**Using ADO.Net in Asp.net core Web Api**

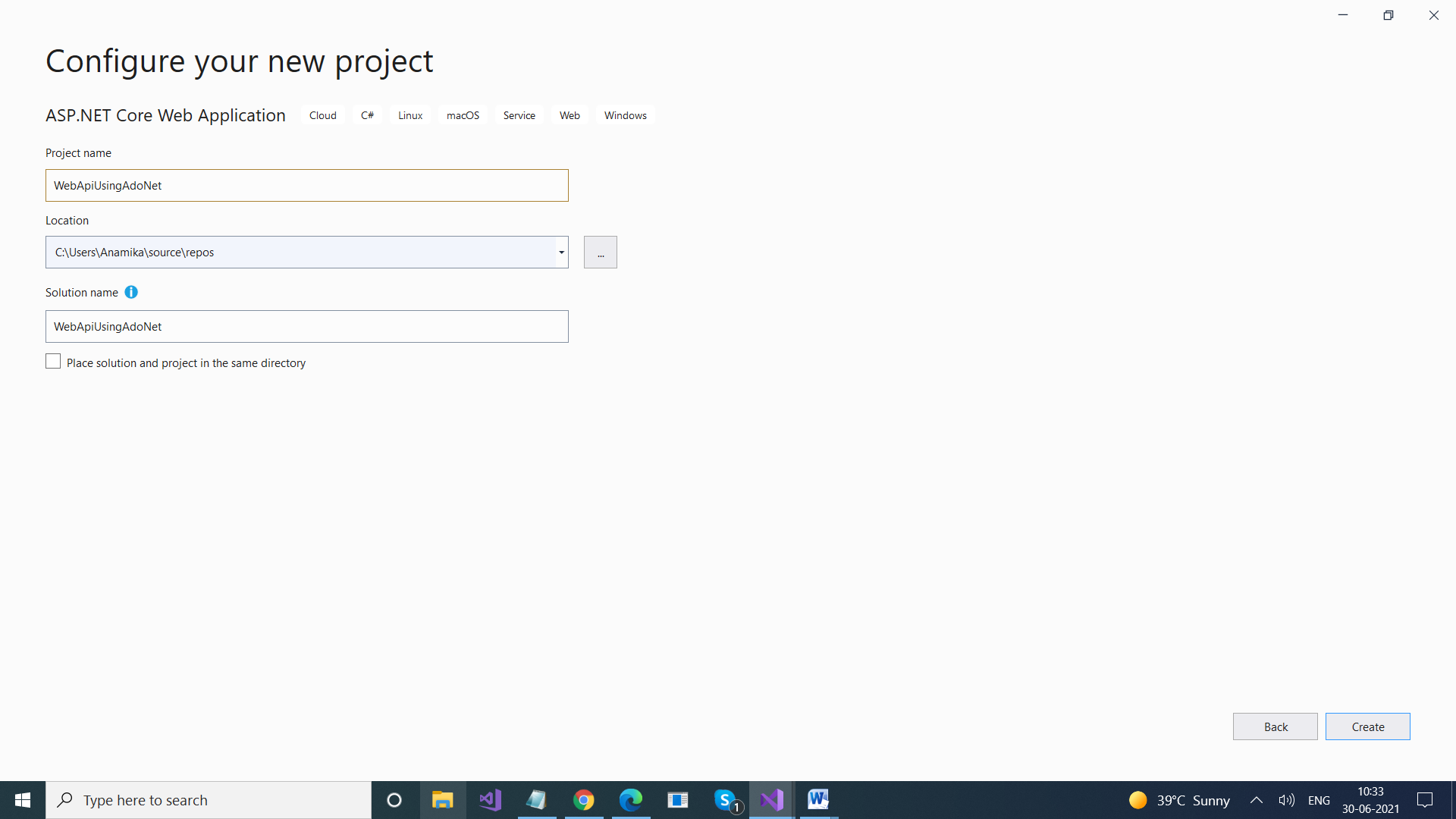
Whenever we have to connect FE with BE (DataSource)

We can use two different technologies

1. ADO.Net
2. Entity Framework

Entity Framework : ORM > There is a mapping between objects & tables

Code First , Database Approach



Create table in Database

create database PracticeWebApi

Use PracticeWebApi

Create table users (Id int primary key, Name varchar(20) not null, EmailId varchar(20), Mobile varchar(10) not null, Address varchar(50), IsActive bit)

insert into Users values(1,'Ajay','ajay@yahho.com','90909098','Delhi',1)

