**Understand FormCollection in MVC Controller**

This article shows the use of a FormCollection in the MVC architecture. Using the FomrCollection class we can capture the form's values within the controller.  
  
We know that there are various ways to collect data from a view to a collection, FomrCollection is one of them. We will implement a simple example of a FomrCollection shortly.  
  
To implement it we need to create a MVC application.  
  
**Here is the view to create the HTML form element**  
We are using a .aspx view in this application and a HTML helper class to generate HTML elements. If we look closely at the form element we will see that the form data will be submited for the “savedata” action defined within the “person” controller. Have a look at the following code.

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage<MVC\_Test.Models.person>" %>

<!DOCTYPE html>

<html>

<head runat="server">

    <meta name="viewport" content="width=device-width" />

    <title>Index</title>

</head>

<body>

    <div>

       <% using(Html.BeginForm("savedata","person")){ %>

        Name <%= Html.TextBox("name") %> <br />

        Surname : <%= Html.TextBox("surname") %> <br />

        Age: <%= Html.TextBox("age") %> <br />

        Gender <%= Html.RadioButton("gender","Male") %> Male

        <%= Html.RadioButton("gender","Female") %> Female <br />

        <input type="submit" name="Submit" value="Submit" />

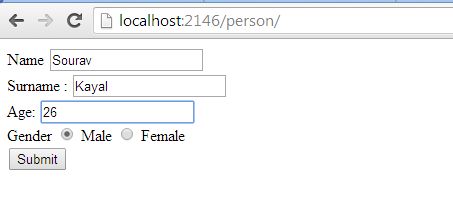
       <%} %>

    </div>

</body>

</html>

When we run the application, we will get the following output. And when the Submit button is pressed, the form data will be submitted to the targeted action.

  
We now need to create the controller that will return the view and handle the action. In this example we created a “person” controller with two actions. Once we run the person controller it will return an index view that is nothing but the HTML form that we designed.

 using System;  
using System.Collections.Generic;  
using System.Linq;

using System.Web;

using System.Web.Mvc;

using MVC\_Test.Models;

namespace MVC\_Test.Controllers

{

    public class personController : Controller

    {

        public ActionResult Index()

        {

            return View("Index");

        }

        public void savedata(FormCollection fomr)

        {

            string name = Request.Form["name"];

            string surname = Request.Form["surname"];

            string age = Request.Form["age"];

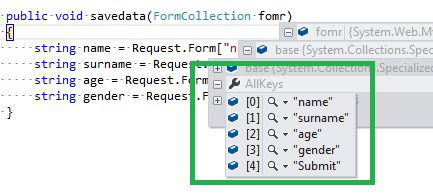
            string gender = Request.Form["gender"];

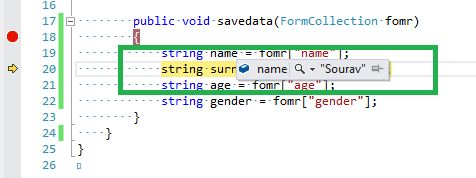
        }

    }

}

The savedata() action is taking an object of the FormCollection class. When we submit the form then all the control's information will be in the “form” object. Have a look at the following output.

  
  
We are seeing that the controls collection has been created within the “AllKeys” property. Now, if we want to access those form values then we need to use the form element by supplying the name in the object. Here we are getting all the form values.

  
  
**Conclusion**  
In this example we learned to collect all the form elements using the FormCollection class. This is one way to pass a value from a view to the controller.