

---

# Task 2: Web API with Swagger and Postman Integration

---

## Objectives

1. Demonstrate installation and configuration of Swagger in a .NET Core Web API project.
  2. Explain and demonstrate the usage of Postman to test Web API endpoints.
  3. Show the usage of route customization and the ActionName attribute in controller methods.
- 

## Explanation

### 1. Swagger Integration

Swagger is an API documentation and testing tool. It auto-generates interactive documentation for APIs in .NET Core through the `Swashbuckle.AspNetCore` package. It provides a user-friendly UI for exploring endpoints.

### 2. Postman Usage

Postman is an external API testing tool that allows testing of HTTP methods (GET, POST, PUT, DELETE), adding custom headers like Authorization tokens, and sending/receiving JSON payloads. Postman also enables the creation of collections and saving test cases.

### 3. Route Customization and ActionName

Using `[Route("api/Emp")]`, the controller endpoint is customized for ease of access. The `[ActionName("...")]` attribute can be used to differentiate methods that use the same HTTP verb.

---

# Implementation

## Swagger Installation

Install Swagger using NuGet:

```
dotnet add package Swashbuckle.AspNetCore
```

## Configure Swagger in Program.cs

```
builder.Services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new OpenApiInfo
    {
        Title = "Swagger Demo",
        Version = "v1",
        Description = "TBD",
        TermsOfService = new Uri("https://example.com/terms"),
        Contact = new OpenApiContact
        {
            Name = "John Doe",
            Email = "john@xyzmail.com",
            Url = new Uri("https://example.com")
        },
        License = new OpenApiLicense
        {
            Name = "License Terms",
            Url = new Uri("https://example.com")
        }
    });
});

app.UseSwagger();
app.UseSwaggerUI(c =>
{
    c.SwaggerEndpoint("/swagger/v1/swagger.json", "Swagger Demo");
});
```

