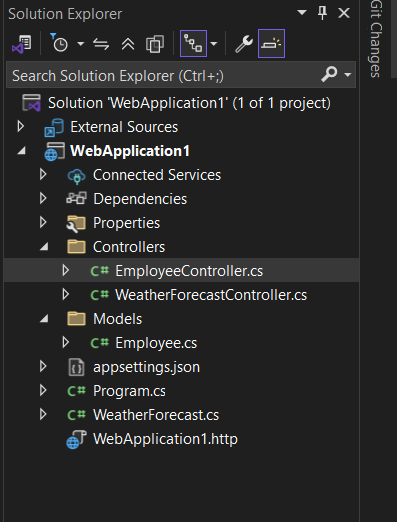
**WebAPI\_HandsOn-4**

**Web Api CRUD operation**

**File Structure:**



**Models/Employee.cs**

namespace WebApplication1.Models

{

public class Employee

{

public int Id { get; set; }

public string? Name { get; set; }

public string? Department { get; set; }

public double Salary { get; set; }

}

}

**EmployeeController.cs**

using Microsoft.AspNetCore.Mvc;

using WebApplication1.Models;

using WebApplication1.Models;

namespace MyWebAPIProject.Controllers

{

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

[ApiController]

public class EmployeeController : ControllerBase

{

private static List<Employee> employees = new List<Employee>()

{

new Employee { Id = 1, Name = "Anurag", Department = "HR", Salary = 50000 },

new Employee { Id = 2, Name = "Bob", Department = "IT", Salary = 70000 },

new Employee { Id = 3, Name = "Charlie", Department = "Finance", Salary = 60000 }

};

// GET api/employee

[HttpGet]

public ActionResult<IEnumerable<Employee>> GetAll()

{

return Ok(employees);

}

// GET api/employee/{id}

[HttpGet("{id}")]

public ActionResult<Employee> GetById(int id)

{

var emp = employees.FirstOrDefault(e => e.Id == id);

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

return Ok(emp);

}

// POST api/employee

[HttpPost]

public ActionResult<Employee> AddEmployee([FromBody] Employee emp)

{

emp.Id = employees.Max(e => e.Id) + 1;

employees.Add(emp);

return CreatedAtAction(nameof(GetById), new { id = emp.Id }, emp);

}

// PUT api/employee/2

[HttpPut("{id}")]

public ActionResult<Employee> UpdateEmployee(int id, [FromBody] Employee updated)

{

if (id <= 0)

return BadRequest("Invalid employee id");

var existing = employees.FirstOrDefault(e => e.Id == id);

if (existing == null)

return BadRequest("Invalid employee id");

// update the values

existing.Name = updated.Name;

existing.Department = updated.Department;

existing.Salary = updated.Salary;

return Ok(existing);

}

// DELETE api/employee/2

[HttpDelete("{id}")]

public ActionResult DeleteEmployee(int id)

{

var emp = employees.FirstOrDefault(e => e.Id == id);

if (emp == null)

return NotFound();

employees.Remove(emp);

return NoContent();

}

// For testing error

[HttpGet("GetStandard")]

public IActionResult GetStandard()

