1. **Web Api CRUD operation**

Update Employee data as per the input thru Web API PUT action method call

Employee information has to be updated based on the user input. Use Swagger tool to invoke the action method mapped with Http PUT action verb to update an employee data.

Modify the action method to return Employee data thru ActionResult.

Check if the id value is lesser than or equal to 0. If true, throw BadRequest action result with the message ‘Invalid employee id’

If the value is greater than 0 but not available in the list of employee ids that is there in the hardcoded list of employees, throw BadRequest action result with the same message as stated above.

If the id value is valid, use the JSON data from the input body and update the hardcoded list. Filter the employee list data for the input id and return that as the output.

CODE:

EmployeeController.cs

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using MyWebApi.Models;

namespace MyWebApi.Controllers

{

[Route("api/Emp")]

[ApiController]

public class EmployeeController : ControllerBase

{

private readonly List<Employee> \_employees;

public EmployeeController()

{

\_employees = GetStandardEmployeeList();

}

private List<Employee> GetStandardEmployeeList()

{

return new List<Employee>

{

new Employee { Id = 1, Name = "John Doe", Salary = 50000, Permanent = true, Department = new Department { Id = 1, Name = "IT" }, Skills = new List<Skill> { new Skill { Id = 1, Name = "C#" } }, DateOfBirth = new System.DateTime(1990, 1, 1) },

new Employee { Id = 2, Name = "Jane Smith", Salary = 60000, Permanent = false, Department = new Department { Id = 2, Name = "HR" }, Skills = new List<Skill> { new Skill { Id = 2, Name = "HRM" } }, DateOfBirth = new System.DateTime(1985, 5, 10) }

};

}

[HttpPut("{id}")]

[ProducesResponseType(StatusCodes.Status200OK)]

[ProducesResponseType(StatusCodes.Status400BadRequest)]

public ActionResult<Employee> Put(int id, [FromBody] Employee employee)

{

if (id <= 0)

{

return BadRequest("Invalid employee id");

}

var existingEmployee = \_employees.FirstOrDefault(e => e.Id == id);

if (existingEmployee == null)

{

return BadRequest("Invalid employee id");

}

if (employee == null || string.IsNullOrEmpty(employee.Name)) // Basic validation for required fields

{

return BadRequest("A non-empty request body is required.");

}

existingEmployee.Name = employee.Name;

existingEmployee.Salary = employee.Salary;

existingEmployee.Permanent = employee.Permanent;

existingEmployee.Department = employee.Department;

existingEmployee.Skills = employee.Skills;

existingEmployee.DateOfBirth = employee.DateOfBirth;

return Ok(existingEmployee);

}

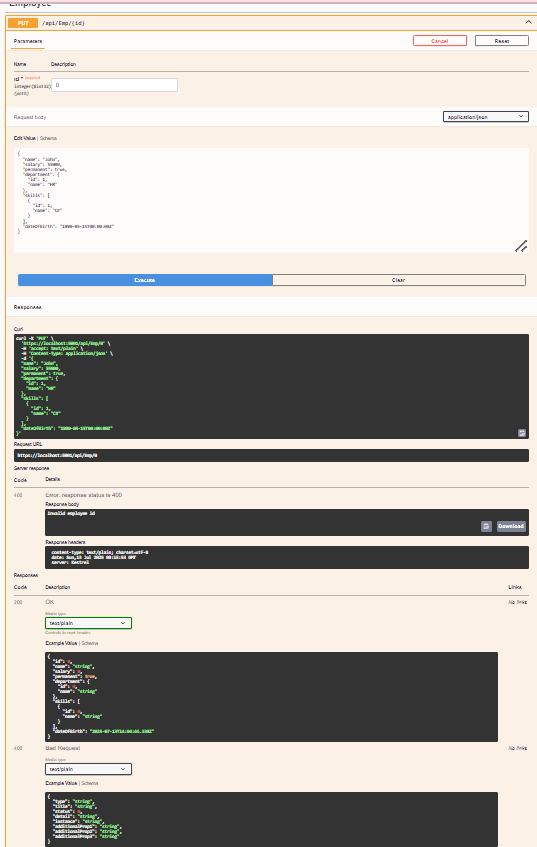
}

}

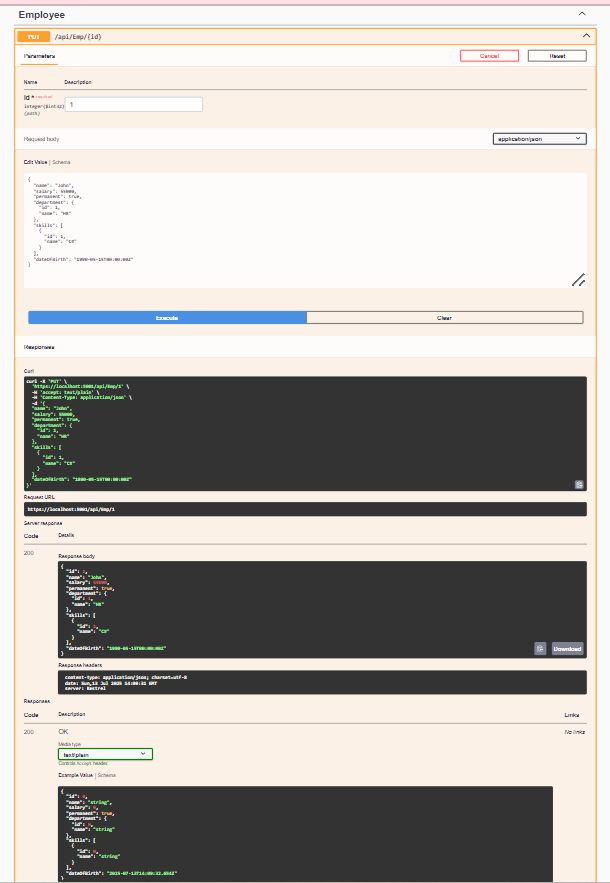
OUTPUT:

Swagger

1. When id <= 0, it shows Error: response status is 400.



1. When id = 1, it shows valid 200 OK.



CODE: