08/10/2025

Somashila Akhilesh

DotnetTask2_WEBAPI1_Collection

Employee Crud with Collection using Web API Core with Client Console Application (.Net Framework)

Creating Web API Application

Step 1: Create a Web API Core application in Visual Studio

Step 2: Create Model class of Employee with the properties and their respective validations as mentioned

Step 3: Create interface, IEmployeeRepository.cs to define the Action Methods

Step 4: Create Class, EmployeeRepository.cs to implement those actions defined in Interface(IEmployeeRepository)

Step 5: Create Controller[Select API Controller while creating], EmployeeController.cs to manage the requests.

It Should Inherit from ControllerBase and should have

[ApiController]

Step 6: Add Dependency Injection in Program.cs File as

builder.Services.AddScoped<IEmployeeRepository,EmployeeRepository>();

Step 7: Run the Project; It will be live on Localhost.

Creating HttpClient for Web API – Console Application

Step 1: Create Console Application while API Project is live ensure to use Asynchronization in the Main.

Step 2: Install Nuget Package: System.Net.Http.Json;

Step 3: Import the namespaces:

- using System.Net.Http;
- using System.Net.Http.Json;

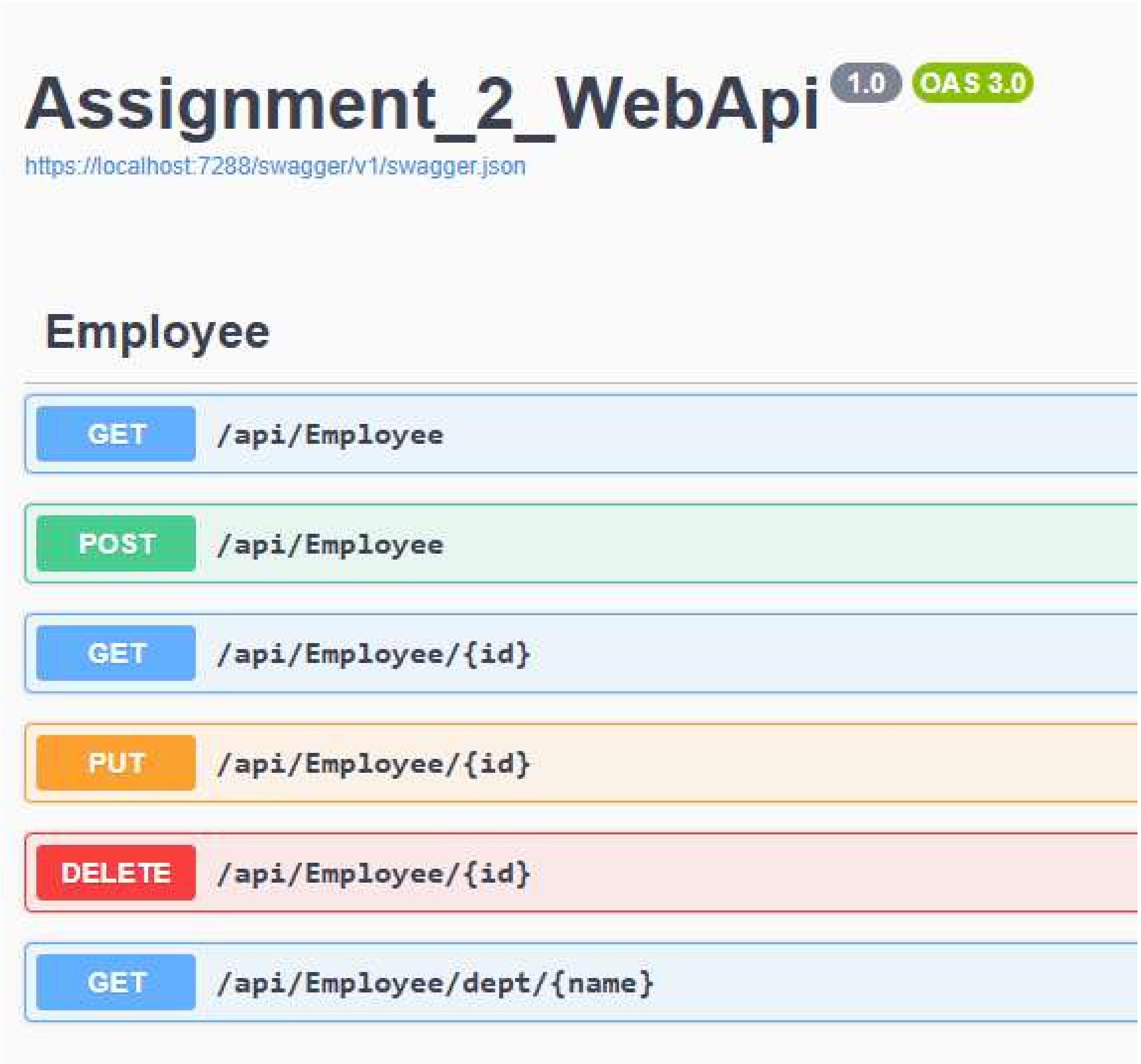
Step 4: Create an Employee Class and set the properties accordingly.

Step 5: Create an Object of HttpClient and use the localhost url onto the project as string.

Step 6: Define a list and load them from the Httpclient using the apibaseurl and Display the Employee list on the Console.

OUTPUTS

HOMEPAGE



Assignment_2_WebApi

https://localhost:7288/swagger/v1/swagger.json

Employee

GET /api/Employee

POST /api/Employee

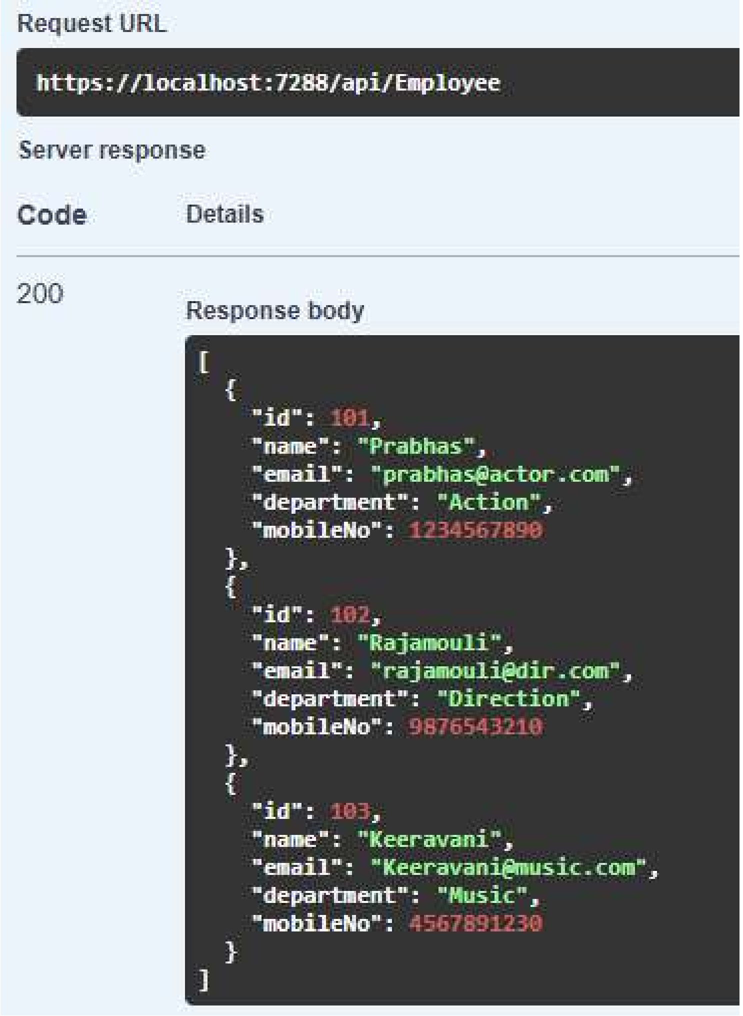
GET /api/Employee/{id}

PUT /api/Employee/{id}

DELETE /api/Employee/{id}

GET /api/Employee/dept/{name}

GET



Request URL

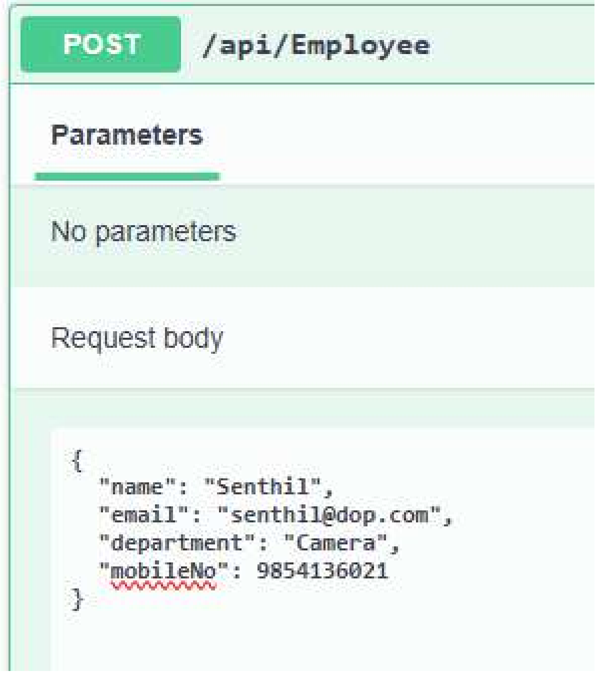
```
https://localhost:7288/api/Employee
```

Server response

Code	Details
200	Response body

```
[
  {
    "id": 101,
    "name": "Prabhas",
    "email": "prabhas@actor.com",
    "department": "Action",
    "mobileNo": 1234567890
  },
  {
    "id": 102,
    "name": "Rajamouli",
    "email": "rajamouli@dir.com",
    "department": "Direction",
    "mobileNo": 9876543210
  },
  {
    "id": 103,
    "name": "Keeravani",
    "email": "Keeravani@music.com",
    "department": "Music",
    "mobileNo": 4567891230
  }
]
```

POST



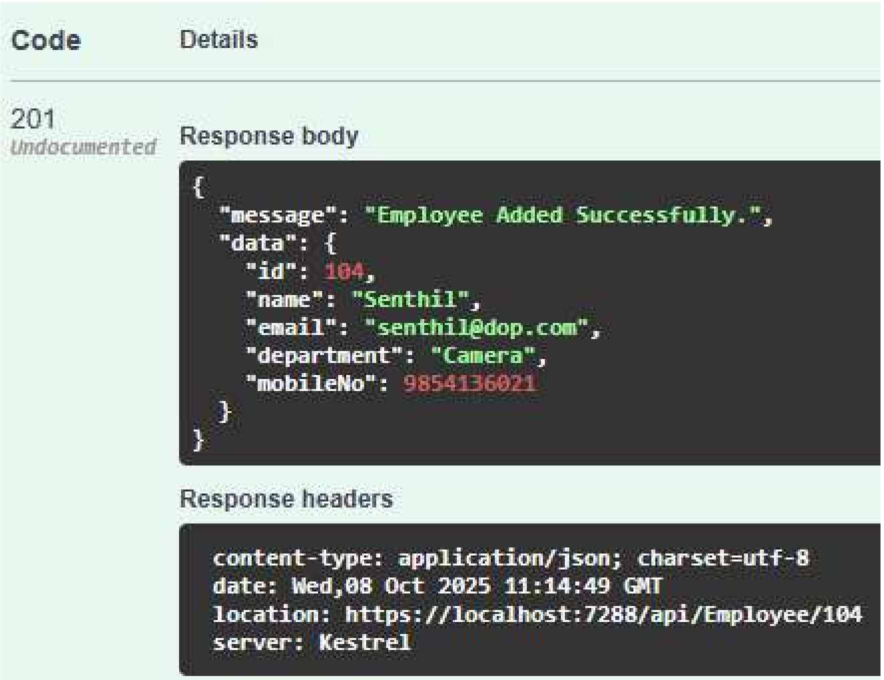
/api/Employee

Parameters

No parameters

Request body

```
{
  "name": "Senthil",
  "email": "senthil@dop.com",
  "department": "Camera",
  "mobileNo": 9854136021
}
```



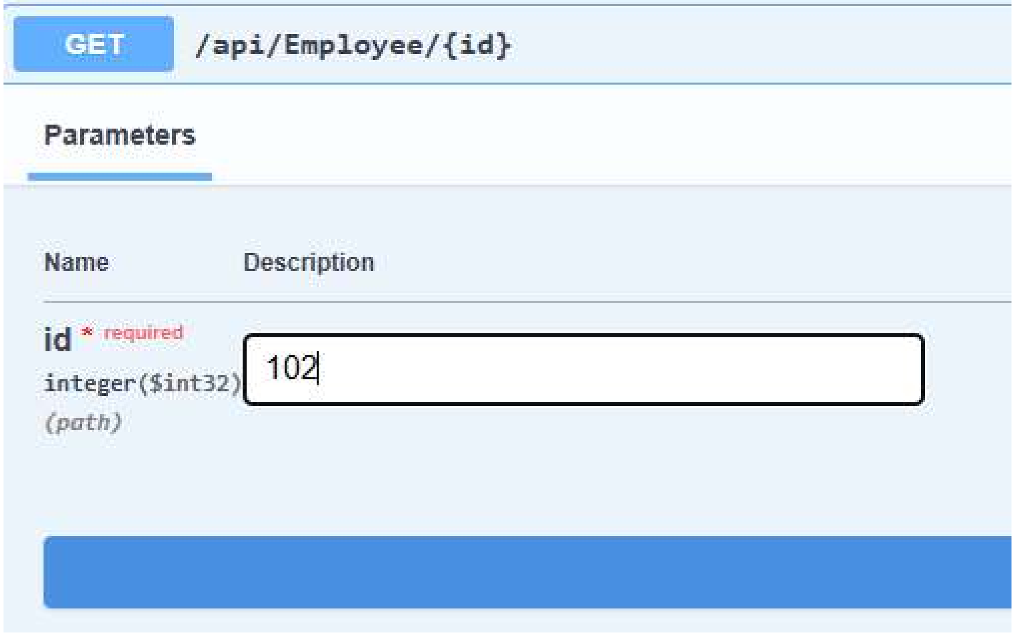
Code	Details
201 *Undocumented*	Response body

```
{
  "message": "Employee Added Successfully.",
  "data": {
    "id": 104,
    "name": "Senthil",
    "email": "senthil@dop.com",
    "department": "Camera",
    "mobileNo": 9854136021
  }
}
```

