# Code:

**Employee.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace EmployeeService.Models

{

public class Employee

{

public int Id { get; set; }

[Required]

public string FirstName { get; set; }

public string LastName { get; set; }

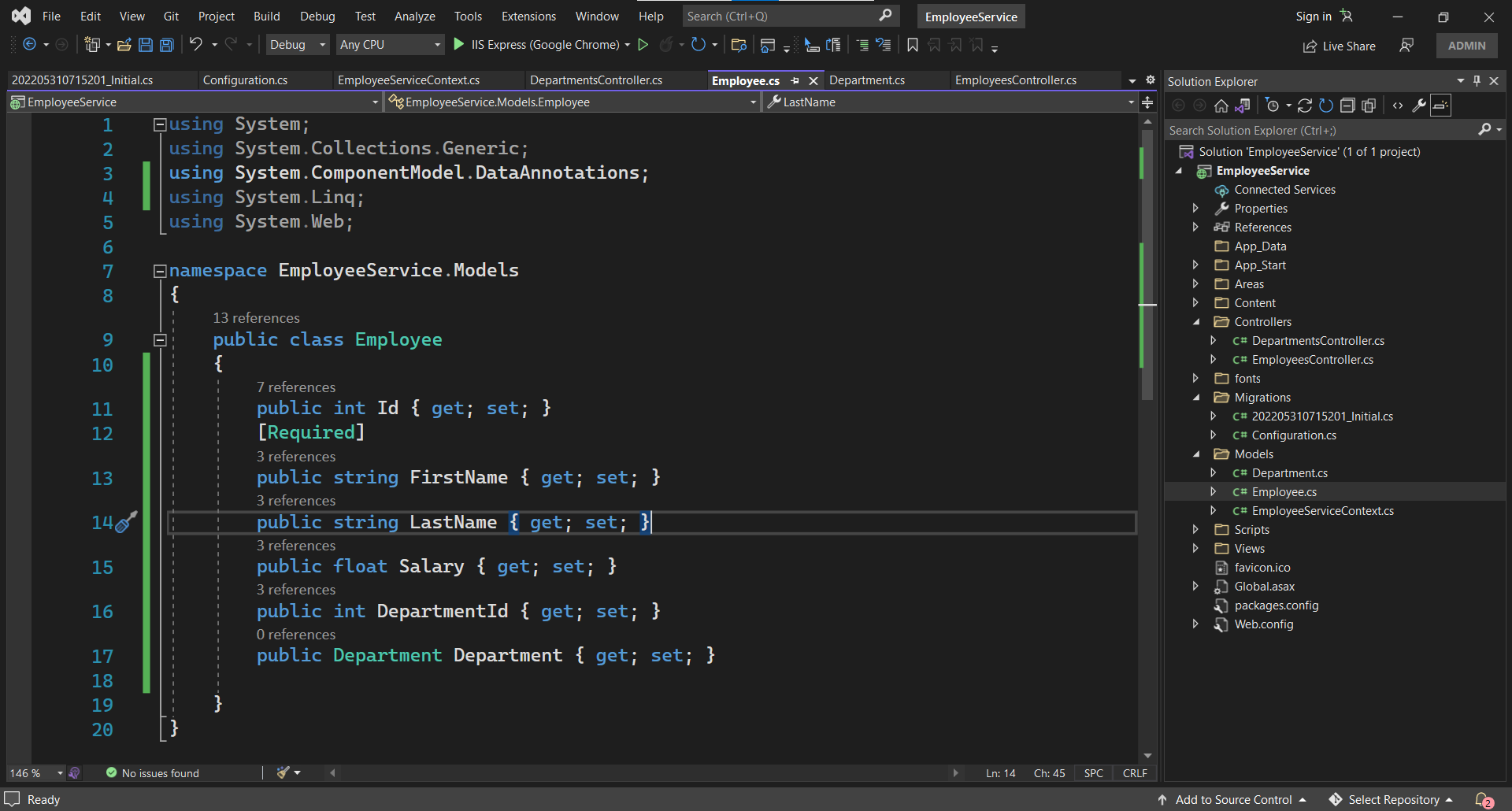
public float Salary { get; set; }

public int DepartmentId { get; set; }

public Department Department { get; set; }

}

}



**Department.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace EmployeeService.Models

{

