

## Action Results in ASP.NET Core (Instructor Explanation)

In ASP.NET Core MVC/Web API, every controller action **returns something to the client**. This return value is called an **Action Result**.

There are **three main types**:

1. **Return Simple Data (non-action results)**
    - Example: string, int, List<Product>
    - ASP.NET Core wraps them automatically into 200 OK.
  2. **IActionResult** (common in Web API)
    - You return different result types: Ok(), NotFound(), BadRequest(), etc.
  3. **ActionResult<T>**
    - Combines benefits of both: Strong typing + multiple status codes.
    - **Recommended** for APIs.
- 

## Why Do We Use Action Results?

Because APIs must return:

- ✓ HTTP status code
- ✓ Response body
- ✓ Optional headers

Example:

HTTP/1.1 404 Not Found

```
{  
  "message": "Product not found"  
}
```

So Action Results give **full control**.

---

## 1. IActionResult (Flexible but not strongly typed)

```
[HttpGet("{id}")]  
public IActionResult GetProduct(int id)
```

```
{  
    var product = _repo.Get(id);  
  
    if (product == null)  
        return NotFound();  
  
    return Ok(product);  
}
```

- 
- ✓ Can return any result type
  - ✗ Not strongly typed

---

## 📌 2. ActionResult<T> (Best for Web APIs)

```
[HttpGet("{id}")]  
public ActionResult<Product> GetProduct(int id)  
{  
    var product = _repo.Get(id);  
  
    if (product == null)  
        return NotFound("Product not found");  
  
    return Ok(product);  
}
```

- ✓ Strong return type (Product)
  - ✓ API docs (Swagger) show correct response formats
  - ✓ Most recommended approach
-

### 3. Returning Simple Types (Not recommended for production)

[HttpGet]

```
public string Ping() => "API is running";
```

Automatically becomes:

200 OK

"API is running"

---

## Common Action Results (With examples)

### ✓ 1. Ok() – 200

```
return Ok(products);
```

### ✓ 2. Created() / CreatedAtAction() – 201

Used for POST create operations.

```
return CreatedAtAction(nameof(GetProduct), new { id = product.Id }, product);
```

### ✓ 3. NoContent() – 204

For successful update/delete without response body.

```
return NoContent();
```

### ✓ 4. BadRequest() – 400

```
return BadRequest("Invalid input");
```

### ✓ 5. Unauthorized() – 401

```
return Unauthorized();
```

### ✓ 6. Forbidden() – 403

```
return Forbid();
```

### ✓ 7. NotFound() – 404

```
return NotFound("Product not found");
```

### ✓ 8. Conflict() – 409

```
return Conflict("Duplicate record");

✓ 9. StatusCode() – Custom

return StatusCode(500, "Internal server error");
```

---

### Real Controller Example (without response wrapper)

```
[ApiController]

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

public class ProductsController : ControllerBase
{
    private readonly IProductRepository _repo;

    public ProductsController(IProductRepository repo)
    {
        _repo = repo;
    }

    [HttpGet]
    public ActionResult<IEnumerable<Product>> GetAll()
    {
        return Ok(_repo.GetAll());
    }

    [HttpGet("{id}")]
    public ActionResult<Product> Get(int id)
    {
        var product = _repo.Get(id);
```

```
    if (product == null)
        return NotFound();

    return Ok(product);
}

[HttpPost]
public ActionResult<Product> Create(Product product)
{
    _repo.Add(product);
    return CreatedAtAction(nameof(Get), new { id = product.Id }, product);
}

[HttpPut("{id}")]
public IActionResult Update(int id, Product product)
{
    if (id != product.Id)
        return BadRequest("Id mismatch");

    var existing = _repo.Get(id);
    if (existing == null)
        return NotFound();

    _repo.Update(product);
    return NoContent();
}
```

```

    }

[HttpDelete("{id}")]
public IActionResult Delete(int id)
{
    var existing = _repo.Get(id);
    if (existing == null)
        return NotFound();

    _repo.Delete(id);
    return NoContent();
}

```

---

### **Summary Table**

<b>Scenario</b>	<b>Best Action Result</b>
Return a list/object	Ok()
Resource created	CreatedAtAction()
Input validation failure	BadRequest()
Record not found	NotFound()
Authentication required	Unauthorized()
No return body required	NoContent()
Custom error	StatusCode()