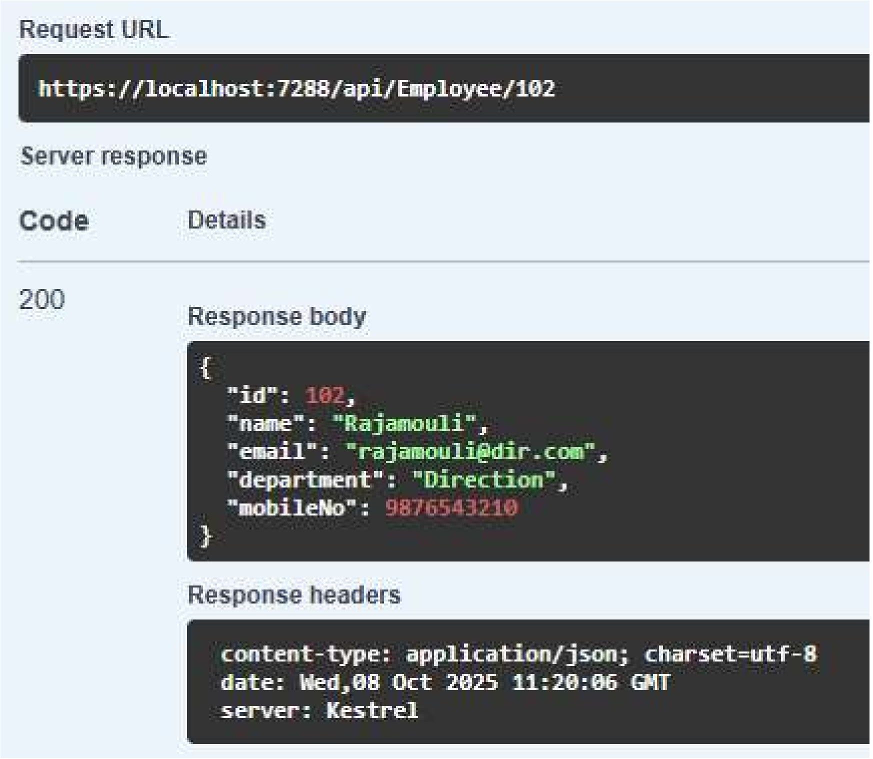
Response headers

```
content-type: application/json; charset=utf-8
date: Wed,08 Oct 2025 11:14:49 GMT
location: https://localhost:7288/api/Employee/104
server: Kestrel
```

GET BY ID



id: 102



Request URL

```
https://localhost:7288/api/Employee/102
```

Server response

Code: 200

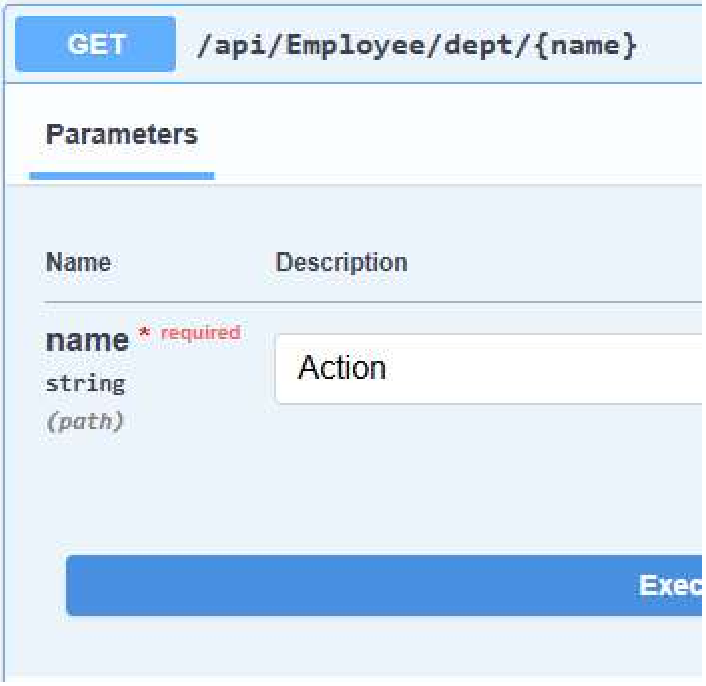
Response body

```
{
  "id": 102,
  "name": "Rajamouli",
  "email": "rajamouli@dir.com",
  "department": "Direction",
  "mobileNo": 9876543210
}
```

