1. Open visual studio
2. Create a new project
3. Select ASP.Net core web API -> next
4. Create a new project name & folder location -> next

* Press create

After project Creation

1. Create a new controller -> right click on controller ->add

->controller ->add -> MVC Controller – Empty -> in Name (anynameController.cs) ->next

1. (anynameController.cs) inside
2. Projectname -> new folder ->Model
3. Model -> Add -> Class -> name.cs ->add ->

public int id {get ; set;}

namecontroller.cs

[HttpGet("school/Getstudentdetails")]

public JsonResult Getstudentdetails()

{

List<student> List = new List<student>();

List.Add(new student { Id = 1, Name = "a" });

List.Add(new student { Id = 2, Name = "b" });

List.Add(new student { Id = 3, Name = "c" });

return Json(new {stuList = List});

}

Run

For post and all method

using Microsoft.AspNetCore.Mvc;

using Microsoft.Data.SqlClient;

using new1.Model;

using System.Data;

using System.Collections.Generic;

namespace new1.Controllers

{

public class SchoolController : Controller

{

private readonly IConfiguration \_configuration;

public SchoolController(IConfiguration Configuration)

{

\_configuration = Configuration;

}

public IActionResult Index()

{

return View();

}

[HttpGet("school/Getstudentdetails")]

public JsonResult Getstudentdetails()

{

List<student> List = new List<student>();

List.Add(new student { Id = 1, Name = "a" });

List.Add(new student { Id = 2, Name = "b" });

List.Add(new student { Id = 3, Name = "c" });

return Json(new { stuList = List });

}

[HttpGet("school/GetEmployee")]

public JsonResult GetEmployee()

{

List<Employee> List = new List<Employee>();

List.Add(new Employee { EmployeeId = 1, Name = "a", Age = 20 });

List.Add(new Employee { EmployeeId = 2, Name = "b", Age = 21 });

List.Add(new Employee { EmployeeId = 3, Name = "c", Age = 23 });

return Json(new { stuList = List });

}

[HttpGet("Ecom/GetCategoryDetails")]

public JsonResult GetCategoryDetails()

{

string constr = \_configuration.GetConnectionString("con");

DataTable dataTable = new DataTable();

List<category> list = new List<category>();

SqlConnection con = new SqlConnection(constr);

SqlCommand sqlCmd = new SqlCommand("CategorySP", con);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@Action", "GetCategory");

SqlDataAdapter adapter = new SqlDataAdapter(sqlCmd);

adapter.Fill(dataTable);

con.Close();

if (dataTable.Rows.Count > 0)

{

for (int i = 0; i < dataTable.Rows.Count; i++)

{

list.Add(new category

{

categoryId = int.Parse(dataTable.Rows[i]["CategoryId"].ToString()),

categoryName = dataTable.Rows[i]["Category"].ToString()

});

}

}

return Json(new { catList = list });

}

[HttpGet("Ecom/GetSpecificCategoryDetails")]

public JsonResult GetSpecificCategoryDetails(int catId)

{

string constr = \_configuration.GetConnectionString("con");

DataTable dataTable = new DataTable();

List<category> list = new List<category>();

SqlConnection con = new SqlConnection(constr);

SqlCommand sqlCmd = new SqlCommand("CategorySP", con);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@Action", "GetSpecificCategoryProducts");

sqlCmd.Parameters.AddWithValue("@categoryId", catId);

SqlDataAdapter adapter = new SqlDataAdapter(sqlCmd);

adapter.Fill(dataTable);

con.Close();

if (dataTable.Rows.Count > 0)

{

for (int i = 0; i < dataTable.Rows.Count; i++)

{

list.Add(new category

{

categoryId = int.Parse(dataTable.Rows[i]["CategoryId"].ToString()),

categoryName = dataTable.Rows[i]["Category"].ToString()

});

}

}

return Json(new { catList = list });

}

[HttpPost("Test/PostMethod")]

public JsonResult PostValue([FromBody] category c)

{

string constr = \_configuration.GetConnectionString("con");

SqlConnection con = new SqlConnection(constr);

con.Open();

SqlCommand sqlCmd = new SqlCommand("CategorySP", con);

sqlCmd.CommandType = CommandType.StoredProcedure;

sqlCmd.Parameters.AddWithValue("@Action", "InsertCategory");

sqlCmd.Parameters.AddWithValue("@category", c.categoryName);

string str = sqlCmd.ExecuteScalar().ToString();

con.Close();

if(str == "1")

{

return Json(new { success = true });

}

else

{

return Json(new { success = false });

}

}

}

}

App.setting

"AllowedHosts": "\*",

"ConnectionStrings": {

"con": "Server=DESKTOP-E7FP35U\\SQLEXPRESS;Database=ECOM1;Trusted\_Connection=True;TrustServerCertificate=Yes"

}

Model.cs

public class category

{

public int categoryId { get; set; }

public string categoryName { get; set; }

}