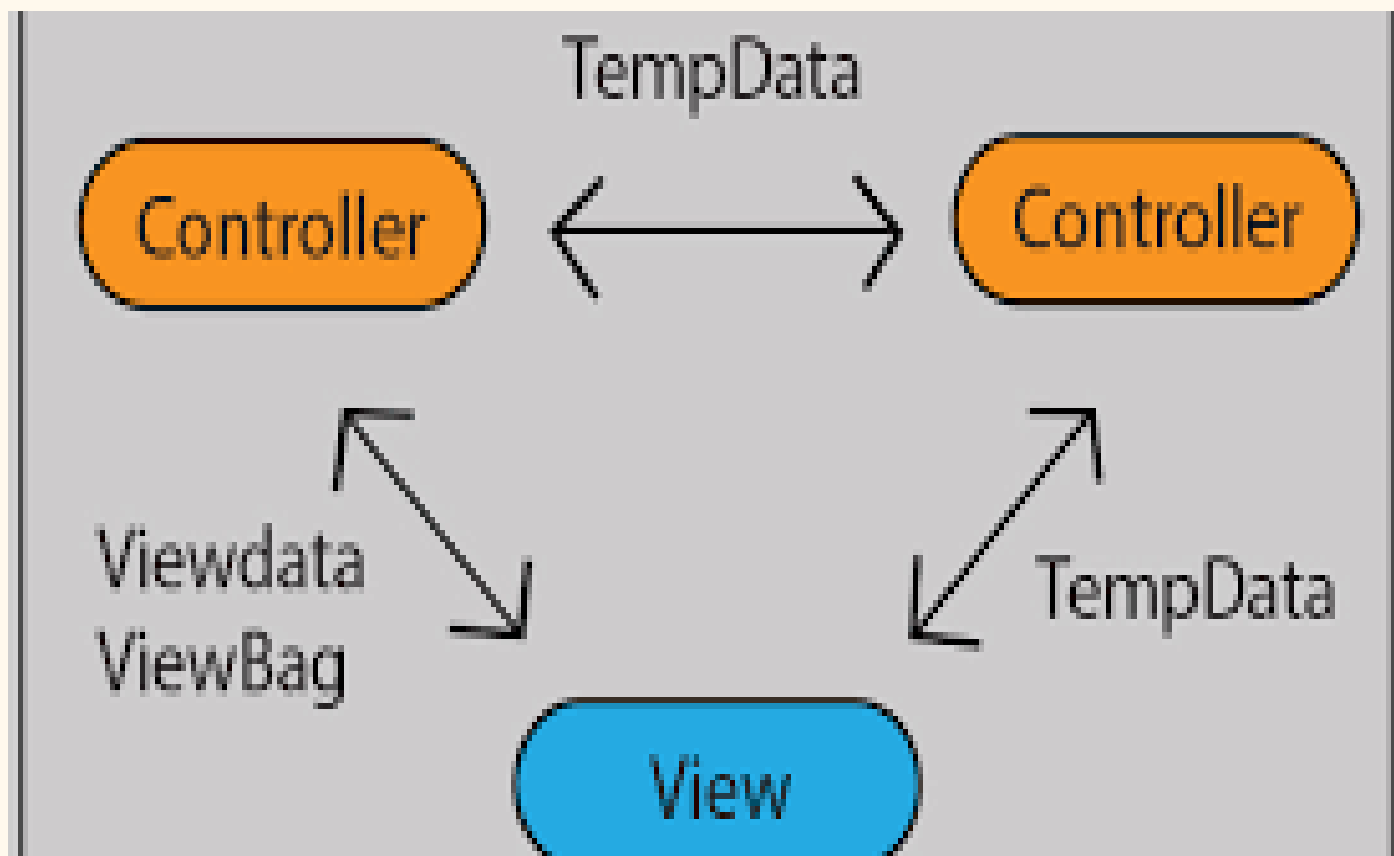


Group: Binary Beasts

Mentor: Mr. Soroush Sadr

ViewData vs ViewBag vs TempData

Date: 1402/10/25



Viewdata چیست؟

در ابتدا یک تعریف و یک مثال برای هر کدام از موضوعات بالا مطرح می‌کنیم و بعد تفاوت های این سه را توضیح می‌دهیم.