Response headers

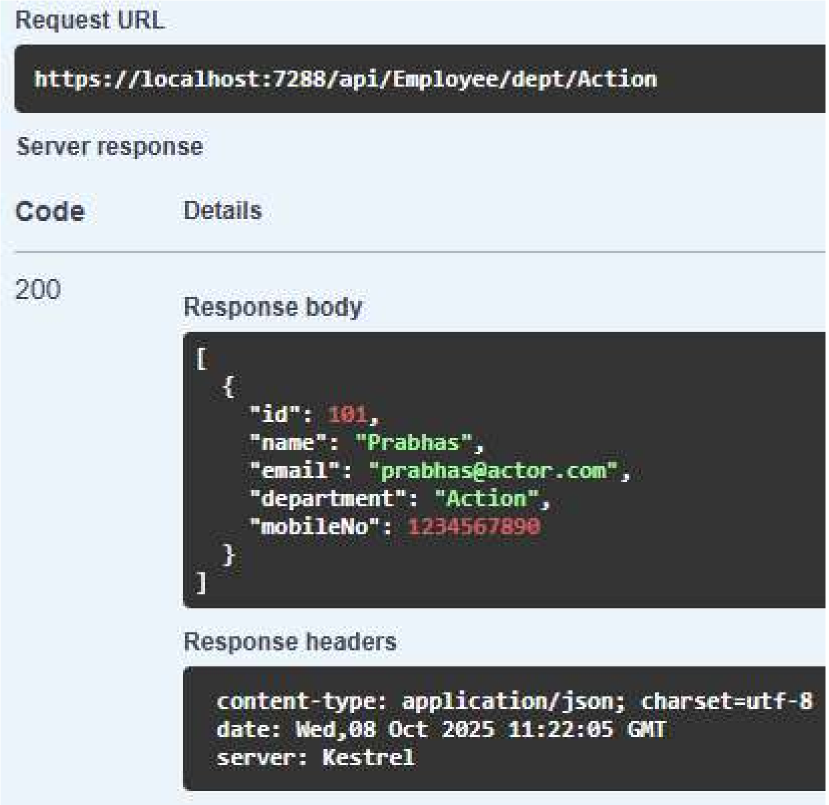
```
 content-type: application/json; charset=utf-8
 date: Wed,08 Oct 2025 11:20:06 GMT
 server: Kestrel
```

GET BY DEPT



/api/Employee/dept/{name}

Name	Description
name * required string *(path)*	Action


Request URL

```
https://localhost:7288/api/Employee/dept/Action
```

Server response

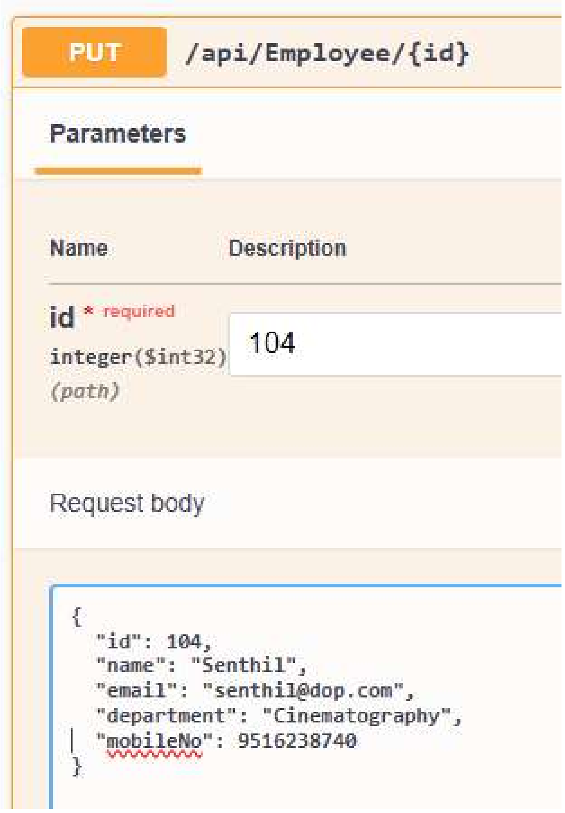
Code	Details
200	Response body

```
[
  {
    "id": 101,
    "name": "Prabhas",
    "email": "prabhas@actor.com",
    "department": "Action",
    "mobileNo": 1234567890
  }
]
```

