Please watch [Part 90](http://csharp-video-tutorials.blogspot.com/2013/09/part-90-remote-validation-in-mvc-when.html) from [asp.net mvc tutorial](http://www.youtube.com/user/kudvenkat/videos?view=1&flow=grid), before proceeding.  
  
Out of the box, Remote attribute only works when JavaScript is enabled. If the end user, disables JavaScript on his/her machine then the validation does not work. This is because **RemoteAttribute**requires JavaScript to make an asynchronous AJAX call to the server side validation method. As a result, the user will be able to submit the form, bypassing the validation in place. This why it is always important to have server side validation.  
  
   
  
**To make server side validation work, when JavaScript is disabled, there are 2 ways**  
**1.** Add model validation error dynamically in the controller action method - Discussed in [Part 90](http://csharp-video-tutorials.blogspot.com/2013/09/part-90-remote-validation-in-mvc-when.html)  
**2.** Create a custom remote attribute and override IsValid() method  
  
**In this video, we will discuss, creating a custom remote attribute**  
**Step 1:** Right click on the project name in solution explorer and a folder with name = "Common"  
  
**Step 2:**Right click on the "Common" folder, you have just added and add a class file with name = RemoteClientServer.cs  
  
**Step 3:**Copy and paste the following code.   
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.Mvc;  
using System.ComponentModel.DataAnnotations;  
using System.Reflection;  
  
namespace MVCDemo.Common  
{  
    public class RemoteClientServerAttribute : RemoteAttribute  
    {  
        protected override ValidationResult IsValid(object value, ValidationContext validationContext)  
        {  
            // Get the controller using reflection  
            Type controller = Assembly.GetExecutingAssembly().GetTypes()  
                .FirstOrDefault(type => type.Name.ToLower() == string.Format("{0}Controller",

                    this.RouteData["controller"].ToString()).ToLower());  
            if (controller != null)  
            {  
                // Get the action method that has validation logic  
                MethodInfo action = controller.GetMethods()  
                    .FirstOrDefault(method => method.Name.ToLower() ==   
                        this.RouteData["action"].ToString().ToLower());  
                if (action != null)  
                {  
                    // Create an instance of the controller class  
                    object instance = Activator.CreateInstance(controller);  
                    // Invoke the action method that has validation logic

                    object response = action.Invoke(instance, new object[] { value });  
                    if (response is JsonResult)  
                    {  
                        object jsonData = ((JsonResult)response).Data;  
                        if (jsonData is bool)  
                        {  
                            return (bool)jsonData ? ValidationResult.Success :   
                                new ValidationResult(this.ErrorMessage);  
                        }  
                    }  
                }  
            }  
  
            return ValidationResult.Success;  
            // If you want the validation to fail, create an instance of ValidationResult  
            // return new ValidationResult(base.ErrorMessageString);  
        }  
  
        public RemoteClientServerAttribute(string routeName)  
            : base(routeName)  
        {  
        }  
  
        public RemoteClientServerAttribute(string action, string controller)  
            : base(action, controller)  
        {  
        }  
  
        public RemoteClientServerAttribute(string action, string controller,   
            string areaName) : base(action, controller, areaName)  
        {  
        }  
    }  
}  
  
**Step 4:** Open **"User.cs"** file, that is present in **"Models"**folder. Decorate **"UserName"**property with RemoteClientServerAttribute.   
  
RemoteClientServerAttribute is in MVCDemo.Common namespace, so please make sure you have a using statement for this namespace.  
public class UserMetadata  
{  
    [RemoteClientServer("IsUserNameAvailable", "Home",   
        ErrorMessage="UserName already in use")]  
    public string UserName { get; set; }  
}  
  
Disable JavaScript in the browser, and test your application. Notice that, we don't get client side validation, but when you submit the form, server side validation still prevents the user from submitting the form, if there are validation errors.