

Sign up for our free weekly [Web Developer Newsletter](#).



[articles](#) [Q&A](#) [forums](#) [lounge](#)



ASP.NET MVC Client Side Validation



Sandeep Singh Shekhawat, 1 Oct 2014 [CPOL](#)

Rate this:

★★★★★ 4.79 (43 votes)

This article explains how to implement client-side validation in an ASP.NET MVC application.

[ClientValidation.zip - 3.4 MB](#)

Introduction

This article explains how to implement client-side validation in an ASP.NET MVC application. The validation implemented using jQuery and jQuery validation plug-in (jquery.validate.min.js and jquery.validate.unobtrusive.min.js).

In the server-side validation ([ASP.NET MVC Server-Side Validation](#)), the page must be submitted via a postback to be validated on the server and if the model data is not valid then the server sends a response back to the client. With client-side validation, the input data is checked as soon as they are submitted, so there is no postback to the server and there is no page refresh.

Using the Code

When you are developing an MVC application in Visual Studio 2012 then the client-side becomes enabled by default, but you can easily enable or disable the writing of the following app setting code snippet in the web.config file.

[Hide](#) [Copy Code](#)

```
<configuration>
  <appSettings>
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  </appSettings>
</configuration>
```

The jQuery validation plug-in takes advantage of the Data Annotation attributes defined in the model, which means that you need to do very little to start using it. Let's create an **Employee** model (*Employee.cs* class file under the Models folder) that has two properties with Data annotation attributes.

[Hide](#) [Copy Code](#)

```
using System.ComponentModel.DataAnnotations;
namespace ClientValidation.Models
{
    public class Employee
    {
```

```

[Required(ErrorMessage = "Name is Requirde")]
public string Name { get; set; }
[Required(ErrorMessage = "Email is Requirde")]
[RegularExpression(@"^([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}" +
    @"\.[0-9]{1,3}\. [0-9]{1,3}\.)|([a-zA-Z0-9\-" +
    @".)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\?)$",
    ErrorMessage = "Email is not valid")]
public string Email { get; set; }
}

```

After that you need to create the controller's action methods. These render views on the UI and bind a model with the view. So let's create a controller with two action methods that handle both request types (GET and POST respectively).

[Hide](#) [Copy Code](#)

```

using System.Web.Mvc;
using ClientValidation.Models;
namespace ClientValidation.Controllers
{
    public class EmployeeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
        [AcceptVerbs(HttpVerbs.Post)]
        public ActionResult Index(Employee model)
        {
            if (ModelState.IsValid)
            {
                ViewBag.Name = model.Name;
                ViewBag.Email = model.Email;
            }
            return View(model);
        }
    }
}

```

Thereafter create a strong view that the input field binds with the model and create a JavaScript function that uses the HTML Helper extension method **FieldIdFor()** to get the Id of the input field and shows the input field's value in an alert message.

