Capstone Project

**Blog Tracker Application**

Source Code:

AdminInfo.cs

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace CapstoneProject.Models

{

[Table("AdminInfo")]

public class AdminInfo

{

[Key]

[Required(ErrorMessage = "The EmailId field is required.")]

[EmailAddress(ErrorMessage = "The EmailId field is not a valid e-mail address.")]

public string? EmailId { get; set; }

[Required(ErrorMessage = "The Password field is required.")]

public string? Password { get; set; }

}

}

BlogInfo.cs

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace CapstoneProject.Models

{

[Table("BlogInfo")]

public class BlogInfo

{

[Key]

public int BlogId { get; set; }

[Required(ErrorMessage = "The Title field is required.")]

public string? Title { get; set; }

[Required(ErrorMessage = "The Subject field is required.")]

public string? Subject { get; set; }

public DateTime DateOfCreation { get; set; }

public string? BlogUrl { get; set; }

[Required(ErrorMessage = "The EmpEmailId field is required.")]

public string? EmpEmailId { get; set; }

}

}

EmpInfo.cs

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace CapstoneProject.Models

{

[Table("EmpInfo")]

public class EmpInfo

{

[Key]

[Required(ErrorMessage = "The Email field is required.")]

public string? EmailId { get; set; }

[Required(ErrorMessage = "The Name field is required.")]

public string? Name { get; set; }

public DateTime DateOfJoining { get; set; }

[Required(ErrorMessage = "The PassCode field is required.")]

public int PassCode { get; set; }

}

}

AdminInfoesController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using CapstoneProject.Data;

using CapstoneProject.Models;

namespace CapstoneProject.Controllers

{

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

[ApiController]

public class AdminInfoesController : ControllerBase

{

private readonly CapstoneProjectDbContext \_context;

public AdminInfoesController(CapstoneProjectDbContext context)

{

\_context = context;

}

// GET: api/AdminInfoes

[HttpGet]

public async Task<ActionResult<IEnumerable<AdminInfo>>> GetAdminInfo()

{

if (\_context.AdminInfo == null)

{

return NotFound();

}

return await \_context.AdminInfo.ToListAsync();

}

// GET: api/AdminInfoes/5

[HttpGet("{id}")]

public async Task<ActionResult<AdminInfo>> GetAdminInfo(string id)

{

if (\_context.AdminInfo == null)

{

return NotFound();

}

var adminInfo = await \_context.AdminInfo.FindAsync(id);

if (adminInfo == null)

{

return NotFound();

}

return adminInfo;

}

// PUT: api/AdminInfoes/5

// To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754

[HttpPut("{id}")]

public async Task<IActionResult> PutAdminInfo(string id, AdminInfo adminInfo)

{

if (id != adminInfo.EmailId)

{

return BadRequest();

}

\_context.Entry(adminInfo).State = EntityState.Modified;

try

{

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!AdminInfoExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return NoContent();

}

// POST: api/AdminInfoes

// To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754

[HttpPost]

public async Task<ActionResult<AdminInfo>> PostAdminInfo(AdminInfo adminInfo)

{

if (\_context.AdminInfo == null)

{

return Problem("Entity set 'CapstoneProjectDbContext.AdminInfo' is null.");

}

\_context.AdminInfo.Add(adminInfo);

try

{

await \_context.SaveChangesAsync();

}

catch (DbUpdateException)

{

if (AdminInfoExists(adminInfo.EmailId))

{

return Conflict();

}

else

{

throw;

}

}

return CreatedAtAction("GetAdminInfo", new { id = adminInfo.EmailId }, adminInfo);

}

// DELETE: api/AdminInfoes/5

[HttpDelete("{id}")]

public async Task<IActionResult> DeleteAdminInfo(string id)

{

if (\_context.AdminInfo == null)

{

return NotFound();

}

var adminInfo = await \_context.AdminInfo.FindAsync(id);

if (adminInfo == null)

{

return NotFound();

}

\_context.AdminInfo.Remove(adminInfo);

await \_context.SaveChangesAsync();

return NoContent();

}

private bool AdminInfoExists(string id)

{

return (\_context.AdminInfo?.Any(e => e.EmailId == id)).GetValueOrDefault();

}

}

}

EmpInfoesController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using CapstoneProject.Data;

using CapstoneProject.Models;

namespace CapstoneProject.Controllers

{

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

[ApiController]

public class EmpInfoesController : ControllerBase

{

private readonly CapstoneProjectDbContext \_context;

public EmpInfoesController(CapstoneProjectDbContext context)

{

\_context = context;

}

// GET: api/EmpInfoes

[HttpGet]

public async Task<ActionResult<IEnumerable<EmpInfo>>> GetEmpInfo()

{

if (\_context.EmpInfo == null)

{

return NotFound();

}

return await \_context.EmpInfo.ToListAsync();

}

// GET: api/EmpInfoes/5

[HttpGet("{id}")]

public async Task<ActionResult<EmpInfo>> GetEmpInfo(string id)

{

if (\_context.EmpInfo == null)

{

return NotFound();

}

var empInfo = await \_context.EmpInfo.FindAsync(id);

if (empInfo == null)

{

return NotFound();

}

return empInfo;

}

// PUT: api/EmpInfoes/5

// To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754

[HttpPut("{id}")]

public async Task<IActionResult> PutEmpInfo(string id, EmpInfo empInfo)

{

if (id != empInfo.EmailId)

{

return BadRequest();

}

\_context.Entry(empInfo).State = EntityState.Modified;

try

{

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!EmpInfoExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return NoContent();

}

// POST: api/EmpInfoes

// To protect from overposting attacks, see https://go.microsoft.com/fwlink/?linkid=2123754

[HttpPost]

public async Task<ActionResult<EmpInfo>> PostEmpInfo(EmpInfo empInfo)

{

if (\_context.EmpInfo == null)

{

return Problem("Entity set 'CapstoneProjectDbContext.EmpInfo' is null.");

}

\_context.EmpInfo.Add(empInfo);

try

{

await \_context.SaveChangesAsync();

}

catch (DbUpdateException)

{

if (EmpInfoExists(empInfo.EmailId))

{

return Conflict();

}

else

{

throw;

}

}

return CreatedAtAction("GetEmpInfo", new { id = empInfo.EmailId }, empInfo);

}

// DELETE: api/EmpInfoes/5

[HttpDelete("{id}")]

public async Task<IActionResult> DeleteEmpInfo(string id)

{

if (\_context.EmpInfo == null)

{

return NotFound();

}

var empInfo = await \_context.EmpInfo.FindAsync(id);

if (empInfo == null)

{

return NotFound();

}

\_context.EmpInfo.Remove(empInfo);

await \_context.SaveChangesAsync();

return NoContent();

}

private bool EmpInfoExists(string id)

{

return (\_context.EmpInfo?.Any(e => e.EmailId == id)).GetValueOrDefault();

}

}

}