**UserModel Class**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Serialization;

using System.Threading.Tasks;

namespace WebApplication16.Models

{

[DataContract]

public class UsersModel

{

[DataMember(Name = "Id")]

public int Id { get; set; }

[DataMember(Name = "Name")]

public string Name { get; set; }

[DataMember(Name = "EmailId")]

public string EmailId { get; set; }

[DataMember(Name = "Mobile")]

public string Mobile { get; set; }

[DataMember(Name = "Address")]

public string Address { get; set; }

[DataMember(Name = "IsActive")]

public bool IsActive { get; set; }

}

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

**"UserConnectionString": "server=LAPTOP-53S2KQS8;database=PracticeWebApi;integrated security=true",**

"StudentDBContext": "server=LAPTOP-53S2KQS8;database=BookStore;integrated security=true"

},

"Jwt": {

"Key": "ThisismySecretKey",

"Issuer": "Test.com"

}

}

--------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using WebApplication16.Models;

using System.Data;

using System.Data.SqlClient;

using Microsoft.Extensions.Configuration;

using Microsoft.Data.SqlClient;

namespace WebApplication16.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class EmployeesADOController : ControllerBase

{

IConfiguration \_config;

public EmployeesADOController(IConfiguration config)

{

\_config = config;

}

// GET: api/Employees

[HttpGet]

[Route("GetUsers")]

public List<UsersModel> GetUsers()

{

return LoadUsers().ToList();

}

[HttpGet]

[Route("GetUserById")]

public UsersModel GetUserById(int id)

{

return LoadUsers().Where(x => x.Id == id).FirstOrDefault();

}

[HttpPost]

[Route("InsertUser")]

public IActionResult InsertUser(UsersModel user)

{

SqlConnection connection = new SqlConnection(\_config.GetConnectionString("UserConnectionString"));

SqlCommand cmd = new SqlCommand("Insert into users(id, name, emailid, mobile,address,isactive) values (@id, @name,@emailid,@mobile,@address,@isactive)", connection);

//string Query = "Insert into users(id, name, emailid, mobile,address,isactive) values (user.id + ','" + user.Name + ",'" + user.EmailId + ",'" + user.Mobile + ",'" + user.Address + ",' + user.IsActive)";

cmd.Parameters.AddWithValue("@id", user.Id);

cmd.Parameters.AddWithValue("@name", user.Name);

cmd.Parameters.AddWithValue("@address", user.Address);

cmd.Parameters.AddWithValue("@mobile", user.Mobile);

cmd.Parameters.AddWithValue("@emailid", user.EmailId);

cmd.Parameters.AddWithValue("@isactive", user.IsActive);

// SqlCommand cmd = new SqlCommand(Query, connection);

connection.Open();

cmd.ExecuteNonQuery();

connection.Close();

return Ok(user);

}

[HttpPut("{id}")]

[Route("EditUser")]

public IActionResult PutUser(int id, UsersModel user)

{

SqlConnection connection = new SqlConnection(\_config.GetConnectionString("UserConnectionString"));

string Quey = "Update users set name ='" + user.Name + "', address = '" + user.Address + "' where id = "+ id ;

SqlCommand cmd = new SqlCommand(Quey, connection);

// SqlCommand cmd = new SqlCommand(Query, connection);

//string qry = "Select \* from users WHERE [Name]='" + pLoginName + "' AND [Pwd]='" + pPassword + "'";

connection.Open();

cmd.ExecuteNonQuery();

connection.Close();

return Ok(user);

}

[HttpDelete("{id}")]

[Route("DeleteUser")]

public IActionResult Delete(int id)

{

SqlConnection connection = new SqlConnection(\_config.GetConnectionString("UserConnectionString"));

SqlCommand cmd = new SqlCommand("'Delete from users where id= '+ id ",connection);

connection.Open();

cmd.ExecuteNonQuery();

connection.Close();

return Ok();

}

private List<UsersModel> LoadUsers()

{

List<UsersModel> users = new List<UsersModel>();

SqlConnection connection = new SqlConnection(\_config.GetConnectionString("UserConnectionString"));

SqlCommand cmd = new SqlCommand("Select \* from users", connection);

SqlDataAdapter ada = new SqlDataAdapter(cmd);

DataTable dataTable = new DataTable();

ada.Fill(dataTable);

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

{

UsersModel user = new UsersModel();

user.Id = (int)dataTable.Rows[i]["Id"];

user.Name = dataTable.Rows[i]["Name"].ToString();

user.EmailId = dataTable.Rows[i]["EmailId"].ToString();

user.Address = dataTable.Rows[i]["Address"].ToString();

user.IsActive = (bool)dataTable.Rows[i]["isActive"];

users.Add(user);

