**916217 GPavan Kumar**

**Stage 4 - Day 75 - Web API – Hands-on**

**EmployeesController.cs**

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 WebApiCorsJwtToken.Models;

namespace WebApiCoreActionExcptionFilter.Controllers

{

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

[ApiController]

//[Authorize(Roles = "POC,Admin")]

public class EmployeeController : ControllerBase

{

private Employee[] employees = new Employee[]

{

new Employee{Id=1,Name="Pavan",Salary=90000,Permanent=true,Department=new Department{Id=1,Name="CDE"},Skills=new List<Skill>{new Skill{Id=1,Name="Python"},new Skill { Id = 2, Name = "Java" } },DateOfBirth=DateTime.Parse("08/04/2000") },

new Employee{Id=2,Name="uday",Salary=90000,Permanent=true,Department=new Department{Id=2,Name="CDE"},Skills=new List<Skill>{new Skill{Id=1,Name="Java"},new Skill { Id = 2, Name = "Python" } },DateOfBirth=DateTime.Parse("07/12/1999") }

};

private IEnumerable<Employee> GetStandardEmployeeList()

{

return employees;

}

// GET: api/Employee

[HttpGet]

[ProducesResponseType(StatusCodes.Status200OK)]

public IActionResult Get()

{

return Ok(GetStandardEmployeeList());

}

// GET: api/Employee/5

[HttpGet("{id}", Name = "Get")]

public string Get(int id)

{

return "value";

}

// POST: api/Employee

[HttpPost]

public void Post([FromBody] string value)

{

}

// PUT: api/Employee/5

[HttpPut("{id}")]

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

{

}

// DELETE: api/ApiWithActions/5

[HttpDelete("{id}")]

public void Delete(int id)

{

}

}

}

**AuthController.cs**

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.IdentityModel.Tokens;

using System.Security.Claims;

using System.IdentityModel.Tokens.Jwt;

using System.Text;

using Microsoft.AspNetCore.Authorization;

using WebApiFourth.Models;

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

namespace WebApiFourth.Controllers

{

[AllowAnonymous]

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

[ApiController]

public class AuthController : ControllerBase

{

[HttpPost]

public IActionResult Post([FromBody] UserModel user)

{

string userRole = "Admin";

string tokenStr = GenerateJSONWebToken(user.UserName, userRole); // Token generated for Admin Role

return new OkObjectResult(new { token = tokenStr });

}

private string GenerateJSONWebToken(string userId, string userRole)

{

var securityKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("mysuperdupersecret"));

var credentials = new SigningCredentials(securityKey, SecurityAlgorithms.HmacSha256);

var claims = new List<Claim>

{

new Claim(ClaimTypes.Role, userRole),

new Claim("UserId", userId)

};

var token = new JwtSecurityToken(

issuer: "mySystem",

audience: "myUsers",

claims: claims,

expires: DateTime.Now.AddMinutes(10),

signingCredentials: credentials);

return new JwtSecurityTokenHandler().WriteToken(token);

}

}

}

**Department.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace WebApiCorsJwtToken.Models

{

public class Department

{

public Department()

{ }

public Department(int DepID, string DepName)

{

this.DepID = DepID;

this.DepName = DepName;

}

public int DepID { get; set; }

public string DepName { get; set; }

}

}

**Employee.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Day2\_Handson.Model;

namespace WebApiCorsJwtToken.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Salary { get; set; }

public bool Permanent { get; set; }

public Department Department { get; set; }

public List<Skill> Skills { get; set; }

public DateTime DateOfBirth { get; set; }

}

}

**Skill.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace WebApiCorsJwtToken.Models{

public class Skill

{

public Skill() { }

public Skill(int skillID, string skillName)

{

this.SkillID = skillID;

this.SkillName = skillName;