public class Department

{

public int Id { get; set; }

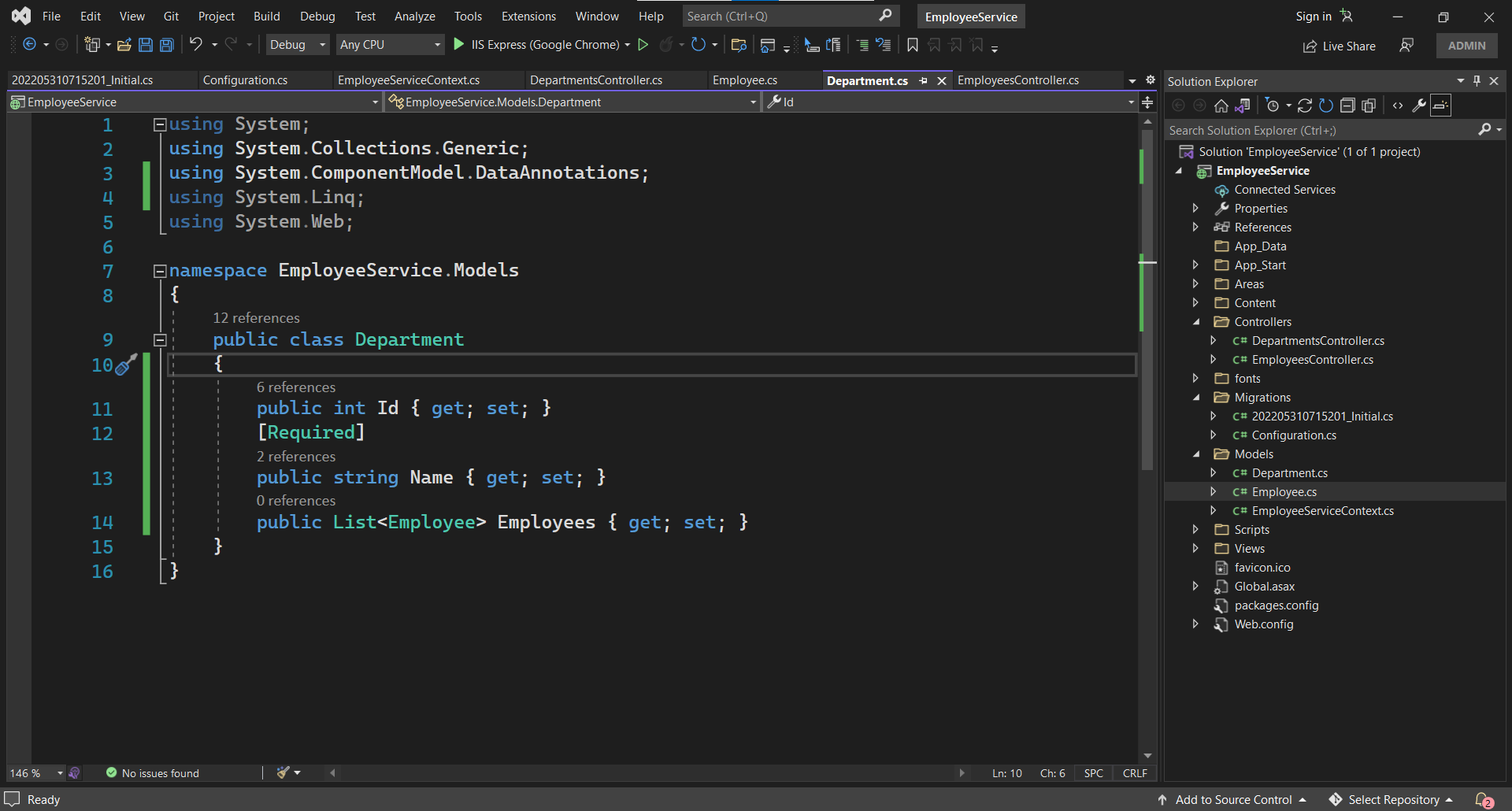
[Required]

public string Name { get; set; }

public List<Employee> Employees { get; set; }

}

}

****

**EmployeeController:**

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 EmployeeService.Models;

namespace EmployeeService.Controllers

{

public class EmployeesController : ApiController

{

private EmployeeServiceContext db = new EmployeeServiceContext();