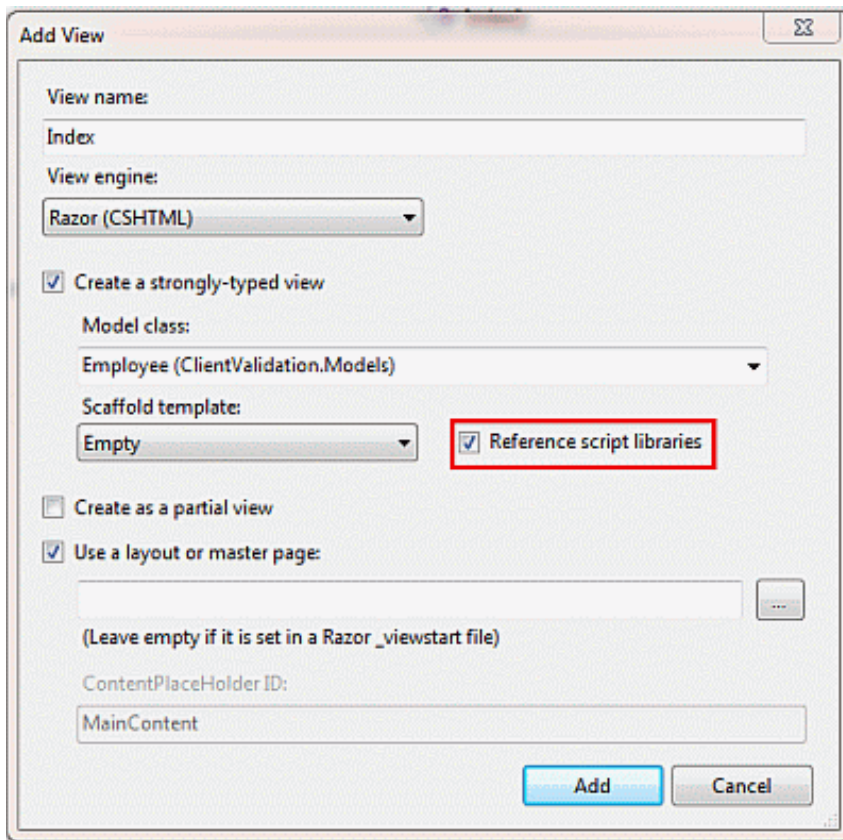


Figure 1.1 : Add New View Screen

Here I checked "Reference script libraries," which means Visual Studio adds these references automatically. Visual Studio adds the following code snippet to the bottom of the view.

Hide Copy Code

```
@section Scripts
{
    @Scripts.Render("~/bundles/jqueryval")
}
```

Now click on the "Add" button and the view is created as in the following code snippet:

Hide Shrink ▲ Copy Code

```
@model ClientValidation.Models.Employee
@{
    ViewBag.Title = "Index";
}
@if (ViewData.ModelState.IsValid)
{
    if(@ViewBag.Name != null)
    {

        Name : @ViewBag.Name
        Email : @ViewBag.Email

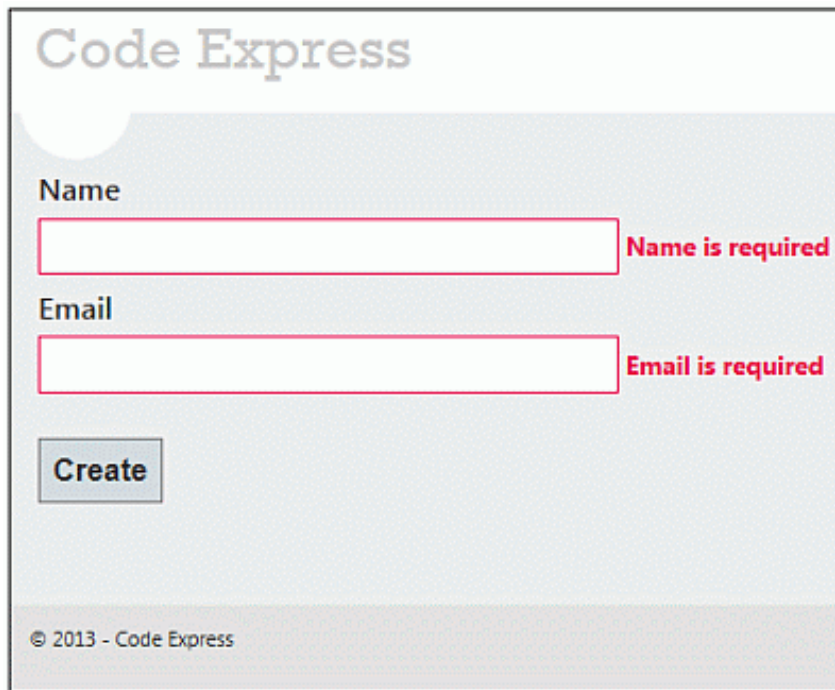
    }
}
@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)
    <fieldset>
        <legend>Employee</legend>
        <div class="editor-label">
            @Html.LabelFor(model => model.Name)
        </div>
```

```
<div class="editor-field">
    @Html.EditorFor(model => model.Name)
    @Html.ValidationMessageFor(model => model.Name)
</div>
<div class="editor-label">
    @Html.LabelFor(model => model.Email)
</div>
<div class="editor-field">
    @Html.EditorFor(model => model.Email)
    @Html.ValidationMessageFor(model => model.Email)
</div>
<p>
    <input type="submit" value="Create" />
</p>
</fieldset>
}
@section Scripts
{
    @Scripts.Render("~/bundles/jqueryval")
}
```

Output of Application

Let's run the application and test the following scenario.

