*1. Cung cấp một đối tượng view model*  
Chúng ta có thể một đối tượng từ action đến view bằng cách truyền nó như là một tham số đến phương thức View như đoạn mã sau:

|  |  |
| --- | --- |
| 1  2  3  4 | public ViewResult Index() {      DateTime date = DateTime.Now;      return View(date);  } |

Chúng ta truyền đối tượng DateTime như là view model. Chúng ta có thể truy xuất vào đối tượng này bằng cách dùng từ khóa Model của Razor như đoạn mã sau

|  |  |
| --- | --- |
| 1  2  3  4  5 | @{      ViewBag.Title = "Index";  }  <h2>Index</h2>  The day is: @(((DateTime)Model).DayOfWeek) |

Cách mà chúng ta làm theo ở trên được gọi là untyped hay weakly type view. View không biết bất cứ điều gì về đối tượng view model và cư xử với nó như là một thể hiển của đối tượng object. Để lấy được giá trị của thuộc tính DayOfWeek, chúng ta cần chuyển đổi đối tượng thành thể hiện của lớp DateTime. Cách này thì chạy, nhưng nó là làm cho code trong view của chúng ta thành một đóng bùi nhùi. Chúng ta có thể làm cho view của chúng ta sáng sủa, sạch sẽ hơn bằng cách dùng loại strongly typed view như đoạn mã sau:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | @model DateTime  @{      ViewBag.Title = "Index";  }  <h2>Index</h2>  The day is: @Model.DayOfWeek |

Chúng ta chỉ định kiểu của model view bằng cách sử dụng từ khóa model của Razor. Chúng ta sử dụng chữ m in thường khi chỉ định kiểu cho model và chữ M in hoa khi chúng ta đọc giá trị. Điều này không chỉ giúp chúng ta dọn dẹp sạch sẽ view của chúng ta mà còn hỗ trợ IntelliSense cho strongly typed view