// GET: api/Employees

public IQueryable<Employee> GetEmployees()

{

return db.Employees;

}

// GET: api/Employees/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/Employees/5

[ResponseType(typeof(void))]

public IHttpActionResult PutEmployee(int id, Employee employee)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

if (id != employee.Id)

{

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/Employees

[ResponseType(typeof(Employee))]

public IHttpActionResult PostEmployee(Employee employee)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

db.Employees.Add(employee);

db.SaveChanges();

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

}

// DELETE: api/Employees/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.Id == id) > 0;

}

}

}

**DepartmentsController:**

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 EmployeeService.Models;

namespace EmployeeService.Controllers

{

public class DepartmentsController : ApiController

{

private EmployeeServiceContext db = new EmployeeServiceContext();

// GET: api/Departments

public IQueryable<Department> GetDepartments()

{

return db.Departments;

}

// GET: api/Departments/5

[ResponseType(typeof(Department))]

public IHttpActionResult GetDepartment(int id)

{

Department department = db.Departments.Find(id);

if (department == null)

{

return NotFound();

}

return Ok(department);

}

// PUT: api/Departments/5

[ResponseType(typeof(void))]

public IHttpActionResult PutDepartment(int id, Department department)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

if (id != department.Id)

{

return BadRequest();

}

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

try

{

db.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

if (!DepartmentExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return StatusCode(HttpStatusCode.NoContent);

}

// POST: api/Departments

[ResponseType(typeof(Department))]

public IHttpActionResult PostDepartment(Department department)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

db.Departments.Add(department);

db.SaveChanges();

return CreatedAtRoute("DefaultApi", new { id = department.Id }, department);

}

// DELETE: api/Departments/5

[ResponseType(typeof(Department))]

public IHttpActionResult DeleteDepartment(int id)

{

Department department = db.Departments.Find(id);

if (department == null)

{

return NotFound();

}

db.Departments.Remove(department);

db.SaveChanges();

return Ok(department);

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

private bool DepartmentExists(int id)

{

return db.Departments.Count(e => e.Id == id) > 0;

}

}

}

**Configuration.cs**

namespace EmployeeService.Migrations

{

using System;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

using EmployeeService.Models;

internal sealed class Configuration : DbMigrationsConfiguration<EmployeeService.Models.EmployeeServiceContext>

{

public Configuration()

{

AutomaticMigrationsEnabled = false;

}

protected override void Seed(EmployeeService.Models.EmployeeServiceContext context)

{

// This method will be called after migrating to the latest version.

// You can use the DbSet<T>.AddOrUpdate() helper extension method

// to avoid creating duplicate seed data.

context.Departments.AddOrUpdate(x => x.Id,

new Department() { Id = 100, Name = "HR" },

new Department() { Id = 101, Name = "Technical" }

);

context.Employees.AddOrUpdate(x => x.Id,

new Employee() { Id = 1, FirstName = "John", LastName = "Smith", DepartmentId = 101, Salary = 30000 },

new Employee() { Id = 2, FirstName = "Mary", LastName = "Jane", DepartmentId = 100, Salary = 20000 },

new Employee() { Id = 3, FirstName = "Steve", LastName = "Lopez", DepartmentId = 101, Salary = 50000 }

);

}

}

}

**EmployeeSeviceContext.cs:**

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Web;

namespace EmployeeService.Models

{

public class EmployeeServiceContext : DbContext

{

// You can add custom code to this file. Changes will not be overwritten.