}

return users;

}

}

}

-------------------------------

**Attribute Routing**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using System.Data;

using System.Data.SqlClient;

using WebApiUsingAdoNet.Models;

// For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

namespace WebApiUsingAdoNet.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class UsersController : ControllerBase

{

IConfiguration configuration;

public UsersController(IConfiguration \_configuration)

{

configuration = \_configuration;

}

// GET: api/<UsersController>

[HttpGet]

**[Route("GetUsers")]**

**public IActionResult Get()**

{

List<UserModel> users = new List<UserModel>();

users = GetAllUsers();

return Ok(users);

}

// Laod all users & return them

private List<UserModel> GetAllUsers()

{

List<UserModel> users = new List<UserModel>();

SqlConnection connection = new SqlConnection(configuration.GetConnectionString("UserConnectionString"));

SqlCommand command = new SqlCommand("Select \* from users", connection);

SqlDataAdapter ada = new SqlDataAdapter(command);

DataTable dt = new DataTable();

ada.Fill(dt);

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

{

UserModel obj = new UserModel();

obj.Id = (int)dt.Rows[i]["Id"];

obj.Name = dt.Rows[i]["Name"].ToString();

obj.Mobile = dt.Rows[i]["Mobile"].ToString();

obj.Address = dt.Rows[i]["Address"].ToString();

obj.EmailId = dt.Rows[i]["EmailId"].ToString();

obj.IsActive = (bool)dt.Rows[i]["IsActive"];

users.Add(obj);

}

return users;

}

// GET api/<UsersController>/5

**[HttpGet("{id}")]**

**[Route("GetUserById")]**

**public IActionResult Get(int id)**

{

UserModel user = new UserModel();

user = GetAllUsers().Where(x => x.Id == id).FirstOrDefault();

if (user != null)

return Ok(user);

else

return NotFound();

}

// POST api/<UsersController>

**[HttpPost]**

**[Route("InsertRecord")]**

**public IActionResult SaveRecord(UserModel user)**

**{**

SqlConnection connection = new SqlConnection(configuration.GetConnectionString("UserConnectionString"));

//SqlCommand command = new SqlCommand();

//string query = "Insert into users(id, name, address, emailid, mobile, isactive) values (@id, @name, @address, @emailid, @mobile,@isactive)";

//command.CommandText = query;

//command.Connection = connection;

SqlCommand command = new SqlCommand("Insert into users(id, name, address, emailid, mobile, isactive) values (@id, @name, @address, @emailid, @mobile,@isactive)", connection);

command.Parameters.AddWithValue("@id", user.Id);

command.Parameters.AddWithValue("@name", user.Name);

command.Parameters.AddWithValue("@address", user.Address);

command.Parameters.AddWithValue("@emailid", user.EmailId);

command.Parameters.AddWithValue("@mobile", user.Mobile);

command.Parameters.AddWithValue("@isactive", user.IsActive);

connection.Open();

int res = command.ExecuteNonQuery();

connection.Close();

if (res > 0)

return Ok(user);

else

return BadRequest();

}

// PUT api/<UsersController>/5

[HttpPut("{id}")]

[Route("EditUser")]

public ActionResult EditUser(int id,UserModel user)

{

SqlConnection connection = new SqlConnection(configuration.GetConnectionString("UserConnectionString"));

string updateQuery = "Update users set name ='" + user.Name + "', address = '" + user.Address + "' where id = " + id;

SqlCommand command = new SqlCommand(updateQuery, connection);

connection.Open();

int res= command.ExecuteNonQuery();

connection.Close();

if (res > 0)

return Ok(user);

else

return BadRequest();

}

// DELETE api/<UsersController>/5

[HttpDelete("{id}")]

[Route("DeleteUser")]

public ActionResult RemoveUser(int id)

{

SqlConnection connection = new SqlConnection(configuration.GetConnectionString("UserConnectionString"));

string updateQuery = "Delete users where id = " + id;

SqlCommand command = new SqlCommand(updateQuery, connection);

connection.Open();

int res = command.ExecuteNonQuery();

connection.Close();

if (res > 0)

return Ok();

else

return BadRequest();

