**Company Controller**

|  |
| --- |
| using Microsoft.AspNetCore.Mvc;  using System.Threading.Tasks;  using Company.Data.Repositories;  using Company.Data.Models;  using Serilog;  namespace Company.Controllers  {  [ApiController]  [Route("api/[controller]")]  public class CompanyController : ControllerBase  {  private readonly ICompanyRepository \_companyRepository;  private readonly ILogger \_logger;  public CompanyController(ICompanyRepository companyRepository, ILogger logger)  {  \_companyRepository = companyRepository;  \_logger = logger;  }  [HttpGet]  public async Task<IActionResult> GetCompanies()  {  try  {  var companies = await \_companyRepository.GetCompaniesAsync();  return Ok(companies);  }  catch (Exception ex)  {  \_logger.Error(ex, "Error getting companies");  return StatusCode(500, "Error getting companies");  }  }  [HttpGet("{id}")]  public async Task<IActionResult> GetCompany(int id)  {  try  {  var company = await \_companyRepository.GetCompanyAsync(id);  if (company == null)  {  return NotFound();  }  return Ok(company);  }  catch (Exception ex)  {  \_logger.Error(ex, "Error getting company {id}", id);  return StatusCode(500, "Error getting company");  }  }  [HttpPost]  public async Task<IActionResult> CreateCompany(Company company)  {  try  {  await \_companyRepository.CreateCompanyAsync(company);  return CreatedAtAction(nameof(GetCompany), new { id = company.Id }, company);  }  catch (Exception ex)  {  \_logger.Error(ex, "Error creating company");  return StatusCode(500, "Error creating company");  }  }  [HttpPut("{id}")]  public async Task<IActionResult> UpdateCompany(int id, Company company)  {  try  {  await \_companyRepository.UpdateCompanyAsync(id, company);  return Ok(company);  }  catch (Exception ex)  {  \_logger.Error(ex, "Error updating company {id}", id);  return StatusCode(500, "Error updating company");  }  }  [HttpDelete("{id}")]  public async Task<IActionResult> DeleteCompany(int id)  {  try  {  await \_companyRepository.DeleteCompanyAsync(id);  return NoContent();  }  catch (Exception ex)  {  \_logger.Error(ex, "Error deleting company {id}", id);  return StatusCode(500, "Error deleting company");  }  }  }  } |

**Employee Repository**

|  |
| --- |
| using Company.Data.Models;  using System.Collections.Generic;  using System.Threading.Tasks;  namespace Company.Data.Repositories  {  public interface IEmployeeRepository  {  Task<List<Employee>> GetEmployeesAsync();  Task<Employee> GetEmployeeAsync(string employeeCode);  Task CreateEmployeeAsync(Employee employee);  Task UpdateEmployeeAsync(Employee employee);  Task DeleteEmployeeAsync(string employeeCode);  }  public class EmployeeRepository : IEmployeeRepository  {  private readonly DbContext \_context;  public EmployeeRepository(DbContext context)  {  \_context = context;  }  public async Task<List<Employee>> GetEmployeesAsync()  {  return await \_context.Employees.ToListAsync();  }  public async Task<Employee> GetEmployeeAsync(string employeeCode)  {  return await \_context.Employees.FindAsync(employeeCode);  }  public async Task CreateEmployeeAsync(Employee employee)  {  \_context.Employees.Add(employee);  await \_context.SaveChangesAsync();  }  public async Task UpdateEmployeeAsync(Employee employee)  {  \_context.Employees.Update(employee);  await \_context.SaveChangesAsync();  }  public async Task DeleteEmployeeAsync(string employeeCode)  {  var employee = await GetEmployeeAsync(employeeCode);  if (employee != null)  {  \_context.Employees.Remove(employee);  await \_context.SaveChangesAsync();  }  }  }  } |
|  |
|  |

**Employee Model**

|  |
| --- |
| using System;  namespace Company.Data.Models  {  public class Employee  {  public string SiteId { get; set; }  public string CompanyCode { get; set; }  public string EmployeeCode { get; set; }  public string EmployeeName { get; set; }  public string Occupation { get; set; }  public string EmployeeStatus { get; set; }  public string EmailAddress { get; set; }  public string Phone { get; set; }  public DateTime LastModified { get; set; }  }  } |

**Employee Controller**

|  |
| --- |
| using Microsoft.AspNetCore.Mvc;  using System.Threading.Tasks;  using Company.Data.Repositories;  using Company.Data.Models;  using Serilog;  namespace Company.Controllers  {  [ApiController]  [Route("api/[controller]")]  public class EmployeeController |