}

public int SkillID { get; set; }

public string SkillName { get; set; }

}

}

**SchollDBcontext.cs**

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using WebApiFourth;

namespace WebApiFourth.Models

{

public class SchoolDbContext: DbContext

{

public virtual DbSet<UserModel> UserModels { get; set; }

}

}

**UserModel.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace WebApiFourth.Models

{

public class UserModel

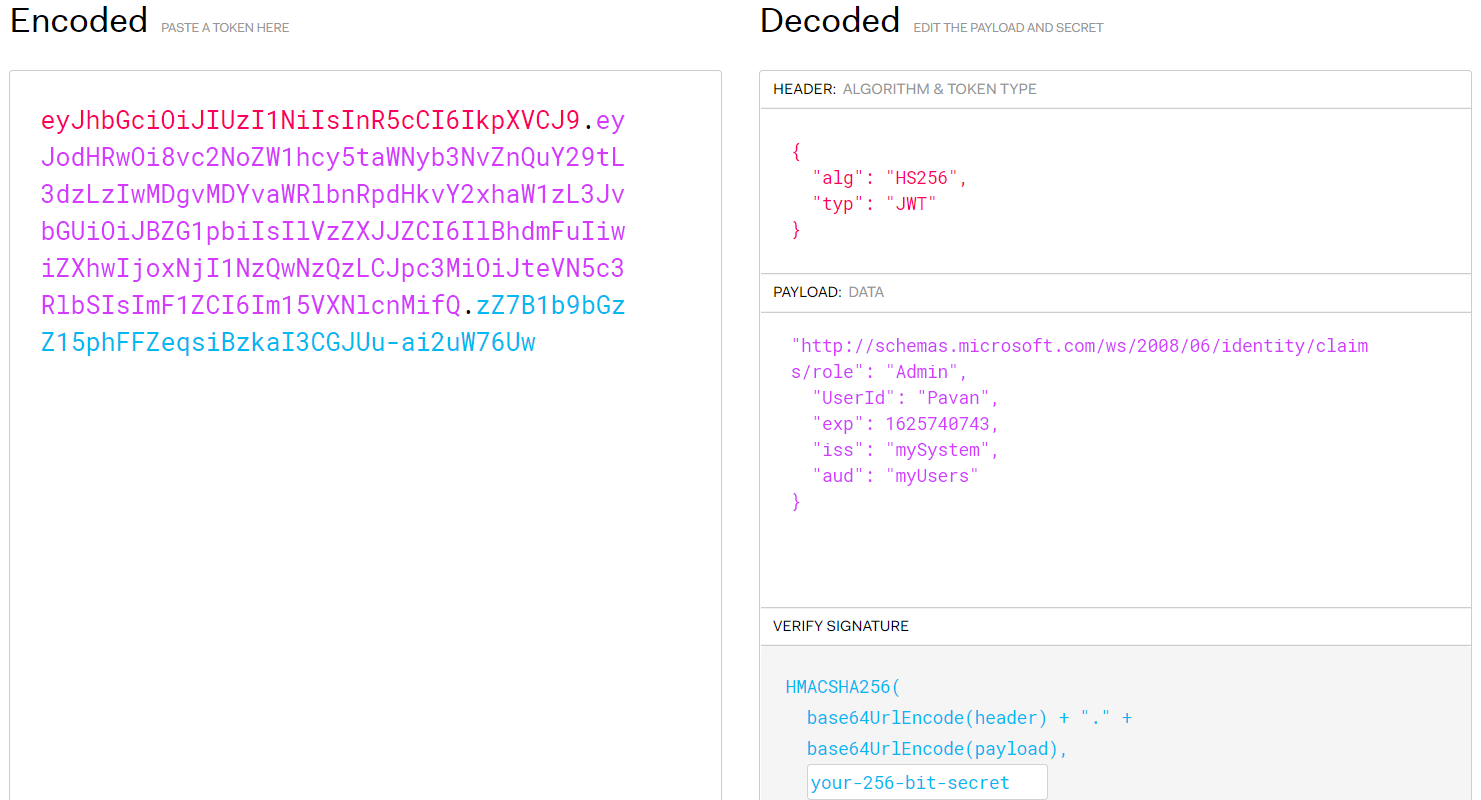
{

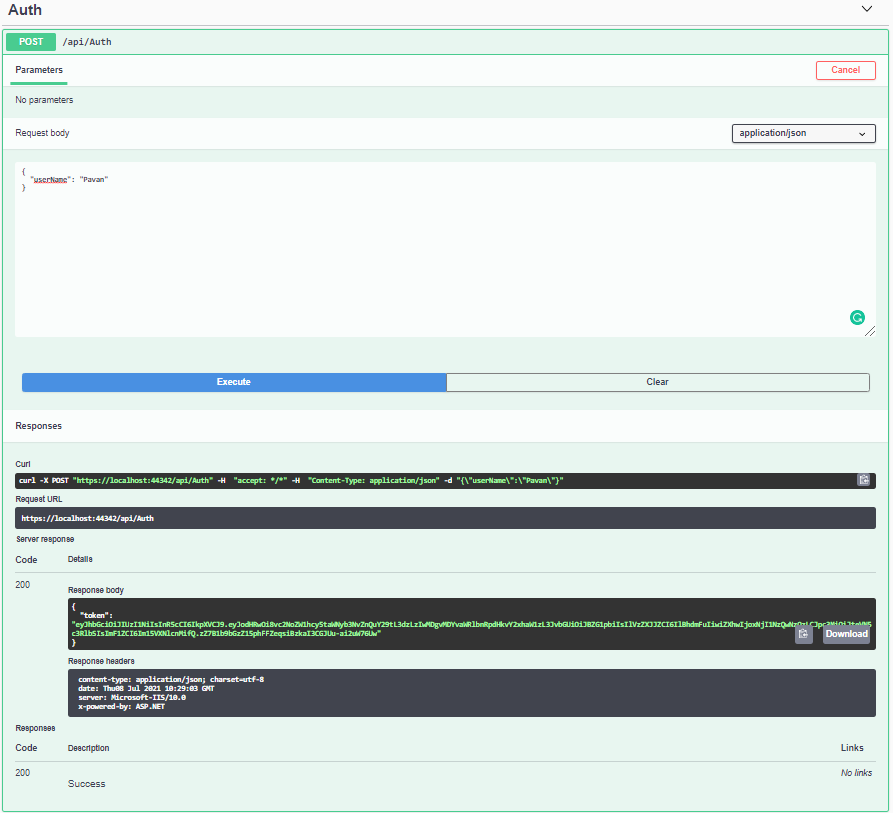
public string UserName { get; set; }

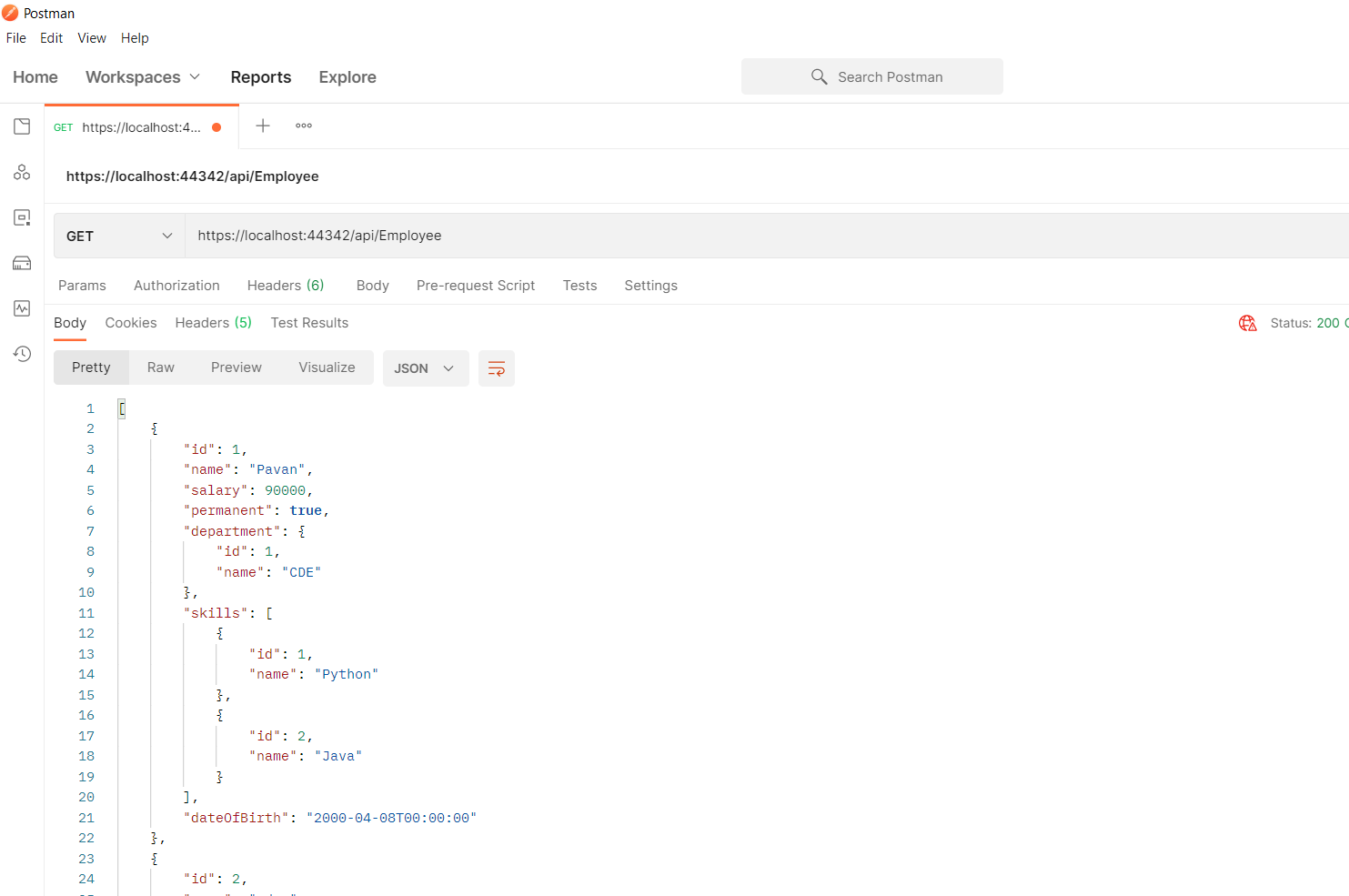
}

}

