```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using PDLERP.Models;
using PDLERP.ServiceInterfaces.Emp;

namespace PDLERP.Controllers
{
    [AllowAnonymous]
    [Route("emp/")]
    //[ApiController]
    public class ApiController : ControllerBase
    {
        private readonly IEmployee _employee;

        public ApiController(IEmployee employee)
        {
            _employee = employee;
        }

        [HttpGet("GetAll")]
        public async Task<IActionResult> GetAll()
        {
            try
            {
                var employees = await _employee.GetAll();

                if (employees.Any())
                    return Ok(employees);
                return NoContent();
            }
            catch (Exception)
            {
                return BadRequest();
            }
        }

        [HttpGet("Find/{id}")]
        public async Task<IActionResult> FindById(int id)
        {
            try
            {
                var employees = await _employee.All();
                var firstOrDefault = employees.FirstOrDefault(e => e.Id.Equals
                  (id));

                if (firstOrDefault == null)
                    return NoContent();

                return Ok(firstOrDefault);
            }
            catch (Exception)
            {
                return BadRequest();
            }
        }

        [HttpPost("Post")]
        //[Route("/Post")]
        public async Task<IActionResult> PostEmployee(Employee employee)
        {
            try
            {
                await _employee.GetInsertedObjByAsync(employee);
                return CreatedAtAction(nameof(FindById), new { id = employee.Id }, employee);
            }
            catch (Exception)
            {
                return BadRequest();
            }
        }

        [HttpPut("Update")]
        public async Task<IActionResult> UpdateEmployee(Employee employee)
        {
            try
            {
                var findByIdAsync = await _employee.FindByIdAsync(employee.Id);

                if (findByIdAsync == null)
                    return NotFound();

                findByIdAsync.FirstName = employee.FirstName;
                findByIdAsync.MiddleName = employee.MiddleName;
                findByIdAsync.LastName = employee.LastName;

                if (await _employee.Update(findByIdAsync))
                    return CreatedAtAction(nameof(FindById), new { id = findByIdAsync.Id }, findByIdAsync);
                return NoContent();
            }
            catch (Exception)
            {
                return BadRequest();
            }
        }

        [HttpDelete("Delete/{id}")]
        public async Task<IActionResult> DeleteEmployee(int id)
        {
            try
            {
                var findByIdAsync = await _employee.FindByIdAsync(id);

                if (findByIdAsync == null)
                    return NotFound();

                await _employee.Delete(findByIdAsync);
                return NoContent();
            }
            catch (Exception)
            {
                return BadRequest();
            }
        }
    }
}

```