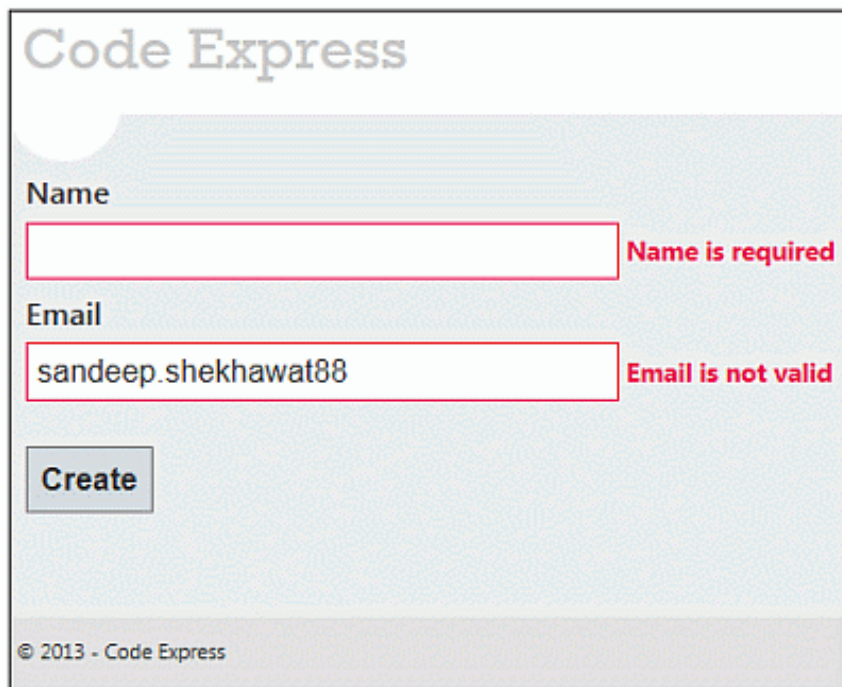
1. When all fields are empty:



The screenshot shows a web form titled "Code Express". It contains two input fields: "Name" and "Email". Both fields are empty and have a red border. To the right of each field is a red error message: "Name is required" and "Email is required". Below the fields is a "Create" button. At the bottom of the form, there is a copyright notice: "© 2013 - Code Express".

Figure 1.2: Validation Message when both fields are empty

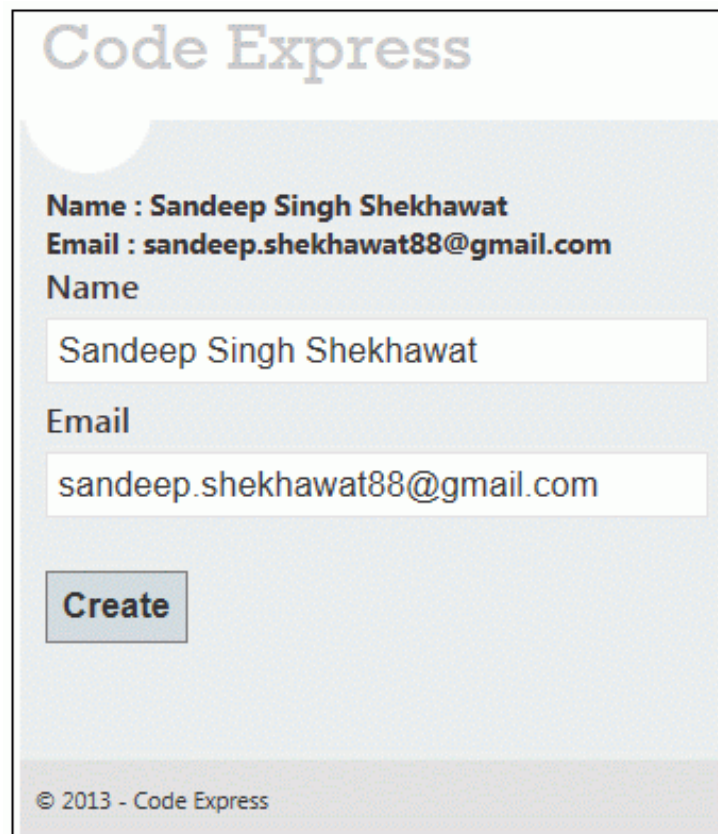
2. When the Name field is empty but Email is not valid:



