**Lab 4: Web\_API\_HandsOn**

**Code**

**Employee.cs**

**namespace EmployeeAPI.Models**

**{**

**public class Employee**

**{**

**public int Id { get; set; } // Employee ID**

**public string Name { get; set; } // Employee Name**

**public string Department { get; set; } // Employee Department**

**}**

**}**

**EmployeeController.cs**

**using Microsoft.AspNetCore.Mvc;**

**using EmployeeAPI.Models;**

**namespace EmployeeAPI.Controllers**

**{**

**[ApiController]**

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

**public class EmployeeController : ControllerBase**

**{**

**// Static list of employees for demo purpose**

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

**{**

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

**new Employee { Id = 2, Name = "Alice", Department = "Finance" },**

**new Employee { Id = 3, Name = "Bob", Department = "IT" }**

**};**

**[HttpPut("{id}")]**

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

**{**

**// Validate id**

**if (id <= 0)**

**{**

**return BadRequest("Invalid employee id");**

**}**

**// Find employee by id**

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

**if (existingEmployee == null)**

**{**

**return BadRequest("Invalid employee id");**

**}**

**// Update the employee**

**existingEmployee.Name = updatedEmployee.Name;**

**existingEmployee.Department = updatedEmployee.Department;**

**// Return the updated employee**

**return Ok(existingEmployee);**

**}**

**// We'll add the PUT method in the next step**

**}**

**}**

**Screenshot**

**A screenshot of a computer

AI-generated content may be incorrect.**