|  |  |
| --- | --- |
| Action Result .Net Core |  |

When we have .net core template we have Action methods like below, we have return type IActionResult

// GET: api/employees

[HttpGet]

public IActionResult GetAll()

{

var employees = \_service.GetAll();

return Ok(employees);

}

Let’s discuss

**Common Implementations of IActionResult**

When you return IActionResult from a controller method, you can return a variety of result types, including:

| **Type** | **Description** |
| --- | --- |
| Ok() | Returns HTTP 200 with optional content. |
| NotFound() | Returns HTTP 404. |
| BadRequest() | Returns HTTP 400. |
| Unauthorized() | Returns HTTP 401. |
| Forbid() | Returns HTTP 403. |
| Created() | Returns HTTP 201 (used when creating a resource). |
| NoContent() | Returns HTTP 204. |
| Content() | Returns raw content (e.g., string, HTML). |
| File() | Returns a file for download or inline viewing. |
| Redirect() | Issues a redirect to another URL. |
| View() | Returns a view (in MVC). |
| Json() | Returns JSON-formatted data. |
| StatusCode() | Returns any arbitrary status code. |

**Code Example**

|  |
| --- |
| public IActionResult GetItem(int id)  {  var item = \_repository.GetItemById(id);  if (item == null)  return NotFound();  return Ok(item);  } |

In this example:

* The method returns NotFound() if no item is found.
* It returns Ok(item) if the item exists.

## ActionResult<T>

This is a generic version of ActionResult introduced in **ASP.NET Core 2.1+**. It lets you return either a type T or any other action result (like NotFound()).

csharp

CopyEdit

[HttpGet("{id}")]

public ActionResult<Product> Get(int id)

{

var product = \_repository.Find(id);

if (product == null)

return NotFound();

return product; // Returns 200 with serialized product

}

The return type is ActionResult<Product>:

* return product; automatically becomes Ok(product)
* You can still use return NotFound();, return BadRequest();, etc.

## IActionResult vs ActionResult<T>

| **IActionResult** | **ActionResult<T>** |
| --- | --- |
| More flexible, returns any result | Strongly typed, ideal for APIs |
| Less useful for Swagger/OpenAPI | Improves OpenAPI docs and tooling |
| No type enforcement | Type-safe response |

### Returning Errors with Details

return BadRequest(new { message = "Invalid ID." });

## Best Practices

1. **Use ActionResult<T> for APIs** – it’s more descriptive and improves Swagger/OpenAPI generation.
2. **Use meaningful status codes** – don't just return 200 OK for everything.
3. **Centralize error handling** – use middleware like UseExceptionHandler.
4. **Use CreatedAtAction for POST methods** – helps clients discover the new resource location.