Response headers

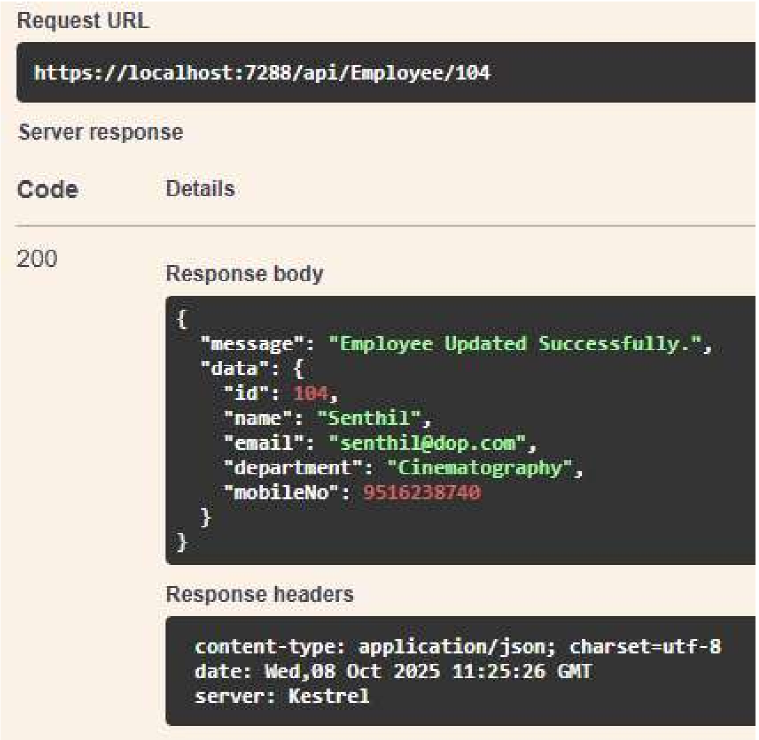
```
 content-type: application/json; charset=utf-8
 date: Wed,08 Oct 2025 11:22:05 GMT
 server: Kestrel
```

PUT



id * required
integer($int32)
(path)
104

```
{
  "id": 104,
  "name": "Senthil",
  "email": "senthil@dop.com",
  "department": "Cinematography",
  "mobileNo": 9516238740
}
```



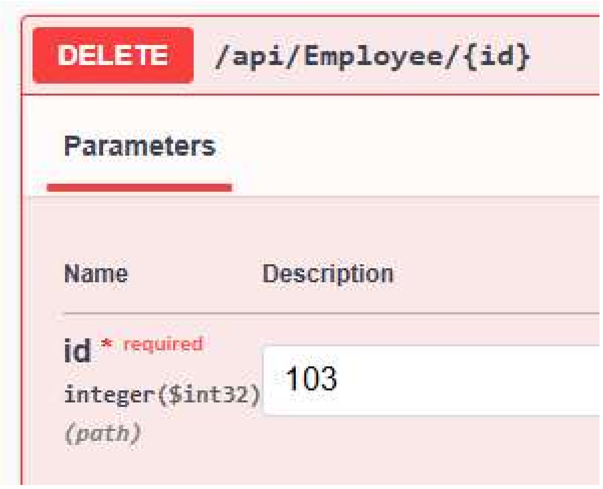
```
https://localhost:7288/api/Employee/104
```

200

```
{
  "message": "Employee Updated Successfully.",
  "data": {
    "id": 104,
    "name": "Senthil",
    "email": "senthil@dop.com",
    "department": "Cinematography",
    "mobileNo": 9516238740
  }
}
```

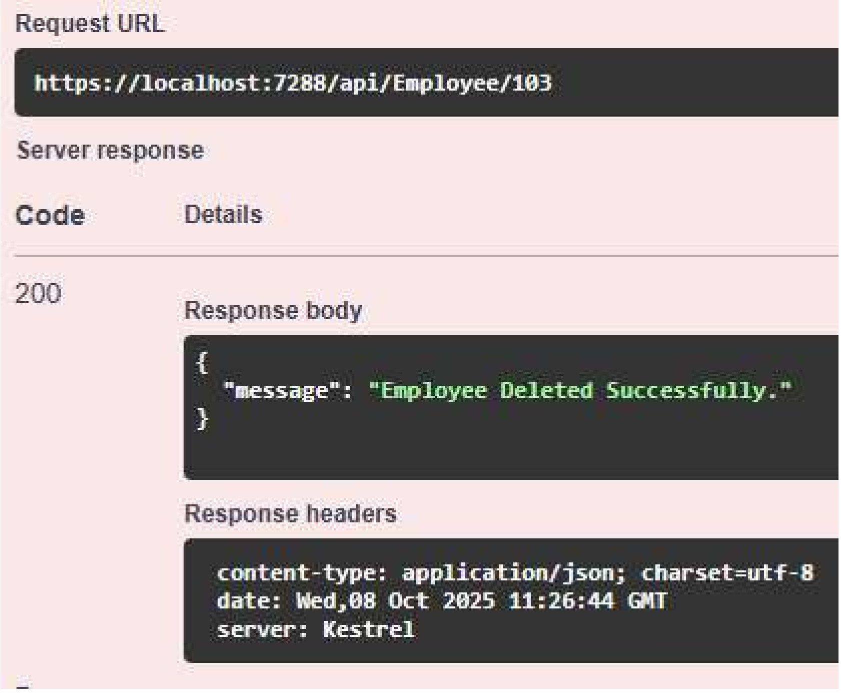
```
content-type: application/json; charset=utf-8
date: Wed,08 Oct 2025 11:25:26 GMT
server: Kestrel
```

