Aman Raj

Superset ID: 6358186

1. **ASP.NET Core 8.0 Web API**

**RESTful Web Service, Web API & Microservice:-**

**RESTful Web Service**

A RESTful Web Service follows the REST architecture which stands for Representational State Transfer. It uses standard HTTP methods (GET, POST, PUT, DELETE) to access and manipulate resources.

Key Features of REST:

* Stateless: Each HTTP request is independent and doesn’t store user context on the server.

Client-Server: Separation of concerns – client and server are independent.

* Cacheable: Responses can be cached for better performance.
* Uniform Interface: URI for resource identification, HTTP verbs for actions.
* Messages: Data is usually sent in JSON or XML format.

**Web API**

Web API is a framework for building HTTP-based services that expose business logic/data to clients.

* It follows REST principles.
* In ASP.NET Core, it's a set of controllers returning JSON.
* Web API is not restricted to return XML (like traditional SOAP-based services); it returns JSON by default, but can also return XML or custom formats.

**Microservices**

Microservices are an architectural style where a large application is composed of small, independent services that communicate via APIs.

* Each microservice has its own data source, business logic, and deployment lifecycle.
* Promotes loose coupling and high scalability.

**HttpRequest & HttpResponse:-**

**HttpRequest**

* Represents the incoming request from a client.
* Contains: URL, Headers, Cookies, Query Strings, Body, Method (GET/POST/etc.)

**HttpResponse**

* Represents the server’s response sent to the client.
* Contains: StatusCode, Headers, Cookies, Body.

**First Web Api using .Net core**

**ValuesController.cs**

using Microsoft.AspNetCore.Mvc;

// For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

namespace MyFirstWebApi.Controllers

{

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

[ApiController]

public class ValuesController : ControllerBase

{

// GET: api/<ValuesController>

[HttpGet]

public IEnumerable<string> Get()

{

return new string[] { "value1", "value2" };

}

// GET api/<ValuesController>/5

[HttpGet("{id}")]

public string Get(int id)

{

return "value";

}

// POST api/<ValuesController>

[HttpPost]

public void Post([FromBody] string value)

{

}

// PUT api/<ValuesController>/5

[HttpPut("{id}")]

public void Put(int id, [FromBody] string value)

{

}

// DELETE api/<ValuesController>/5

[HttpDelete("{id}")]

public void Delete(int id)

{

}

}

}

**Output:-**

