**Name: Deepanshu Rathore**

**Superset ID: 6358199**  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Week 4  
**4. WebApi\_Handson**When building a Web API that allows creation, updating, and deletion of data, the core component is an action method within a controller. Such a method should accept data sent in the request body, map it into a custom model class, and then apply the required changes to an in‑memory collection or other storage. The [FromBody] attribute instructs the framework to deserialize the JSON payload into your model type. For update operations in particular, you first validate the incoming identifier. If the identifier is not a positive integer, the method returns a bad request response indicating that the identifier is invalid. If the identifier is positive but does not correspond to any existing record in the hardcoded list, the same bad request response is returned. Only when the identifier is valid and present do you apply the new values from the deserialized model to the existing entry and then return the updated entry wrapped in an ActionResult<T> to allow for flexible HTTP responses.

EmployeeController.cs:

using Microsoft.AspNetCore.Mvc;

using EmployeeApiDemo.Models;

using System.Collections.Generic;

using System.Linq;

namespace EmployeeApiDemo.Controllers

{

[ApiController]

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

public class EmployeesController : ControllerBase

{

private static readonly List<Employee> \_employees = new List<Employee>

{

new Employee { Id = 1, Name = "Alice Johnson", Age = 30, Department = "Finance" },

new Employee { Id = 2, Name = "Bob Smith", Age = 45, Department = "IT" },

new Employee { Id = 3, Name = "Carol Peters", Age = 28, Department = "HR" }

};

[HttpGet]

public ActionResult<IEnumerable<Employee>> GetAll()

{

return Ok(\_employees);

}

[HttpPut("{id}")]

public ActionResult<Employee> Update(int id, [FromBody] Employee updatedEmployee)

{

if (id <= 0)

{

return BadRequest("Invalid employee id");

}

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

if (existing == null)

{

return BadRequest("Invalid employee id");

}

existing.Name = updatedEmployee.Name;

existing.Age = updatedEmployee.Age;

existing.Department = updatedEmployee.Department;

return Ok(existing);

}

[HttpPost]

public ActionResult<Employee> Create([FromBody] Employee newEmployee)

{

if (newEmployee == null || newEmployee.Id <= 0)

{

return BadRequest("Invalid employee data");

}

if (\_employees.Any(e => e.Id == newEmployee.Id))

{

return BadRequest("Employee id already exists");

}

\_employees.Add(newEmployee);

return CreatedAtAction(nameof(GetAll), new { id = newEmployee.Id }, newEmployee);

}

[HttpDelete("{id}")]

public IActionResult Delete(int id)

{

if (id <= 0)

{

return BadRequest("Invalid employee id");

}

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

if (existing == null)

{

return BadRequest("Invalid employee id");

}

\_employees.Remove(existing);

return NoContent();

}

}

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