//

// If you want Entity Framework to drop and regenerate your database

// automatically whenever you change your model schema, please use data migrations.

// For more information refer to the documentation:

// http://msdn.microsoft.com/en-us/data/jj591621.aspx

public EmployeeServiceContext() : base("name=EmployeeServiceContext")

{

}

public System.Data.Entity.DbSet<EmployeeService.Models.Department> Departments { get; set; }

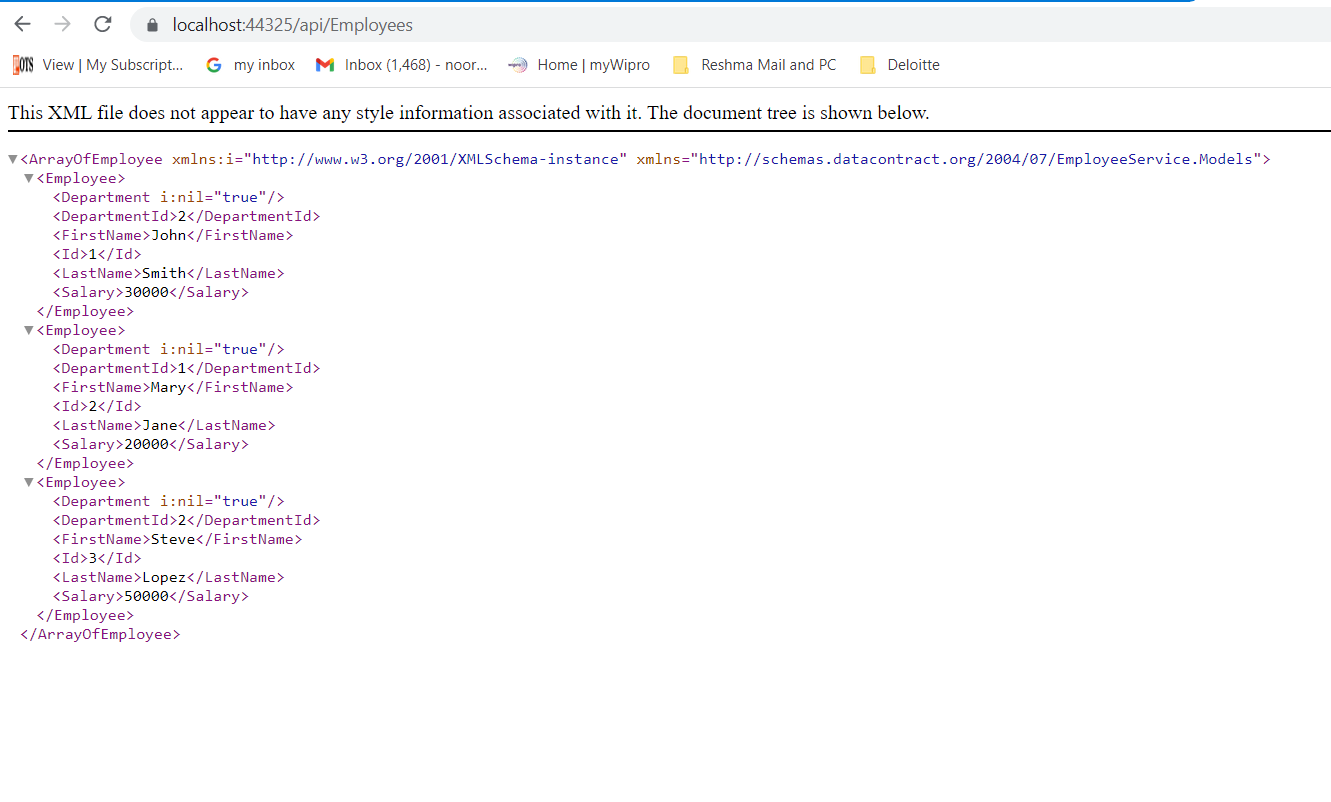
public System.Data.Entity.DbSet<EmployeeService.Models.Employee> Employees { get; set; }

}

}

Output:

url: /api/Employees



url: /api/Departments