**Output:**







**Hands on-2**

**EmployeeAPI**

**Models**

**DBcontext**

using System;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity;

using System.Linq;

namespace EmployeeAPI.Models

{

public partial class DBcontext : DbContext

{

public DBcontext()

: base("name=DBcontext")

{

}

public virtual DbSet<Employee> Employees { get; set; }

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

modelBuilder.Entity<Employee>()

.Property(e => e.Name)

.IsUnicode(false);

modelBuilder.Entity<Employee>()

.Property(e => e.Position)

.IsUnicode(false);

}

}

}

**Employee.cs**

namespace EmployeeAPI.Models

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("Employee")]

public partial class Employee

{

public int EmployeeID { get; set; }

[StringLength(50)]

public string Name { get; set; }

[StringLength(50)]

public string Position { get; set; }

public int? Age { get; set; }

public int? Salary { get; set; }

}

}

**EmployeeController.cs**

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

using System.Web.Http.Description;

using EmployeeAPI.Models;

namespace EmployeeAPI.Controllers

{

public class EmployeeController : ApiController

{

private DBcontext db = new DBcontext();

// GET: api/Employee

public IQueryable<Employee> GetEmployees()

{

return db.Employees;

}

// GET: api/Employee/5

[ResponseType(typeof(Employee))]

public IHttpActionResult GetEmployee(int id)