ViewData در واقع یک Dictionary object است و از ViewDataDictionary مشتق (derived) شده است، بنابراین داده ها در دیکشنری به صورت جفت های key value ذخیره می‌کند. دیکشنری در C# یک collection است که اطلاعات را در یک جفت کلید-مقدار ذخیره

می‌کند تا بعداً به کمک اسم کلید آن‌ها را بازیابی کند. ViewData فقط می‌تواند برای ارسال (passing) داده‌ها از controller به view استفاده شود و می‌تواند مقدار را فقط تا یک single request نگهداری کند، به این معنی که مقداری که در ViewData ذخیره می‌کنید پس از اولین درخواست به محض خواندن مقدار در view تبدیل به null می‌شود. در واقع به ندرت از ViewData یا ViewBag در برنامه‌ها استفاده می‌شود حداقل تا زمانی که استفاده از آن‌ها ضروری نشده است. نکته مهم بعدی که وجود دارد IntelliSense ها هستند که به برنامه نویس کمک می‌کنند که احتمال خطا را در هنگام نوشتن برنامه کاهش دهد و مزایای بیشتر برای مثال وقتی که از یک کلاس یک شیء ایجاد می‌کنیم و یک نقطه تاییپ می‌کنیم، راهنمای IDE به کمک ما می‌آید و تمام متدها و اعضای آن را نمایش می‌دهد اما جالب است بدانید که ViewData هیچ IntelliSense را به شما نمایش نمی‌دهد پس باید به عقب برگردیم و اسم key هارو به خاطر بیاریم تا بتوانیم مقادیر رو از ViewData برگردونیم.

چند نکته که در مورد ViewData بهتر است به خاطر بسپاریم:

1. برای انواع داده‌ای Complex به Type Casting نیاز داریم.
2. باید حتماً مقادیر Null رو چک کنیم تا از دریافت خطای Null Value Exception جلوگیری کنیم.
3. بدلیل مفهوم Boxing , Unboxing از استفاده غیرضروری ViewData توصیه نمی‌شود.
4. به دلیل پشتیبانی نکردن از IntelliSense این به عهده برنامه‌نویس است که نام Key ها را به خاطر بسپارد.

```
public ViewDataDictionary ViewData { get; set; }
```

ViewData یک property از کلاس ControllerBase است. ViewData برای انتقال داده‌ها از controller به view مربوطه استفاده می‌شود. و این درخواست تنها در طول درخواست فعلی در جریان است. اگر redirection رخ دهد، مقدار آن null می‌شود. برای به دست آوردن داده‌ها نیاز به type casting و بررسی مقادیر null برای جلوگیری از خطا است.

ViewBag چیزی جز یک wrapper در اطراف ViewData نیست و می‌تواند به عنوان یک ترکیب در کنار ViewData در نظر گرفته شود. یک property پویا است که از ویژگی‌های dynamic جدید در نسخه C# 4.0 بهره می‌برد. همچنین برای ارسال داده‌ها از controller به view در یک درخواست واحد استفاده می‌شود.

