

## HTTP Methods in ASP.NET Core REST API

In RESTful APIs, **HTTP Methods** represent the **action** you want to perform on a resource.

Think of your API resource as a data item — for example, a **Product**.

HTTP Method Meaning	Typical Use
<b>GET</b>	Retrieve data
<b>POST</b>	Create new data
<b>PUT</b>	Update existing data (full update)
<b>DELETE</b>	Remove data
	Replace product details
	Delete product by Id

---

### 1. GET – Read Data

**Purpose:** Fetch one or many resources from the server.

**Safe & Idempotent:** Does not change data.

#### ✓ Example in ASP.NET Core

```
[HttpGet]  
  
public IActionResult GetAllProducts()  
{  
    var products = _service.GetProducts();  
    return Ok(products);  
}
```

```
[HttpGet("{id}")]  
  
public IActionResult GetProductById(int id)  
{  
    var product = _service.GetProduct(id);  
    if (product == null) return NotFound();  
}
```

```
        return Ok(product);  
    }  

```

### 👉 When to use GET

- Fetch product list
  - Fetch product by ID
  - Search/filter items
- 

## 2. POST – Create New Data

**Purpose:** Add a new resource to the server.

**Not Idempotent:** Multiple POST calls create multiple records.

### ✓ Example in ASP.NET Core

```
[HttpPost]
```

```
public IActionResult CreateProduct(ProductDto dto)  
{  
    var product = _service.CreateProduct(dto);  
    return CreatedAtAction(nameof(GetProductById), new { id = product.Id }, product);  
}
```

### 👉 When to use POST

- Create a new product
  - Register user
  - Place an order
- 

## 3. PUT – Update Existing Data (Full Update)

**Purpose:** Replace an entire resource.

**Idempotent:** Multiple calls produce the same result.

### ✓ Example in ASP.NET Core

```
[HttpPut("{id}")]  
  
public IActionResult UpdateProduct(int id, ProductDto dto)  
{  
  
    var updated = _service.UpdateProduct(id, dto);  
  
    if (updated == null) return NotFound();  
  
  
    return Ok(updated);  
}
```

#### 👉 When to use PUT

Use PUT when the **entire object** must be updated.

---

## 4. DELETE – Remove Data

**Purpose:** Delete a resource permanently.

#### ✓ Example in ASP.NET Core

```
[HttpDelete("{id}")]  
  
public IActionResult DeleteProduct(int id)  
{  
  
    var deleted = _service.DeleteProduct(id);  
  
    if (!deleted) return NotFound();  
  
  
    return NoContent();  
}
```