{

Employee employee = db.Employees.Find(id);

if (employee == null)

{

return NotFound();

}

return Ok(employee);

}

// PUT: api/Employee/5

[ResponseType(typeof(void))]

public IHttpActionResult PutEmployee(int id, Employee employee)

{

if (id != employee.EmployeeID)

{

return BadRequest();

}

db.Entry(employee).State = EntityState.Modified;

try

{

db.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

if (!EmployeeExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return StatusCode(HttpStatusCode.NoContent);

}

// POST: api/Employee

[ResponseType(typeof(Employee))]

public IHttpActionResult PostEmployee(Employee employee)

{

db.Employees.Add(employee);

db.SaveChanges();

return CreatedAtRoute("DefaultApi", new { id = employee.EmployeeID }, employee);

}

// DELETE: api/Employee/5

[ResponseType(typeof(Employee))]

public IHttpActionResult DeleteEmployee(int id)

{

Employee employee = db.Employees.Find(id);

if (employee == null)

{

return NotFound();

}

db.Employees.Remove(employee);

db.SaveChanges();

return Ok(employee);

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

private bool EmployeeExists(int id)

{

return db.Employees.Count(e => e.EmployeeID == id) > 0;

}

}

}

**EmployeeMVC**

**Models**

**Employeemodel.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Web;

using System.Web.Mvc;

using Employeemvc.Models;

namespace Employeemvc.Controllers

{

public class EmployeeController : Controller

{

// GET: /Employee/

public ActionResult Index()

{

IEnumerable<Employeemodel> empList;

HttpResponseMessage response = employconnection.WebApiClient.GetAsync("Employee").Result;

empList = response.Content.ReadAsAsync<IEnumerable<Employeemodel>>().Result;

return View(empList);

}

public ActionResult AddOrEdit(int id = 0)

{

if (id == 0)

return View(new Employeemodel());

else

{

HttpResponseMessage response = employconnection.WebApiClient.GetAsync("Employee/" + id.ToString()).Result;

return View(response.Content.ReadAsAsync<Employeemodel>().Result);

}

}

[HttpPost]

public ActionResult AddOrEdit(Employeemodel emp)

{

if (emp.EmployeeID == 0)

{

HttpResponseMessage response = employconnection.WebApiClient.PostAsJsonAsync("Employee", emp).Result;

TempData["SuccessMessage"] = "Saved Successfully";

}

else

{

HttpResponseMessage response = employconnection.WebApiClient.PutAsJsonAsync("Employee/" + emp.EmployeeID, emp).Result;

TempData["SuccessMessage"] = "Updated Successfully";

}

return RedirectToAction("Index");

}

public ActionResult Delete(int id)

{

HttpResponseMessage response = employconnection.WebApiClient.DeleteAsync("Employee/" + id.ToString()).Result;

TempData["SuccessMessage"] = "Deleted Successfully";

return RedirectToAction("Index");

}

}

}

**EmployeeController.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Web;

using System.Web.Mvc;

using Employeemvc.Models;

namespace Employeemvc.Controllers

{

public class EmployeeController : Controller

{

// GET: /Employee/

public ActionResult Index()

{

IEnumerable<Employeemodel> empList;

HttpResponseMessage response = employconnection.WebApiClient.GetAsync("Employee").Result;

empList = response.Content.ReadAsAsync<IEnumerable<Employeemodel>>().Result;

return View(empList);

}

public ActionResult AddOrEdit(int id = 0)

{

if (id == 0)

return View(new Employeemodel());

else

{

HttpResponseMessage response = employconnection.WebApiClient.GetAsync("Employee/" + id.ToString()).Result;

return View(response.Content.ReadAsAsync<Employeemodel>().Result);

}

}

[HttpPost]

public ActionResult AddOrEdit(Employeemodel emp)

{

if (emp.EmployeeID == 0)

{

HttpResponseMessage response = employconnection.WebApiClient.PostAsJsonAsync("Employee", emp).Result;

