<https://www.c-sharpcorner.com/UploadFile/3d39b4/getting-data-from-view-to-controller-in-mvc/>

How to pass data from View to Controllers

There are 4 different ways to pass Data from View to Controllers

1. By using Parameters

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace WebApplication15.Controllers

{

public class AddController : Controller

{

public ActionResult AddNumbers1()

{

return View();

}

// Pass Data as parameters

[HttpPost]

public ActionResult AddNumbers1(string txtNo1, string txtNo2)

{

int n1 = int.Parse(txtNo1);

int n2 = int.Parse(txtNo2);

int result = n1 + n2;

ViewBag.result = result;

return View();

}

}

}

@{

ViewBag.Title = "AddNumbers1";

}

<h2>AddNumbers1</h2>

@{ if (@ViewBag.result != null)

{

<b> Result is : @ViewBag.result </b>

}

else

{

<form method="post">

<label for="txtNo1">Enter No 1</label>

<input type="number" name="txtNo1" id="txtNo1" /> <br />

<label for="txtNo2">Enter No 2</label>

<input type="number" name="txtNo2" id="txtNo2" /> <br />

<button type="submit" value="Add"/>

</form>

}

}

1. Using Traditional Way in which we use Request Object

// 2nd way to pass data from view to controlletr to view using Request collection

public ActionResult AddNumbers2()

{

if (TempData["result"] != null)

ViewBag.result = TempData["result"];

return View();

}

[HttpPost]

public ActionResult AddNumbers()

{

int n1 = int.Parse(Request["txtNo1"]);

int n2 = int.Parse(Request["txtNo2"]);

int result = n1 + n2;

TempData["result"] = result;

return RedirectToAction("AddNumbers2");

}

------------------------------------------------------------------

@{

ViewBag.Title = "AddNumbers2";

}

<h2>AddNumbers2</h2>

@{ if (@ViewBag.result != null)

{

<b> Result is : @ViewBag.result </b>

}

else

{

using (Html.BeginForm("AddNumbers", "Add", FormMethod.Post))

{

@Html.Label("Enter No1")

@Html.TextBox("txtNo1");

<br />

@Html.Label("Enter No2");

@Html.TextBox("txtNo2");

<button type="submit"> Add </button>

}

}

}  
-----------------------------------------------------------------

1. Using Forms Collection

// 3 Way to pass data from View to Controller using FormsCollection

public ActionResult AddNumbers3()

{

return View();

}

[HttpPost]

public ActionResult AddNumbers3(FormCollection frm)

{

int n1 = Convert.ToInt16(frm["txtNo1"]);

int n2 = int.Parse(frm["txtNo2"]);

int result = n1 + n2;

ViewBag.result = result;

return View();

}

@{

ViewBag.Title = "AddNumbers2";

}

<h2>AddNumbers3</h2>

@{ if (@ViewBag.result != null)

{

<b> Result is : @ViewBag.result </b>

}

else

{

using (Html.BeginForm(FormMethod.Post))

{

@Html.Label("Enter No1")

@Html.TextBox("txtNo1");

<br />

@Html.Label("Enter No2");

@Html.TextBox("txtNo2");

<button type="submit"> Add </button>

}

}

}

1. Using Strongly typed View ( Using Model Class)

We need to make a class

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication15.Models

{

public class Numbers

{

public int No1 { get; set; }

public int No2 { get; set; }

}

}

// 4 way using Stronly typed View

public ActionResult AddNumbers4()

{

Numbers numbers = new Numbers();

return View(numbers);

}

[HttpPost]

public ActionResult AddNumbers4(Numbers numbers)

{

int n1 = numbers.No1;

int n2 = numbers.No2;

int result = n1 + n2;

ViewBag.result = result;

return View();

}

}

@model WebApplication15.Models.Numbers

@{

ViewBag.Title = "AddNumbers4";

}

<h2>AddNumbers4</h2>

@{ if (@ViewBag.result != null)

{

<b> Result is : @ViewBag.result </b>

}

else

{

using (Html.BeginForm(FormMethod.Post))

{

@Html.LabelFor(model => model.No1)

@Html.TextBoxFor(model => model.No1)

<br />

@Html.LabelFor(model => model.No2)

@Html.TextBoxFor(model => model.No2)

<br />

<button type="submit"> Add </button>

}

}

}