✅ 1. Model Binding in ASP.NET Core

⭐ What is Model Binding?

Model Binding is the process where ASP.NET Core **automatically maps incoming HTTP request data** (route values, query string, headers, form-data, JSON body) **to action method parameters**.

Example

POST /api/products (JSON body):

```
{
  "id": 1,
  "name": "Laptop",
  "price": 55000
}
```

Controller Action:

```
[HttpPost]
public IActionResult Create(Product product)
{
    // 'product' object is automatically filled by Model Binding
    return Ok(product);
}
```

✓ No conversion or parsing required
✓ Works with primitive types, complex objects, collections, form-data, JSON
✓ ASP.NET will map JSON → Product model automatically

🎯 How Model Binding Works Internally

ASP.NET Core tries to bind data in this order:

1. **Route values**
2. **Query string values**

3. **Form data**
4. **JSON Body**
5. **Headers, cookies**, etc.

🚀 Example with different types of bindings

```
[HttpGet("{id}")]

public IActionResult Get(int id, [FromQuery] string sort)

{

   // id comes from route

   // sort comes from query string

   return Ok($"Id = {id}, Sort = {sort}");

}
```

✅ 2. Model Validation in ASP.NET Core

Model Validation automatically checks whether the incoming data **meets the required rules** defined on your model.

Validation is triggered **after model binding**.

ASP.NET adds errors into:

ModelState

🔍 Example

```
[HttpPost]

public IActionResult Create(Product product)

{

   if (!ModelState.IsValid)

       return BadRequest(ModelState);
```

```
    return Ok("Product Created");
}
```

🏷️ 3. Data Annotations

Data Annotations are **attributes** placed on model properties to enforce rules like:

- Required fields
- Min/Max Length
- Range
- Custom validation
- Formatting

📌 Most Common Data Annotations (with examples)

🔸 [Required]

```
[Required]
public string Name { get; set; }
```

🔸 [StringLength]

```
[StringLength(50, MinimumLength = 3)]
public string Name { get; set; }
```

🔸 [Range]

```
[Range(1, 99999)]
public decimal Price { get; set; }
```

🔸 [EmailAddress]

```
[EmailAddress]
public string Email { get; set; }
```

🔸 **[Phone]**

[Phone]

public string ContactNumber { get; set; }

🔸 **[RegularExpression]**

[RegularExpression(@"^[A-Za-z]+$")]

public string Category { get; set; }

🔸 **[Compare]**

Useful for password confirmation.

[Compare("Password")]

public string ConfirmPassword { get; set; }

🌟 Complete Product Model Example (With Data Annotations)

```
public class Product

{

   public int Id { get; set; }


   [Required(ErrorMessage = "Product Name is mandatory")]

   [StringLength(50, MinimumLength = 3)]

   public string Name { get; set; }


   [Required]

   [Range(1, 100000, ErrorMessage = "Price must be between 1 and 100000")]

   public decimal Price { get; set; }


   [StringLength(100)]

   public string Description { get; set; }
```

```
    [Required]

    [Range(1, 500)]

    public int Stock { get; set; }

}
```

🧪 Controller Example With Model Binding + Validation

```
[HttpPost]

public IActionResult Create([FromBody] Product product)

{

    if (!ModelState.IsValid)

        return BadRequest(ModelState);


    return Ok("Product Created Successfully!");

}
```

🔥 Summary Table

Concept	Purpose	Happens When?	Example
Model Binding	Map HTTP data → C# objects	Before action executes	JSON → Product
Model Validation	Check if data matches rules	After binding	Required, Range
Data Annotations	Attributes defining rules	During validation	[Required], [Range]