The image shows a web form titled "Code Express" with a light blue background. It has two input fields: "Name" and "Email". The "Name" field is empty and has a red border with the text "Name is required" in red. The "Email" field contains the text "sandeep.shekhawat88" and has a red border with the text "Email is not valid" in red. Below the fields is a "Create" button. At the bottom, there is a footer that says "© 2013 - Code Express".

Figure 1.3 : Validation Message when Email is not valid

3. When both fields are valid:



The image shows the same "Code Express" web form, but now both fields are valid. The "Name" field contains the text "Sandeep Singh Shekhawat" and the "Email" field contains the text "sandeep.shekhawat88@gmail.com". Both fields have a light blue border. The "Create" button is still present. The footer at the bottom says "© 2013 - Code Express".

Figure 1.4 All fields are valid

License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](http://www.codeproject.com/Articles/718004/ASP-NET-MVC-Client-Side-Validation)

Share

EMAIL

TWITTER

About the Author

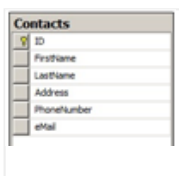


Sandeep Singh Shekhawat

 Software Developer
India 

Sandeep Singh Shekhawat is a Software Developer. He is awarded for C# Corner MVP(2013,2014 and 2015) and CodeProject MVP (2015). His three articles at CodeProject are prize winning in article competition. His more than 10 articles has been selected Article of the Day at Microsoft ASP.NET (<http://www.asp.net/>) community.

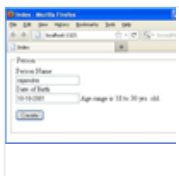
You may also be interested in...



A Beginner's Tutorial on Validating Model Data and Unobtrusive Client side Validation in ASP.NET MVC



Office 365 SharePoint Online Architectural considerations



Custom Client Side Validation in ASP.NET MVC3



Add HTML5 Document Viewer to ASP.NET MVC 5 Project



Enable or Disable ASP.NET Validation on Client Side



Increase System Availability by Leveraging Apache Tomcat Clustering

Comments and Discussions

You must [Sign In](#) to use this message board.

Search Comments

Go

First Prev Next

[My vote of 1] Not client side validation at all. 
Bo83 21-Sep-15 17:00

1

Not client side validation at all. 
Bo83 21-Sep-15 17:00

1

what is difference in coding for client side and server side validation using mvc razor 
Member 10374419 13-Jul-15 1:36

1

Good & Simple 
Slavimir Vesic 17-Feb-15 21:04

2

bundle not existant 
Member 10374498 9-Dec-14 9:22

2

Good One 
Shemeemsha RA 24-Sep-14 22:54

1

Client Side Validation 
Md Kamal Azhar 10-Jul-14 19:21

2

Validation 
NithyaGopu 31-Mar-14 20:34

2

My vote of 5 
Mihai MOGA 14-Feb-14 17:11

2

My vote of 5 
Renju Vinod 10-Feb-14 17:22

2

Awesome 
jjlangatjose 3-Feb-14 19:40

2

My vote of 4 
Pranay Rana 3-Feb-14 1:41

2

My vote of 5 
Humayun Kabir Mamun 29-Jan-14 18:13

2

Refresh

1

