this.ProductList1;

}

PostProduct(product: Products) {

debugger;

const product\_Master = this.productForm.value;

this.productService.postProductData(product\_Master).subscribe(

() => {

this.massage = 'Data Saved Successfully';

this.getProductList();

}

);

}

ProductDetailsToEdit(id: string) {

debugger;

this.productService.getProductDetailsById(id).subscribe(productResult => {

this.productId = productResult.id;

this.productForm.controls['productName'].setValue(productResult.productName);

this.productForm.controls['productCost'].setValue(productResult.productCost);

this.productForm.controls['productDescription'].setValue(productResult.productDescription);

this.productForm.controls['stock'].setValue(productResult.stock);

// this.productForm.controls['product\_category'].setValue(productResult.productCost);

});

}

UpdateProduct(product: Products) {

debugger;

product.id = this.productId;

const product\_Master = this.productForm.value;

this.productService.updateProduct(product\_Master).subscribe(() => {

this.massage = 'Record Updated Successfully';

this.getProductList();

});

}

DeleteProduct(id: number) {

if (confirm('Do you want to delete this product?')) {

this.productService.deleteProductById(id).subscribe(() => {

this.getProductList();

});

}

}

}

Add Routing

const routes: Routes = [

{path: 'crud', component: Net5APICallComponent}

];

Import HttpClientModule, FormsModule and ReactiveFormsModule in app.module.ts file

import {HttpClientModule} from '@angular/common/http';

import { FormsModule, ReactiveFormsModule } from '@angular/forms';

Add modules in imports array in app.module.ts file

imports: [

BrowserModule,

AppRoutingModule,

HttpClientModule,

FormsModule,

ReactiveFormsModule

],

Change app.component.html file

Remove everything from app.component.html file and put below line.

<router-outlet></router-outlet>

Run your Project

Compile your project with below command

ng serve

**Your screen should look like this.**

