File->New Project -> ASP.NET MVC App -> Empty , Check only MVC ,( Change Auth : No Authentication)

Стартираме -> няма нищо

Добавяме контролер –> Empty -> HomeController -> Пак стартираме – не може да намери View

Създаваме View – Index ->Стартираме

Обяснение за \_Layout и \_ViewStart. Закоментираме \_Layout във \_ViewStart и стартираме.

Project Properties -> Web -> Start section

Създаваме Action:

public ActionResult ShowCurrentDate()

{

ViewBag.Username = "Ivan";

//ViewBag.Username = System.Environment.UserName;

return View();

}

Създаваме и View:

Hello <b>@ViewBag.Username</b>

<p>

Current Date is <b>@DateTime.Now</b>

</p>

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

Създаваме Action:

public ActionResult ShowString()

{

// no need for view here

return Content("string value");

}

Създаваме и View: -без нищо

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

Създаваме Action:

public ActionResult GetUrlParameter(string username, string number = "N/A")

{

ViewBag.Username = username;

ViewBag.Number = number;

return View();

}

Създаваме и View:

<h2>Get Url Parameter</h2>

<p>

Име: <b>@ViewBag.Username</b>

<br/>

Номер: @ViewBag.Number

</p>

<p>

==============================

<br />

Test in URL:

<br />

@Html.ActionLink("/Home/GetUrlParameter?username=pesho&number=5", "GetUrlParameter", "Home", new { username ="pesho", number ="5" },null)

<br />

@Html.ActionLink("/Home/GetUrlParameter?username=pesho", "GetUrlParameter", "Home", new { username = "pesho"}, null)

<br />

@Html.ActionLink("/Home/GetUrlParameter?number=5", "GetUrlParameter", "Home", new { number = "5" }, null)

<br />

<br />

Another way to use parameters:

<br />

/Home/GetUrlParameter/?username=pesho

</p>

Стартираме :

/Home/GetUrlParameter?username=pesho

/Home/GetUrlParameter?username=pesho&number=5

/Home/GetUrlParameter/?username=pesho&number=5

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

//public ActionResult GetUrlParameter ([Bind(Prefix = "id")] string username) // .../Home/ GetUrlParameter /pesho

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

Създаваме Action:

public ActionResult ShowSubmitForm()

{

return View();

}

Създаваме и View:

<h2>Show Submit Form</h2>

@if (ViewBag.Message != "")

{

@ViewBag.Message

}

<form action="/Home/SaveSubmitForm" method="post">

<input type="text" name="username" />

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

</form>

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

Създаваме Action:

[HttpPost]

public ActionResult SaveSubmitForm(string username)

{

ViewBag.Username = username;

ViewBag.Message = "We added the user: " + username;

return View("ShowSubmitForm");

}

Задава се същото View

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

Създаваме Action:

public ActionResult ViewPhonebook()

{

return View();

}

Създаваме и View:

<h2>View Phonebook</h2>

@{

string[][] phonebook = new string[][]{

new string[]{ "Ivan", "111111"},

new string[]{ "Pesho", "22222"},

new string[]{ "Lili", "33333"}

};

}

@for (int i = 0; i < phonebook.Length; i++)

{

@phonebook[i][0] @:- @phonebook[i][1]

<text><br /></text>

@\*comment\*@

}

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

Models -> Add Class -> User

public class User

{

public string Username { get; set; }

public string Phonenumber { get; set; }

}

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

Създаваме Action:

public ActionResult ViewPhonebookStrongType()

{

User[] phonebook = new User[3];

phonebook[0] = new User { Username = "Ivan", Phonenumber = "08811111" };

phonebook[1] = new User { Username = "Pesho", Phonenumber = "08822222" };

User user = new User();

user.Username = "Lili";

user.Phonenumber = "08933333";

phonebook[2] = user;

return View(phonebook);

}

Създаваме и View, Empty

@using BaseSample1.Models

@model User[]

@{

ViewBag.Title = "View Phonebook StrongType";

}

<h2>View Phonebook Strongly typed</h2>

@for (int i = 0; i < Model.Length; i++)

{

@Model[i].Username @:- @Model[i].Phonenumber

<text><br /></text>

}

@\* with one more bracket

@{

for (int i = 0; i < Model.Length; i++)

{

@Model[i].Username @:- @Model[i].Phonenumber

<text><br /></text>

}

}

\*@

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

Добавяме още един Action

public ActionResult EditUser(string username)

