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:**

**Employee.cs**

namespace HandsOn4.Model

{

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**

namespace HandsOn4.Controllers

{

using HandsOn4.Model;

using Microsoft.AspNetCore.Mvc;

using System.Collections.Generic;

using System.Linq;

[ApiController]

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

public class EmployeeController : ControllerBase

{

// Static hardcoded list simulating a DB

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

{

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

new Employee { Id = 2, Name = "Alice", Department = "IT", Salary = 60000 },

new Employee { Id = 3, Name = "Bob", Department = "Finance", Salary = 55000 }

};

// PUT: api/employee/2

[HttpPut("{id}")]

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

{

if (id <= 0)

{

return BadRequest("Invalid employee id");

}

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

if (existingEmp == null)

{

return BadRequest("Invalid employee id");

}

// Update employee details

existingEmp.Name = updatedEmp.Name;

existingEmp.Department = updatedEmp.Department;

existingEmp.Salary = updatedEmp.Salary;

return Ok(existingEmp);

}

}

}

**OUTPUT:**



