

Setting up for Validation

We can add rules to model classes to make certain fields required if we like. Also, we can force a certain field to be in a certain format and supply the error message.

Here is an example for a model called Customer:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations; ← You must add this when adding validation code or
                                         or you will get an error.
namespace WebApplication3.Models
{
    public class Customer
    {
        public int CustomerID { get; set; }
        [Required(ErrorMessage = "Please enter your name.")]
        public string Name { get; set; }

        [Required(ErrorMessage = "Please enter your email address.")]
        [EmailAddress(ErrorMessage = "Please enter a valid email address")]
        public string Email { get; set; }

        [Required(ErrorMessage = "Please enter your phone number.")]
        [Phone(ErrorMessage = "Please enter a valid phone number.")]
        public string Phone { get; set; }
    }
}
```

<https://docs.microsoft.com>

- `[CreditCard]` : Validates the property has a credit card format.
- `[Compare]` : Validates two properties in a model match.
- `[EmailAddress]` : Validates the property has an email format.
- `[Phone]` : Validates the property has a telephone format.
- `[Range]` : Validates the property value falls within the given range.
- `[RegularExpression]` : Validates that the data matches the specified regular expression.
- `[Required]` : Makes a property required.
- `[StringLength]` : Validates that a string property has at most the given maximum length.
- `[Url]` : Validates the property has a URL format.

In MVC, validation happens on **both the client and server**.

Non-string value types (such as decimal, int, float, and DateTime) are inherently required (non-nullable) and don't need the Required attribute.