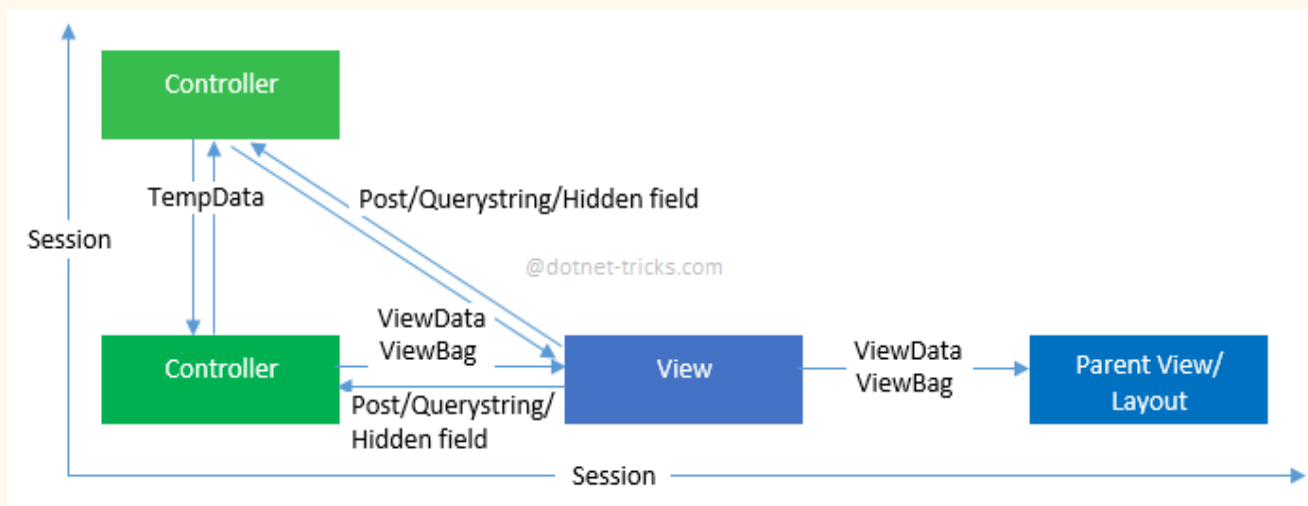
تنها تفاوت این است که ViewBag یک object از Dynamic property است و از reflection داخلی استفاده می‌کند.

نکات مهم مربوط به ViewBag که بهتر است به خاطر بسپارید:

1. برای انواع داده ای Complex به Type Casting نیاز نداریم.
 2. بررسی کردن null نیز ضروری نیست.
 3. Boxing, Unboxing را نیز انجام می‌دهد، بنابراین باید از استفاده های غیر ضروری خودداری شود.
 4. بدون IntelliSense.
- مثال: اول، استفاده از نوع Primitive و ساختن یک Action Method.

```
public Object ViewBag { get; }
```

ViewBag یک property از کلاس ControllerBase است. و عمر آن نیز فقط در طول درخواست فعلی در جریان است. اگر تغییر مسیر رخ دهد، مقدار آن صفر می‌شود. برای دریافت داده ها نیازی به type casting نیست.



TempData همچنین یک object از dictionary object است که از TempDataDictionary مشتق شده است و داده ها را در جفت های Key-value ذخیره می کند و در session های کوتاه در جریان است.

```
public TempDataDictionary TempData { get; set; }
```

TempData یک property از کلاس ControllerBase است. TempData برای انتقال داده ها از درخواست فعلی به درخواست بعدی استفاده می شود (به معنای redirecting از یک صفحه به صفحه دیگر). عمر آن بسیار کوتاه است و فقط تا زمانی که view هدف به طور کامل بارگذاری شود طول می کشد. برای به دست آوردن داده ها نیاز به type casting و بررسی مقادیر null برای جلوگیری از خطا است و فقط برای ذخیره پیام های only time مانند پیام های خطا، پیام های اعتبار سنجی استفاده می شود.

TempData را می‌توان در دو سناریو استفاده کرد:

1. انتقال داده از Controller به View.
2. انتقال داده از یک Action به Action دیگر.

Session

در ASP.NET MVC، یک Session ویژگی از کلاس Controller است که نوع آن HttpSessionStateBase است.

```
public HttpSessionStateBase Session { get; }
```

Session همچنین برای انتقال داده ها در برنامه ASP.NET MVC استفاده می‌شود و برخلاف TempData، تا زمان انقضا باقی می‌ماند (به طور پیش فرض زمان انقضای جلسه 20 دقیقه است اما می‌توان آن را افزایش داد). جلسه برای همه درخواست ها معتبر است، نه برای یک redirection. همچنین برای دریافت داده ها نیاز به type casting و بررسی مقادیر null برای جلوگیری از خطا است.

خلاصه تفاوت‌ها:

همچنین، برخلاف ViewBag و ViewData در واقع TempData می‌تواند مقدار را برای چندین درخواست متوالی نگهداری کند.

در زیر نکات مهمی وجود دارد که باید هنگام استفاده از TempData در نظر داشته باشیم.

1. برای انواع داده ای Complex به Type Casting نیاز داریم.
2. بررسی null بودن الزامی است.
3. Boxing, Unboxing را نیز انجام می‌دهد، بنابراین باید از استفاده های غیر ضروری خودداری شود.
4. بدون IntelliSense.