---

## Modify EmployeeController

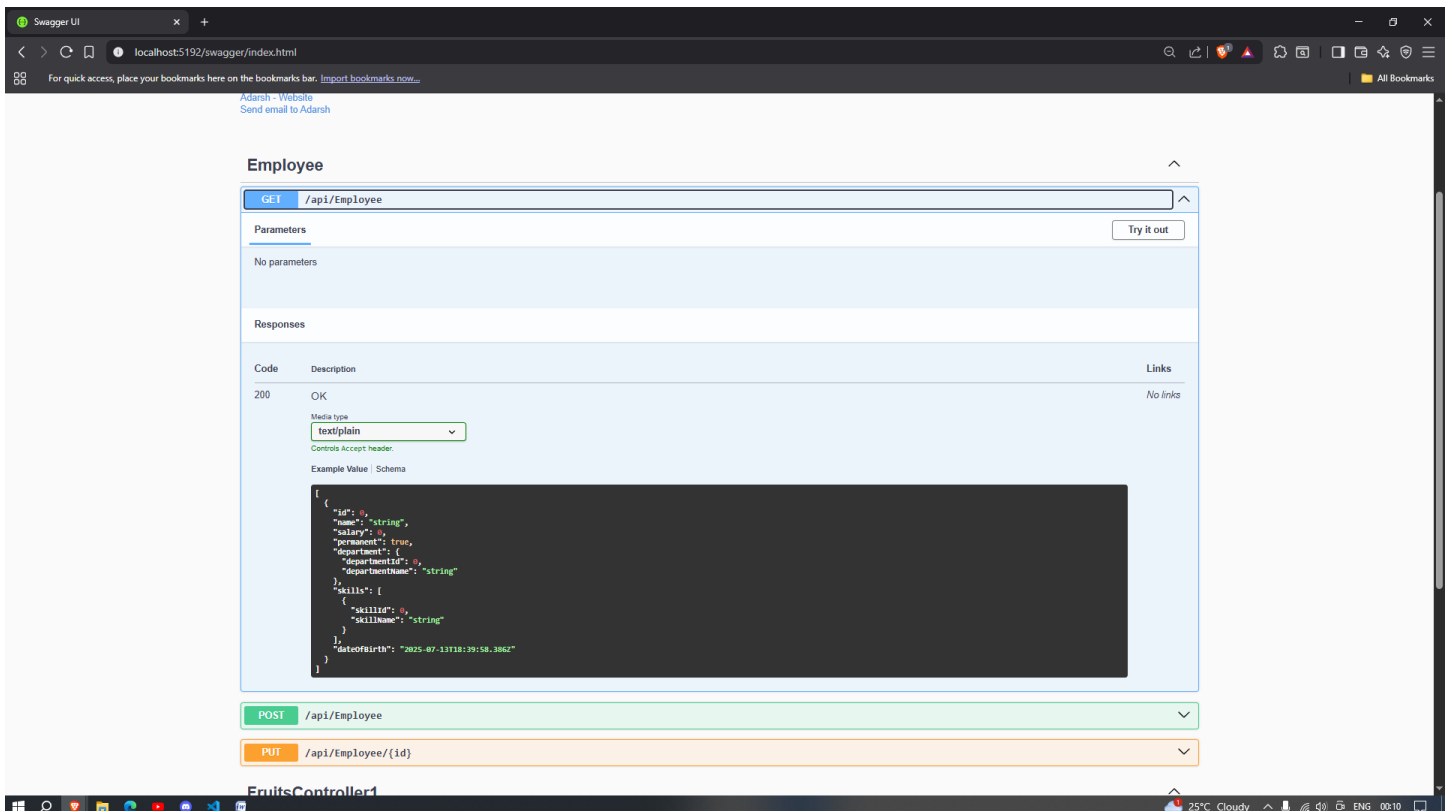
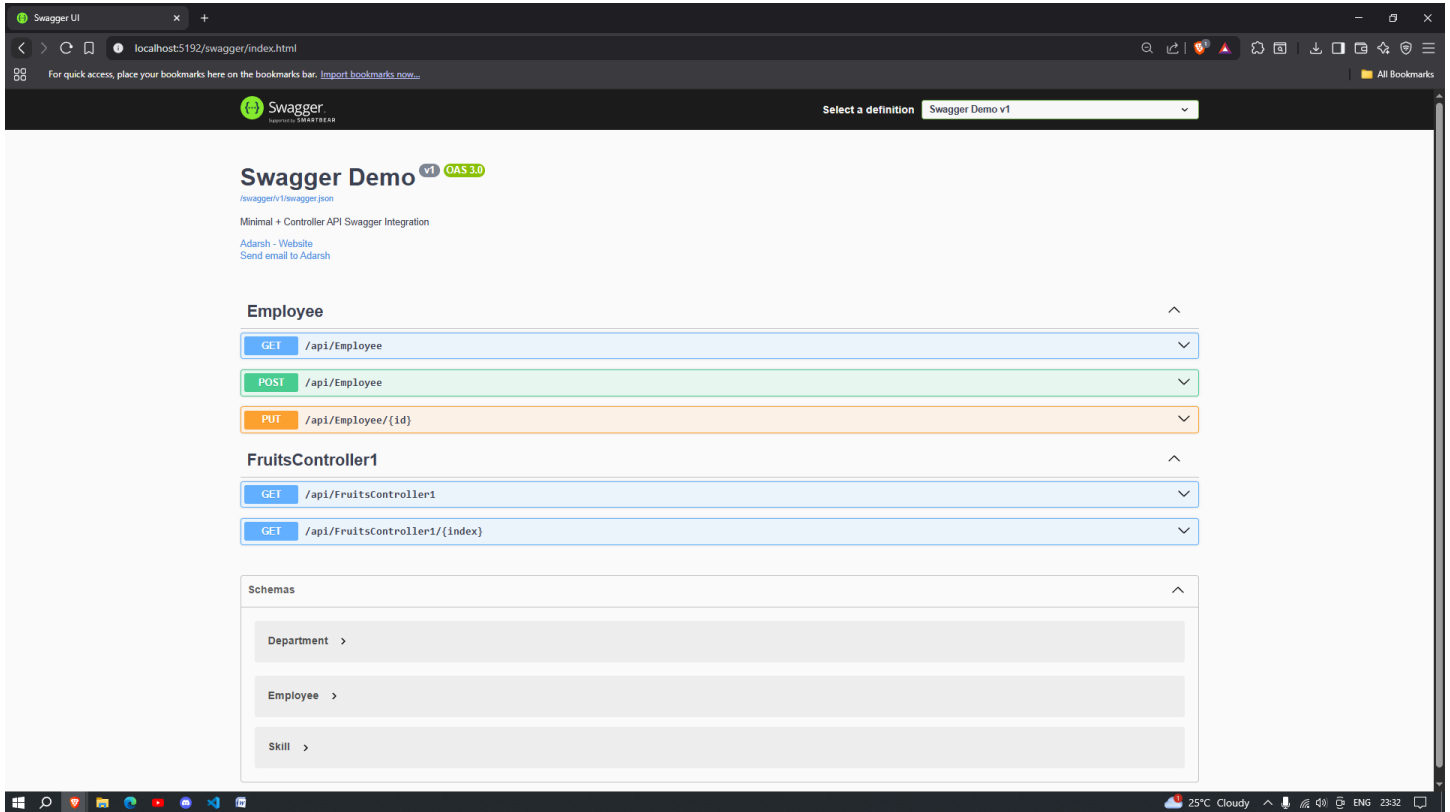
```
[ApiController]
[Route("api/Emp")]
public class EmployeeController : ControllerBase
{
    [HttpGet]
    [ProducesResponseType(typeof(List<Employee>), 200)]
    public ActionResult<List<Employee>> GetEmployees()
    {
        return Ok(new List<Employee>
        {
            new Employee
            {
                Id = 1,
                Name = "Adarsh",
                Salary = 50000,
                Permanent = true,
                Department = new Department { DepartmentId = 1, DepartmentName = "IT" },
                Skills = new List<Skill> { new Skill { SkillId = 1, SkillName = "C#" } },
                DateOfBirth = new DateTime(1995, 01, 01)
            }
        });
    }
}
```

---

## Postman Testing Instructions

1. Open Postman.
2. Set Method: GET
3. Set URL: `http://localhost:<port>/api/Emp`
4. Click Send.
5. Verify:
  - o Status code is 200 OK
  - o Response body contains the list of employees in JSON format.

# Screenshots



## Conclusion

This task demonstrates how Swagger helps in documenting and testing APIs interactively. It also shows the practical use of Postman for manual API testing, as well as the importance of customizing routes for better API usability.