TempData["SuccessMessage"] = "Saved Successfully";

}

else

{

HttpResponseMessage response = employconnection.WebApiClient.PutAsJsonAsync("Employee/" + emp.EmployeeID, emp).Result;

TempData["SuccessMessage"] = "Updated Successfully";

}

return RedirectToAction("Index");

}

public ActionResult Delete(int id)

{

HttpResponseMessage response = employconnection.WebApiClient.DeleteAsync("Employee/" + id.ToString()).Result;

TempData["SuccessMessage"] = "Deleted Successfully";

return RedirectToAction("Index");

}

}

}

**Index.cshtml**

@model IEnumerable<Employeemvc.Models.Employeemodel>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

<a href="@Url.Action("AddOrEdit","Employee")" class="btn btn-default"><i class="fa fa-plus"></i> Create New</a>

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.EmployeeID)

</th>

<th>

@Html.DisplayNameFor(model => model.Name)

</th>

<th>

@Html.DisplayNameFor(model => model.Position)

</th>

<th>

@Html.DisplayNameFor(model => model.Age)

</th>

<th>

@Html.DisplayNameFor(model => model.Salary)

</th>

<th></th>

</tr>

@foreach (var item in Model)

{

<tr>

<td>

@Html.DisplayFor(modelItem => item.EmployeeID)

</td>

<td>

@Html.DisplayFor(modelItem => item.Name)

</td>

<td>

@Html.DisplayFor(modelItem => item.Position)

</td>

<td>

@Html.DisplayFor(modelItem => item.Age)

</td>

<td>

@Html.DisplayFor(modelItem => item.Salary)

</td>

<td>

<a href="@Url.Action("AddOrEdit", "Employee", new { id = item.EmployeeID})" class="btn btn-default"><i class="fa fa-pencil"></i> Edit</a>

<a onclick="Delete(@item.EmployeeID)" class="btn btn-default"><i class="fa fa-trash"></i> Delete</a>

</td>

</tr>

}

</table>

@section scripts{

<script>

$(function () {

var successMessage = '@TempData["SuccessMessage"]'

if (successMessage != '')

alertify.success(successMessage);

});

function Delete(id) {

alertify.confirm('Web Api CRUD Operations','Are You Sure to Delete this Record ?',function(){

window.location.href = '@Url.Action("Delete","Employee")/'+id;

},null );

}

</script>

}

**AddOrEdit.cshtml**

@model Employeemvc.Models.Employeemodel

@{

ViewBag.Title = "AddOrEdit";

}

<div class="form-body">

@using (Html.BeginForm())

{

@Html.HiddenFor(model => model.EmployeeID)

<div class="form-group">

@Html.LabelFor(model => model.Name)

@Html.EditorFor(model => model.Name)

@Html.ValidationMessageFor(model => model.Name)

</div>

<div class="form-group">

@Html.LabelFor(model => model.Position)

@Html.EditorFor(model => model.Position)

@Html.ValidationMessageFor(model => model.Position)

</div>

<div class="form-group">

@Html.LabelFor(model => model.Age)

@Html.EditorFor(model => model.Age)

@Html.ValidationMessageFor(model => model.Age)

</div>

<div class="form-group">

@Html.LabelFor(model => model.Salary)

@Html.EditorFor(model => model.Salary)

@Html.ValidationMessageFor(model => model.Salary)

</div>

<div class="form-group">

<input type="submit" value="Submit" class="btn button" />

<input type="reset" value="Reset" class="btn button" />

</div>

}

</div>

<div>

@Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

**Connection to api**

**Employeeconnction.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Web;

namespace Employeemvc

{

public class employconnection

{

public static HttpClient WebApiClient = new HttpClient();

static employconnection()

{

WebApiClient.BaseAddress = new Uri("https://localhost:44351/api/");

WebApiClient.DefaultRequestHeaders.Clear();

WebApiClient.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

}

}

}

**Output:**

