مقدمه

در این مقاله موارد زیر را یاد خواهید گرفت.

1- TempData چیست و کاربرد آن چیست؟

2- نحوه انتقال داده بین دو کنترلر یا بین دو اکشن متد چگونه است؟

3- چگونه داده ها را در درخواست بعدی حفظ کنید؟

4- متد های Keep() و Peek() چه متد هایی هستند و چه کاربردهایی دارند؟

TempData چیست و کاربرد آن چیست؟

1- TempData بسیار شبیه ViewData است به جز آنکه TempData برای دو درخواست متوالی نگه داشته می شود تا با اینکار مدیریت داده ها بسیار ساده تر می شود به خصوص برای مواقعی که قصد جابجایی بین دو کنترلر یا بین دو اکشن متد را داشته باشیم. بنابراین TempData برای انتقال داده از یک درخواست HTTP به یک درخواست HTTP دیگر استفاده می شود.

2- TempData از مشتقات TempDataDictionary است و جز خصوصیات(property) کلاس BaseController به حساب می آید.

3- Typecasting برای داده های پیچیده، لازم است و توصیه می شود که برای پیشگیری از ارور، null نبودن مقادیر حتما بررسی شود.

4- TempData برای نگه داری داده از session استفاده می کند.

5- TempData به ما اجازه می دهد که اطاعات را بتوانیم نگه داریم، این اطلاعات تا درخواست بعدی نگه داری می شوند. ما ابتدا باید مقداری داده در داخل TempData ذخیره کنیم و سپس انتقال را انجام دهیم. در کنترلر می توانیم این مقادیری که در داخل TempData ذخیره شده بود را دریافت نماییم.

6- به صورت کلی از TempData برای ذخیره داده های یک بار مصرف مانند پیامهای مربوط به خطاها، پیام های اعتبارسنجی و ... استفاده می شود.

7- TempData در واقع یک نمونه با عمر کوتاه است به این معنا که چرخه حیات TempData آنچنان طولانی نیست. برای مثال چرخه حیات TempData از درخواست فعلی تا درخواست بعدی است.

8- از TempData وقتی استفاده کنید که از داده ها فقط در درخواست بعدی نیاز دارید.

نحوه انتقال داده بین دو کنترلر یا بین دو اکشن متد چگونه است؟

مثال زیر، نحوه کار با TempData را به خوبی نمایش می دهد.

روی پوشه Models راست کلیک کرده و از منوی Add یک کلاس با نام BookDetails.cs بسازید و قطعه کد زیر را همانند تصویر در آن وارد نمایید.

حالا یک کنترلر می سازیم و کدهایی برای ذخیره داده در TempData در داخل کنترل اضافه می کنیم، همانند آنچه در تصویر زیر آمده است.

در مثال بالا، ما شی book را در داخل TempData ذخیره کرده ایم این عمل ذخیره در اکشن متد Index انجام شده و در اکشن متد دیگری با نام DisplayBookData که وظیفه نمایش اطلاعات کتاب را دارد، مقادیر داخل TempData را دریافت می نماییم.

به نظر شما آیا می شود این فرایند را به کمک ViewBag یا ViewData انجام داد؟ خیر، چراکه اگر ما بخواهیم همین کار را با ViewBag یا ViewData انجام دهیم، در اکشن متد DisplayBookData با مقادیر null مواجه می شویم و به این دلیل فقط TempData برای جابجایی داده بین اکشن ها مناسب است.

سپس دو view برای اکشن متدهای Index و DisplayBookData می سازیم. نیازی نیست که کدی در داخل index.cshtml قرار دهیم، چراکه این متد قرار نیست اطلاعاتی به کاربر نمایش دهد. کدهای داخل DisplayBookData.cshtml را به همان شکلی که در تصویر پایین آمده است وارد می کنیم.

حالا اگر از برنامه خروجی بگیرید، خروجی به شکل تصویر زیر خواهد بود

چگونه داده ها را در درخواست بعدی حفظ کنید؟

رفتار های TempData (مثلا اینکه مقادیر را برای درخواست بعدی حفظ کند یا خیر)را می توان به دسته های زیر تقسیم کرد.

حالت 1- نخواندن در اولین درخواست

حالت 2- خواندن مقادیر در اولین درخواست

حالت 3- خواندن و حفظ مقادیر به کمک متد Keep

حالت 3- نگه داری داده به کمک متد های Peek و Keep

در اینجا به بررسی جزئیات حالت های ذکر شده می پردازیم

حالت 1- نخواندن در اولین درخواست

اگر ما در داخل اکشن متد، مقداری در TempData قرار دهیم(مانند TempData["BookName"]="Ramayana")، ولی این مقدار را در داخل view دریافت نکنیم، مقادیر TempData تا درخواست بعدی نگه داری می شوند و در درخواست بعدی قابل دسترسی خواهند بود.

حالت 2- اگر ما مقدار TempData را دریافت نماییم(مانند آنچه در پایین نمایش داده شده است)، TempData خالی می شود و در درخواست های بعدی مقدار Null برمی گرداند.

String bookName =TempData["BookName"].ToString()

حالت 3- خواندن و حفظ مقادیر به کمک متد Keep

اگر ما مقدار TempData را در درخواست فعلی دریافت کنیم و از متد Keep همانگونه که در پایین نمایش داده شده، استفاده کنیم. داده های TempData برای درخواست بعدی نیز قابل دسترسی خواهند بود.

String bookName = TempData["BookName"].ToString()

TempData.Keep();

نکته: نمونه ساده از متد Keep، تمامی شی های داخل TempData را نگه می دارد. اما اگر بخواهیم فقط یک مقدار خاص را نگه داریم، باید از overload متد Keep استفاده کنیم، در این overload از متد Keep ما می توانیم نام شی را مشخص کنیم. (مانند کد زیر)

TempData.Keep("BookName");

حالت 4- نگه داری داده به کمک متدهای Peek و Keep

اگر ما مقدار TempData را به کمک دستور Peek دریافت کنیم، مقدار TempData همچنان حفظ می شود و در درخواست بعدی قابل دسترسی خواهد بود. به این روش ما می توانیم مقدار TempData را بخوانیم و آنرا همچنان برای درخواست بعدی حفظ کنیم. نحوه استفاده از متد Peek در کد زیر نمایش داده شده است.

String bookName = TempData.Peek("BookName").ToString();

نکته: برای متد Peek هیچگونه overload وجود ندارد