{

User user = new Models.User() { Username = "Hristo", Phonenumber = "1111" };

return View(user);

}

Да отбележим че атрибута по default за http –винаги е get

Добавяме и view

@model BaseSample1.Models.User

@{

ViewBag.Title = "Edit User";

}

<h2>Edit User</h2>

@using (Html.BeginForm("EditUser", "Home"))

{

@: User: @Html.TextBoxFor(model => model.Username) <br />

@: Phonenumber: @Html.TextBoxFor(model => model.Phonenumber) <br />

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

@\*

@Html.TextBox("username", @Model.Username)

@Html.TextBox("Phonenumber", @Model.Phonenumber)

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

\*@

@\*

@Html.EditorFor(model => model.Username)

@Html.EditorFor(model => model.Phonenumber)

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

\*@

}

@\*

<form action="/Home/EditUser" method="post">

<input type="text" name="username" />

<input type="text" name="Phonenumber" />

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

</form>

\*@

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

Добавяме в контролера един Action

[HttpPost]

//public ActionResult EditUser(string username, string phonenumber)

public ActionResult EditUser(User user) // auto-mapping of object properties from the request

{

// here is important to notice that the User properties are updated

ViewBag.Message = string.Format("The new phone for {0} is {1}", user.Username, user.Phonenumber);

return View("EditUser");

}

Като във view-то EditUser.cshtml добавим след тага <h>Edit User</h>

@{

if (ViewBag.Message != null && ViewBag.Message != "")

{

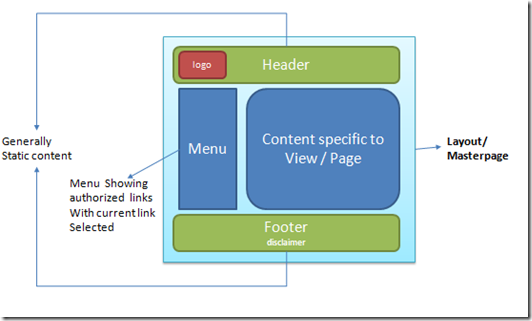
<b>@ViewBag.Message</b>

}

}

================================================================================

(<http://discusscode.blogspot.bg/2011/12/aspnet-mvc-3-razor-section-defined-but.html>)



===========================================================

Views/Shared -> Add View -> (partial) \_TopMenu

<div>

@Html.ActionLink("Home", "Index", "Home")

&nbsp;&nbsp;&nbsp;

@Html.ActionLink("Date", "ShowCurrentDate", "Home")

&nbsp;&nbsp;&nbsp;

@Html.ActionLink("UrlParameter", "GetUrlParameter", "Home")

&nbsp;&nbsp;&nbsp;

@Html.ActionLink("ShowSubmitForm", "ShowSubmitForm", "Home")

&nbsp;&nbsp;&nbsp;

@Html.ActionLink("Phonebook", "ViewPhonebook", "Home")

&nbsp;&nbsp;&nbsp;

@Html.ActionLink("ViewPhonebookStrongType", "ViewPhonebookStrongType", "Home")

&nbsp;&nbsp;&nbsp;

@Html.ActionLink("EditUser", "EditUser", "Home")

</div>

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

Във Index.cshtml добавяме:

<h2>Index</h2>

@section Footer

{

<p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>

}

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

\_Layout.cshtml

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<div>

@Html.Partial("\_TopMenu")

</div>

<div class="container body-content">

@RenderBody()

</div>

<div>

@RenderSection("Footer", false)

</div>

</body>

</html>

Разкоментираме \_Layout от \_ViewStart

Дебъгване!!

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

==================================================

Добавяме css и jquery в layout:

Отваряме файла : ~/Content/Site.css

body {

padding-top: 50px;

padding-bottom: 20px;

->>> background-color: #FFFFD7;

}

И в layout добавяме

<head>

<title>@ViewBag.Title - - My ASP.NET Application</title>

<link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />

</head>

Също в <head>

<script src="@Url.Content("~/Scripts/jquery-1.10.2.js")" type="text/javascript"></script>

А най-отдолу в layout след </html>:

<script>

$("document").ready(function(){

// alert("jquery is working");

});

</script>

Може да разкоментираме и без да спираме проекта да дадем F5 на браузъра.

* Изчистване на кеша на браузъра:

*F5 or Control + R = Reload the current page*

*Control + Shift + R or Control + F5 or Shift + F5 = Reload your current page, ignoring cached content*