{

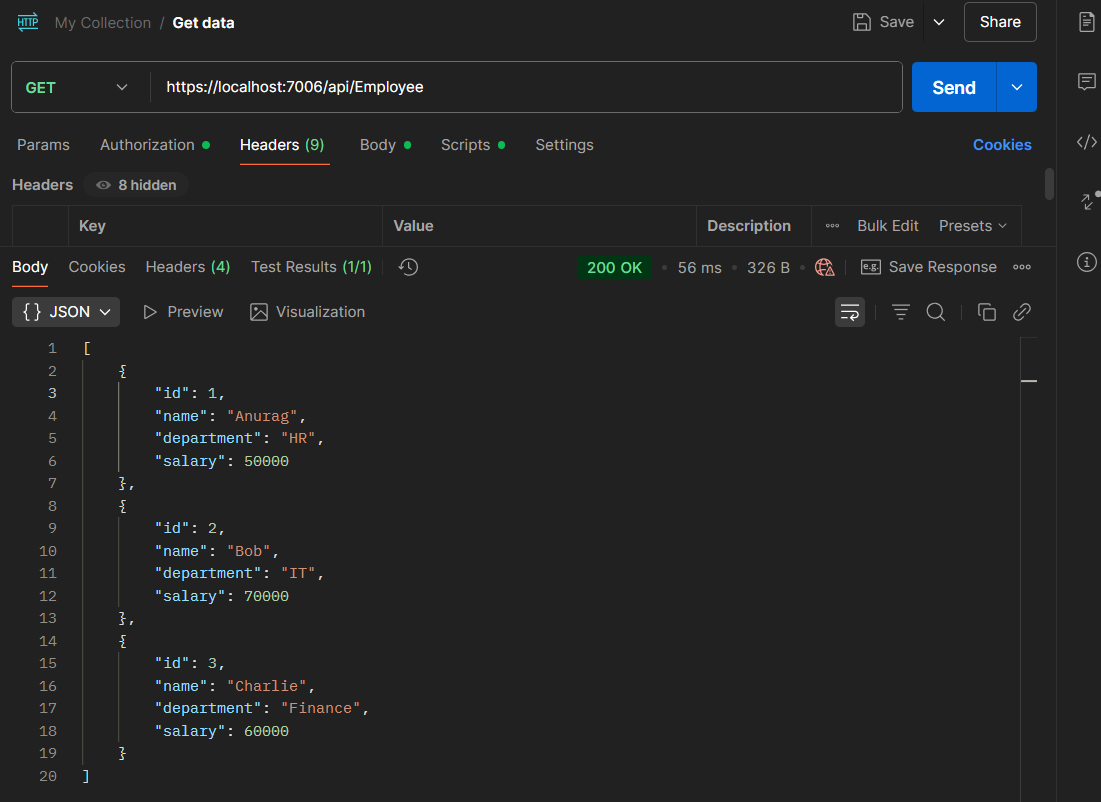
throw new Exception("Something bad happened!");

}

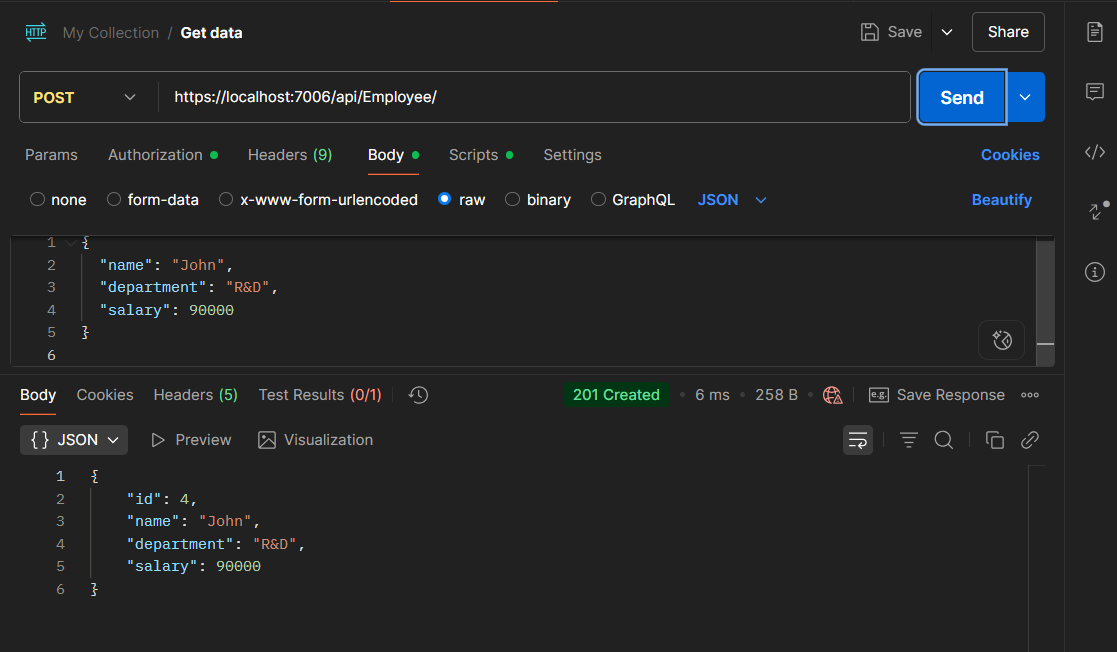
}

}

**// GET api/employee**



**// POST api/employee**



// For testing error

**https://localhost:7006/api/Employee/GetStandard**

