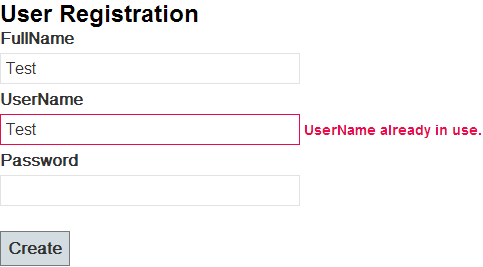
**Sometimes, to check if a field value is valid, we may need to make a database call.** A classic example of this is the user registration page. To register a user, we need a unique username. So, to check, if the username is not taken already, we have to make a call to the server and check the database table. RemoteAttribute is useful in situations like this.   
  
   
  
**Example:**When a user provides a username that already exists, the associated validation error message should be displayed immediately as shown below.   
   
  
**Step 1:** Create tblUsers table  
Create table tblUsers  
(  
 [Id] int primary key identity,  
 [FullName] nvarchar(50),  
 [UserName] nvarchar(50),  
 [Password] nvarchar(50)   
)  
  
**Step 2:** Create an ado.net entity data model using table **tblUsers**. Upon creating the entity model, change the name of the entity from to User. Save changes and build the solution.  
  
**Step 3:**Add HomeController with the following settings  
1. controller Name = HomeController  
2. Template = MVC controller with read/write actions and views, using Entity Framework  
3. Model Class = User (MVCDemo.Models)  
4. Data context class = SampleDBContext (MVCDemo.Models)  
5. Views = Razor  
  
**Step 4:**Copy and paste the following function in HomeController. This is the method which gets called to perform the remote validation. An AJAX request is issued to this method. If this method returns true, validation succeeds, else validation fails and the form is prevented from being submitted. The parameter name (UserName) must match the field name on the view. If they don't match, model binder will not be able bind the value with the parameter and validation may not work as expected.  
public JsonResult IsUserNameAvailable(string UserName)  
{  
    return Json(!db.Users.Any(x => x.UserName == UserName),  
                                         JsonRequestBehavior.AllowGet);  
}  
  
**Step 5:** Right click on the Models folder and a class file with name = User.cs. Copy and paste the following code. Notice that the name of the method (IsUserNameAvailable) and the controller name (Home) and error message are passed as arguments to Remote Attribute  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.ComponentModel.DataAnnotations;  
using System.Web.Mvc;  
  
namespace MVCDemo.Models  
{  
    [MetadataType(typeof(UserMetaData))]  
    public partial class User  
    {  
    }  
  
    public class UserMetaData  
    {  
        [Remote("IsUserNameAvailable", "Home", ErrorMessage="UserName already in use.")]  
        public string UserName { get; set; }  
    }  
}  
  
**Step 6:** Include references to the following css and script files in Create.cshtml view. **jQuery**, **jquery.validate**and **jquery.validate.unobtrusive**script files are required for remote validation to work.  
<link href="~/Content/Site.css" rel="stylesheet" type="text/css" />

<script src="~/Scripts/jquery-1.7.1.min.js" type="text/javascript"></script>  
<script src="~/Scripts/jquery.validate.min.js" type="text/javascript"></script>  
<script src="~/Scripts/jquery.validate.unobtrusive.js" type="text/javascript"></script>  
  
**Step 7:**Make sure **ClientValidation**and **UnobtrusiveJavaScript**are enabled in web.config  
<add key="ClientValidationEnabled" value="true" />  
<add key="UnobtrusiveJavaScriptEnabled" value="true" /> 