DELETE



/api/Employee/{id}

Name	Description
id * required integer($int32) *(path)*	103



Request URL

```
https://localhost:7288/api/Employee/103
```

Server response

Code	Details
200	Response body

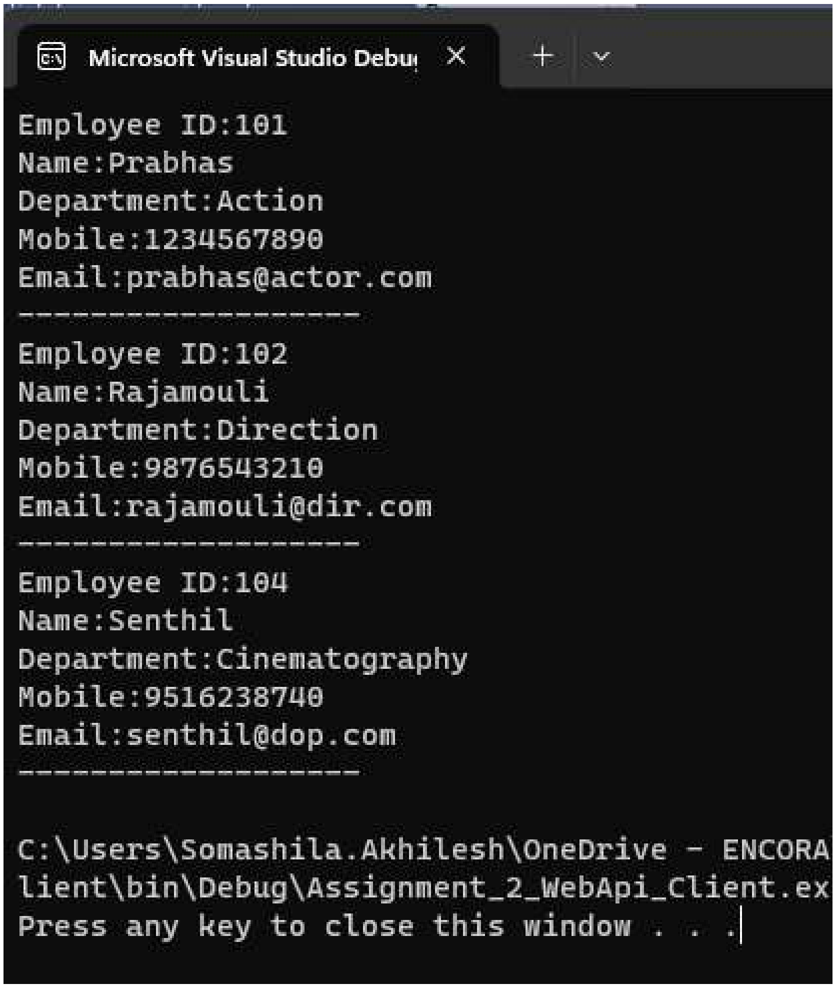
```
{
  "message": "Employee Deleted Successfully."
}
```

Response headers

```
 content-type: application/json; charset=utf-8
 date: Wed,08 Oct 2025 11:26:44 GMT
 server: Kestrel
```

CLIENT APPLICATION – CONSOLE APP

To Get and Show All Employees



```
Employee ID:101
Name:Prabhas
Department:Action
Mobile:1234567890
Email:prabhas@actor.com
-------------------
Employee ID:102
Name:Rajamouli
Department:Direction
Mobile:9876543210
Email:rajamouli@dir.com
-------------------
Employee ID:104
Name:Senthil
Department:Cinematography
Mobile:9516238740
Email:senthil@dop.com
-------------------

C:\Users\Somashila.Akhilesh\OneDrive - ENCORA
lient\bin\Debug\Assignment_2_WebApi_Client.ex
Press any key to close this window . . .
```

```
using System.ComponentModel.DataAnnotations;

namespace Assignment_2_WebApi.Models
{
    public class Employee
    {
        [Required(ErrorMessage = "Id is Mandatory")]
        public int Id { get; set; }

        [Required(ErrorMessage = "Name is Mandatory.")]
        [StringLength(30, MinimumLength = 3, ErrorMessage = "Name must be between 3 and 30 characters.")]
        public required string Name { get; set; }


        [EmailAddress(ErrorMessage = "Invalid email format.")]
        public string Email { get; set; }

        public string Department { get; set; }

        [Range(1000000000, 9999999999, ErrorMessage = "Mobile number must be a 10-digit number.")]
        public long MobileNo { get; set; }


    }
}
```

```
using Assignment_2_WebApi.Models;

namespace Assignment_2_WebApi.Repositories
{
    public interface IEmployeeRepository
    {
        public List<Employee> GetAllEmployees();
        public Employee GetEmployeeById(int id);
        public List<Employee> GetEmployeesByDept(string name);
        public void AddEmployee(Employee employee);
        public void UpdateEmployee(Employee employee);
        public void DeleteEmployee(int id);
    }
}