#### 👉 When to use DELETE

- Delete product by ID
- Cancel order
- Remove user

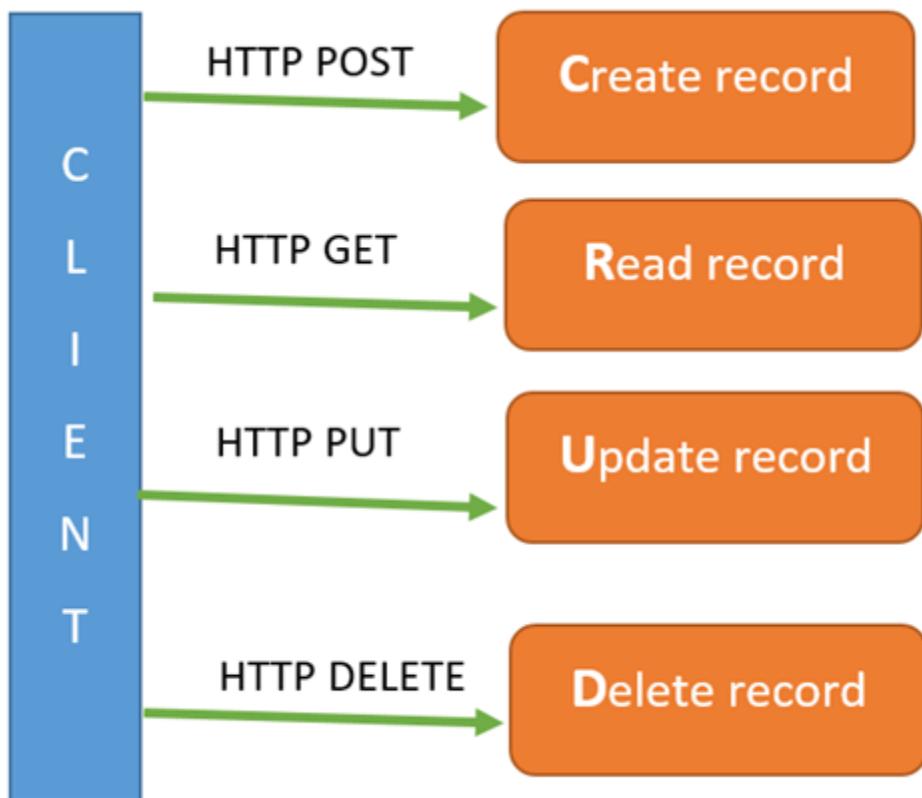
---

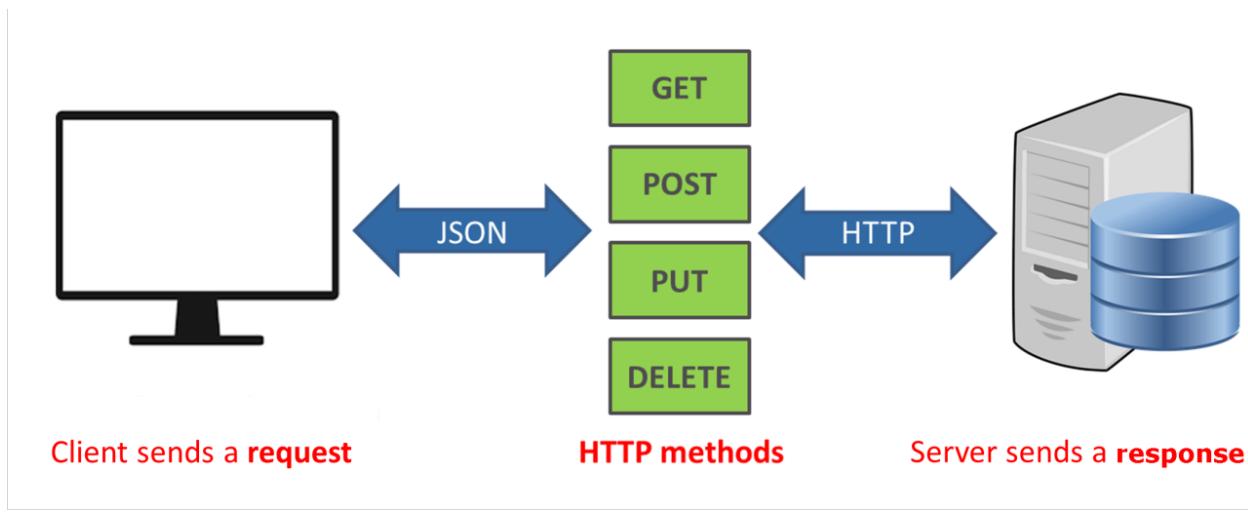
## Summary Table

Operation	HTTP Method	URI Example	Request Body	Response Example
Get all products	<b>GET</b>	/api/products	 No	200 OK
Get by ID	<b>GET</b>	/api/products/10	 No	200 OK / 404
Create product	<b>POST</b>	/api/products	 Yes	201 Created
Update product	<b>PUT</b>	/api/products/10	 Yes	200 OK / 404
Delete product	<b>DELETE</b>	/api/products/10	 No	204 NoContent

---

## REST API Flow Diagram





### 🎯 Mini Practical Demo (E-commerce)

**Controller Name:**

ProductsController.cs

**Route:**

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

```
[ApiController]
```

```
public class ProductsController : ControllerBase
```

Inside controller:

- ✓ GET → /api/products
- ✓ GET → /api/products/5
- ✓ POST → /api/products
- ✓ PUT → /api/products/5
- ✓ DELETE → /api/products/5