}

}

}

Using Stored Procedures

**--Get All Records**

**create proc GetUsers**

**As**

**begin**

**select \* from users**

**end**

**--Insert Record**

**create proc InsertUser(@id int , @name varchar(20),**

**@address varchar(50) , @mobile varchar(10), @emailid varchar(50)**

**,@isactive bit)**

**AS**

**Begin**

**insert into users (id, name, Address, Mobile, EmailId, IsActive)**

**values (@id,@name, @address, @mobile, @emailid,@isactive)**

**end**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using System.Data;

using System.Data.SqlClient;

using WebApiUsingAdoNet.Models;

// For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

namespace WebApiUsingAdoNet.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class UsersController : ControllerBase

{

IConfiguration configuration;

public UsersController(IConfiguration \_configuration)

{

configuration = \_configuration;

}

// GET: api/<UsersController>

[HttpGet]

[Route("GetUsers")]

public IActionResult Get()

{

List<UserModel> users = new List<UserModel>();

users = GetAllUsers();

return Ok(users);

}

// Laod all users & return them

private List<UserModel> GetAllUsers()

{

List<UserModel> users = new List<UserModel>();

SqlConnection connection = new SqlConnection(configuration.GetConnectionString("UserConnectionString"));

**// SqlCommand command = new SqlCommand("Select \* from users", connection);**

**SqlCommand command = new SqlCommand("GetUsers", connection);**

**command.CommandType = CommandType.StoredProcedure;**

SqlDataAdapter ada = new SqlDataAdapter(command);

DataTable dt = new DataTable();

ada.Fill(dt);

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

{

UserModel obj = new UserModel();

obj.Id = (int)dt.Rows[i]["Id"];

obj.Name = dt.Rows[i]["Name"].ToString();

obj.Mobile = dt.Rows[i]["Mobile"].ToString();

obj.Address = dt.Rows[i]["Address"].ToString();

obj.EmailId = dt.Rows[i]["EmailId"].ToString();

obj.IsActive = (bool)dt.Rows[i]["IsActive"];

users.Add(obj);

}

return users;

}

// GET api/<UsersController>/5

[HttpGet("{id}")]

[Route("GetUserById")]

public IActionResult Get(int id)

{

UserModel user = new UserModel();

user = GetAllUsers().Where(x => x.Id == id).FirstOrDefault();

if (user != null)

return Ok(user);

else

return NotFound();

}

// POST api/<UsersController>

[HttpPost]

[Route("InsertRecord")]

public IActionResult SaveRecord(UserModel user)

{

SqlConnection connection = new SqlConnection(configuration.GetConnectionString("UserConnectionString"));

//SqlCommand command = new SqlCommand();

//string query = "Insert into users(id, name, address, emailid, mobile, isactive) values (@id, @name, @address, @emailid, @mobile,@isactive)";

//command.CommandText = query;

//command.Connection = connection;

**// SqlCommand command = new SqlCommand("Insert into users(id, name, address, emailid, mobile, isactive) values (@id, @name, @address, @emailid, @mobile,@isactive)", connection);**

**SqlCommand command = new SqlCommand("InsertUser", connection);**

**command.CommandType = CommandType.StoredProcedure;**

command.Parameters.AddWithValue("@id", user.Id);

command.Parameters.AddWithValue("@name", user.Name);

command.Parameters.AddWithValue("@address", user.Address);

command.Parameters.AddWithValue("@mobile", user.Mobile);

command.Parameters.AddWithValue("@emailid", user.EmailId);

command.Parameters.AddWithValue("@isactive", user.IsActive);

connection.Open();

int res = command.ExecuteNonQuery();

connection.Close();

if (res > 0)

return Ok(user);

else

return BadRequest();

}

// PUT api/<UsersController>/5

[HttpPut("{id}")]

[Route("EditUser")]

public ActionResult EditUser(int id,UserModel user)

{

SqlConnection connection = new SqlConnection(configuration.GetConnectionString("UserConnectionString"));

string updateQuery = "Update users set name ='" + user.Name + "', address = '" + user.Address + "' where id = " + id;

SqlCommand command = new SqlCommand(updateQuery, connection);

connection.Open();

int res= command.ExecuteNonQuery();

connection.Close();

if (res > 0)

return Ok(user);

else

return BadRequest();

}

// DELETE api/<UsersController>/5

[HttpDelete("{id}")]

[Route("DeleteUser")]

public ActionResult RemoveUser(int id)

{

SqlConnection connection = new SqlConnection(configuration.GetConnectionString("UserConnectionString"));

string updateQuery = "Delete users where id = " + id;

SqlCommand command = new SqlCommand(updateQuery, connection);

connection.Open();

int res = command.ExecuteNonQuery();

connection.Close();

if (res > 0)

return Ok();

else

return BadRequest();

}

}

}