How do we add Controllers

Action Methods whose return type is ActionResult

How to pass data from controllers (Action methods) to View

* ViewBag, ViewData

How to pass data from action method to other action method

* TempData

**Today we saw,**

1. How to pass an object to View
2. How to pass data from view to controller

Controller >

Views >

Model > Model represents domain

**We want to display values of an object in View**

**Create a class in Models folder**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication14.Models

{

public class Student

{

public int Id { get; set; }

public string Name { get; set; }

public int Marks { get; set; }

public string BatchCode { get; set; }

}

}

**1st > Pass object in ViewBag**

**Action Method**

public ActionResult Display()

{

Student student = new Student() { Id = 1, Name = "Ajay", Marks = 90, BatchCode = "B001" };

ViewBag.student = student;

return View();

}

**View**

@{

ViewBag.Title = "Display";

}

<h2>Display</h2>

<h2> Studnet Details </h2>

ID : @ViewBag.student.Id

<br />

Name : @ViewBag.student.Name

<br />

Marks : @ViewBag.student.Marks

<br />

Batch Code : @ViewBag.student.BatchCode

<br />

**2nd > Pass List of Objects in ViewBag**

public ActionResult Index()

{

List<Student> students = new List<Student>()

{

new Student(){ Id=1, Name="Ajay", Marks=90, BatchCode="B001"},

new Student(){ Id=2, Name="Deepak", Marks=89, BatchCode="B002"},

new Student(){ Id=3, Name="Ajay", Marks=90, BatchCode="B001"},

new Student(){ Id=4, Name="Deepak", Marks=89, BatchCode="B002"},

new Student(){ Id=5, Name="Ajay", Marks=90, BatchCode="B001"},

new Student(){ Id=6, Name="Deepak", Marks=89, BatchCode="B002"},

new Student(){ Id=7, Name="Ajay", Marks=90, BatchCode="B001"},

new Student(){ Id=8, Name="Deepak", Marks=89, BatchCode="B002"},

};

ViewBag.students = students;

return View();

}

**View**

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

@{ foreach (var temp in ViewBag.students)

{

@temp.Id

@temp.Name

@temp.Marks

@temp.BatchCode

<br />

}

}

Now , we will pass object in the View Directly

Add (10,90);

Add(int x, int y) {}

ActionResult is a abstract base class for so many types

Public actionResult Index()

{

Return Content();

}

**3rd > Pass object directly to View**

public ActionResult Display1()

{

Student student = new Student() { Id = 1, Name = "Ajay", Marks = 90, BatchCode = "B001" };

return View(student);

}

View

@model WebApplication14.Models.Student

@{

ViewBag.Title = "Display1";

}

<h2>Student Details </h2>

ID : @Model.Id

<br />

Name : @Model.Name

<br />

Marks : @Model.Marks

<br />

Batch Code : @Model.BatchCode

<br />

When we pass object to View like this ,

We have to receive it(object) also in the view

@model WebApplication14.Models.Student

When you see this statement in any View, that View is ka Strongly typed View

Strongly typed view is a view which is based on an object

We can access properties through Model

Benefits of Strongly typed View >

We can access properties through Model object

It gives intellisense

**4th > Pass List of objects directly to View**

public ActionResult Index1()

{

List<Student> students = new List<Student>()

{

new Student(){ Id=1, Name="Ajay", Marks=90, BatchCode="B001"},

new Student(){ Id=2, Name="Deepak", Marks=89, BatchCode="B002"},

new Student(){ Id=3, Name="Ajay", Marks=90, BatchCode="B001"},

new Student(){ Id=4, Name="Deepak", Marks=89, BatchCode="B002"},

new Student(){ Id=5, Name="Ajay", Marks=90, BatchCode="B001"},

new Student(){ Id=6, Name="Deepak", Marks=89, BatchCode="B002"},

new Student(){ Id=7, Name="Ajay", Marks=90, BatchCode="B001"},

new Student(){ Id=8, Name="Deepak", Marks=89, BatchCode="B002"},

};

return View(students);

}

@model IEnumerable<WebApplication14.Models.Student>

@{

ViewBag.Title = "Index1";

}

<h2> List of Studnets </h2>

@{ foreach (var temp in Model)

{ @temp.Id

@temp.Name

@temp.Marks

@temp.BatchCode

<br />

}

}

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

**Display List in Formatted Way using table**

@model IEnumerable<WebApplication14.Models.Student>

@{

ViewBag.Title = "Index1";

}

<h2> List of Studnets </h2>

<table style="width:100%; height:80%; background-color:pink">

<tr>

<th> ID </th>

<th> Name </th>

<th> Marks </th>

<th> Batch Code </th>

</tr>

@{ foreach (var temp in Model)

{

<tr>

<td> @temp.Id </td>

<td> @temp.Name </td>

<td> @temp.Marks</td>

<td> @temp.BatchCode </td>

</tr>

}

}

</table>