```

```
using Assignment_2_WebApi.Models;
using Microsoft.AspNetCore.Mvc.ModelBinding;

namespace Assignment_2_WebApi.Repositories
{
    public class EmployeeRepository:IEmployeeRepository
    {
        static List<Employee> _employees = new List<Employee>()
        {
            new Employee(){Id=101,Name="Prabhas",Email="prabhas@actor.com",Department="Action",MobileNo=1234567890},
            new Employee(){Id=102,Name="Rajamouli",Email="rajamouli@dir.com",Department="Direction",MobileNo=9876543210},
            new Employee(){Id=103,Name="Keeravani",Email="Keeravani@music.com",Department="Music",MobileNo=4567891230}
        };

        public void AddEmployee(Employee employee)
        {
            employee.Id = _employees.Max(e => e.Id) + 1;
            _employees.Add(employee);
        }
        public void DeleteEmployee(int id)
        {
            var emp = GetEmployeeById(id);
            if(emp != null)
            {
                _employees.Remove(emp);
            }
        }

        public Employee GetEmployeeById(int id)
        {
            var emp = _employees.FirstOrDefault(e=> e.Id == id);
            if(emp != null)
            {
                return emp;
            }
            return null;
        }

        public List<Employee> GetAllEmployees()
        {
            return _employees;
        }

        public List<Employee> GetEmployeesByDept(string name)
        {
            var emps = _employees.FindAll(e => e.Department == name);
            return emps;
        }

        public void UpdateEmployee(Employee emp)
        {
            var existing = _employees.FirstOrDefault(e => e.Id == emp.Id);
            if(existing != null)
            {
                existing.Name = emp.Name;
                existing.Email = emp.Email;
                existing.Department = emp.Department;
                existing.MobileNo= emp.MobileNo;
            }
        }
    }
}

```

```
using Assignment_2_WebApi.Models;
using Assignment_2_WebApi.Repositories;
using Microsoft.AspNetCore.Mvc;


namespace Assignment_2_WebApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class EmployeeController : ControllerBase
    {
        private readonly IEmployeeRepository _repository;

        public EmployeeController(IEmployeeRepository repository)
        {
            _repository = repository;
        }

        [HttpGet]
        public IActionResult Get()
        {
            if(!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            var employees = _repository.GetAllEmployees();
            return Ok(employees);
        }

        [HttpGet("{id}")]
        public IActionResult Get(int id)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }
            var employee = _repository.GetEmployeeById(id);
            if (employee == null)
            {
                return NotFound();
            }
            return Ok(employee);
        }

        [HttpGet("dept/{name}")]
        public IActionResult GetByDept(string name)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }
            var employees = _repository.GetEmployeesByDept(name);
            if (employees == null || employees.Count == 0)
            {
                return NotFound();
            }
            return Ok(employees);
        }

        [HttpPost]
        public IActionResult CreateEmployee([FromBody] Employee employee)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }
            _repository.AddEmployee(employee);
            return CreatedAtAction(nameof(Get), new { id = employee.Id }, new { message = "Employee Added Successfully.", data = employee });
        }

        [HttpPut("{id}")]
        public IActionResult UpdateEmployee(int id, [FromBody] Employee employee)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }
            _repository.UpdateEmployee(employee);
            return Ok(new { message = "Employee Updated Successfully.", data = employee });
        }

        [HttpDelete("{id}")]
        public IActionResult Delete(int id)
        {
            _repository.DeleteEmployee(id);
            return Ok(new { message = "Employee Deleted Successfully." });

        }
    }
}

```

```
using Assignment_2_WebApi.Repositories;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddScoped<IEmployeeRepository,EmployeeRepository>();
var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();

```

Github Links for my project:
https://github.com/Akhilesh-Dotnet-Hyd/Assignment_2_WebApi_Client.git
https://github.com/Akhilesh-Dotnet-Hyd/Assignment_2_WebApi.git

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Assignment_2_WebApi_Client
{
    internal class Employee
    {
        public int Id { get; set; }

        public string Name { get; set; }

        public string Email { get; set; }

        public string Department { get; set; }

        public long MobileNo { get; set; }

    }
}
```

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Json;
using System.Text;
using System.Threading.Tasks;

namespace Assignment_2_WebApi_Client
{
    internal class Program
    {
        static async Task Main(string[] args)
        {
            string apiBaseUrl = "http://localhost:5099/api/employee";
            List<Employee> employees = new List<Employee>();
            HttpClient client = new HttpClient();
            try
            {
                employees = await client.GetFromJsonAsync<List<Employee>>(apiBaseUrl);
                foreach (var item in employees)
                {
                    Console.WriteLine($"Employee ID:{item.Id}\nName:{item.Name}\nDepartment:{item.Department}\nMobile:{item.MobileNo}\nEmail:{item.Email}\n-------------------");
                }
            }
            catch (HttpRequestException ex)
            {
                Console.WriteLine(ex);
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex);
            }
        }
    }
}

```