In this video, we will discuss using **StringLength**attribute. This attribute is present in **System.ComponentModel.DataAnnotations** namespace and is used to enforce minimum and maximum length of characters that are allowed in a data field. Let's understand this with an example.   
  
   
  
**We will be using table tblEmployee for this demo.**  
Create table tblEmployee  
(  
 Id int primary key identity(1,1),  
 Name nvarchar(50),  
 Email nvarchar(50),  
 Age int,  
 Gender nvarchar(50)  
)  
  
Insert into tblEmployee values('Sara Nan', 'Sara.Nani@test.com', 30, 'Female')  
Insert into tblEmployee values('James Histo', 'James.Histo@test.com', 33, 'Male' )  
Insert into tblEmployee values('Mary Jane', 'Mary.Jane@test.com', 28, 'Female' )  
Insert into tblEmployee values('Paul Sensit', 'Paul.Sensit@test.com', 29, 'Male' )  
  
Generate **ADO.NET entity data model** for table **tblEmployee**. Change the entity name from tblEmployee to Employee. Save and build the project.  
  
**Right click on the "Controllers" folder and select Add - Controller. Set**  
Name = HomeController  
Template = MVC controller with read/write actions and views, using Entity Framework  
Model class = Employee(MVCDemo.Models)  
Data Context Class = EmployeeContext(MVCDemo.Models)  
Views = Razor  
  
To validate data, use validation attributes that are found in System.ComponentModel.DataAnnotations namespace. It is not a good idea, to add these validation attributes to the properties of auto-generated **"Employee"** class, as our changes will be lost, if the class is auto-generated again.  
  
So, let's create another partial "Employee" class, and decorate that class with the validation attributes.  Right click on the "Models" folder and add Employee.cs class file. Copy and paste the following code.   
  
using System.ComponentModel.DataAnnotations;  
namespace MVCDemo.Models  
{  
    [MetadataType(typeof(EmployeeMetaData))]  
    public partial class Employee  
    {  
    }  
  
    public class EmployeeMetaData  
    {  
        [StringLength(10, MinimumLength = 5)]  
        [Required]  
        public string Name { get; set; }  
    }  
}  
  
Notice that, we have decorated **"Name"** property with **"StringLength"** attribute and specified Minimum and Maximum length properties. We also used **[Required]** attribute. So, at this point Name property is required and should be between 5 and 10 characters.  
  
**Points to remember:**  
**1.** **[StringLength]** attribute is present in System.ComponentModel.DataAnnotations namespace.  
**2.** **[StringLength]** attribute verifies that a string is of certain length, but does not enforce that the property is **REQUIRED**. If you want to enforce that the property is required use **[Required]** attribute.

**We will discuss the following attributes in our upcoming video sessions.**  
RegularExpression  
Range