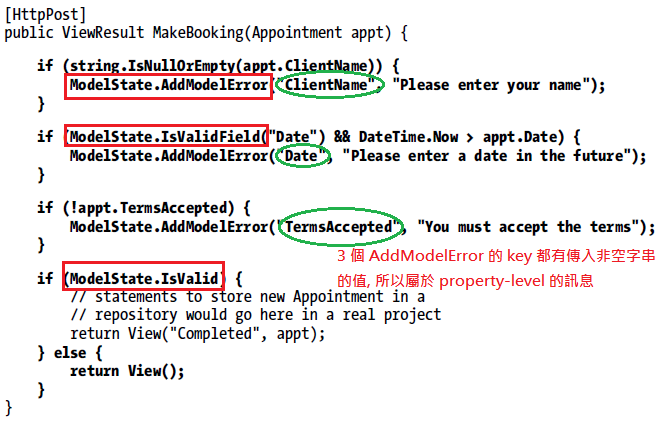
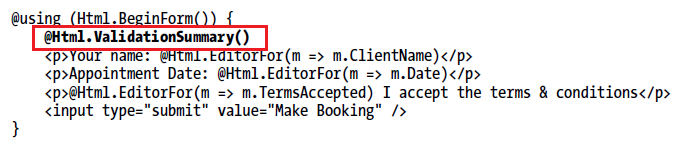
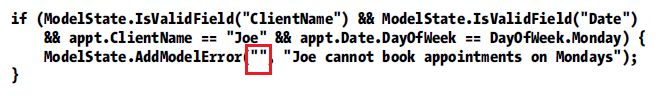
1. **Explicitly Validating a Model**
   1. Displaying Validation Errors to the User
      1. 基本上都是透過ModelState.IsValid搭配ModelState.AddModelError來使用，如下圖



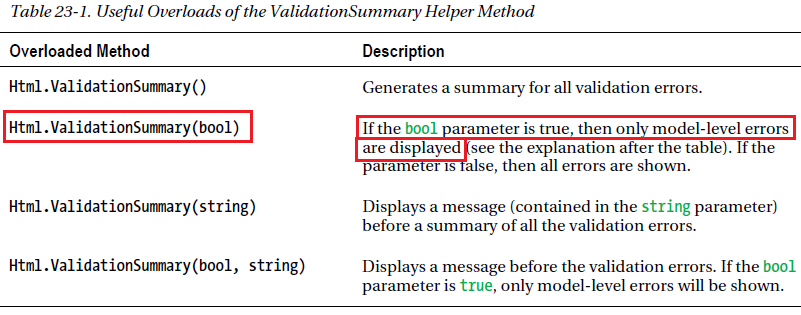
* + 1. View則是搭配Html Helper提供的ValidationSummary方法，用來顯示錯誤訊息給User



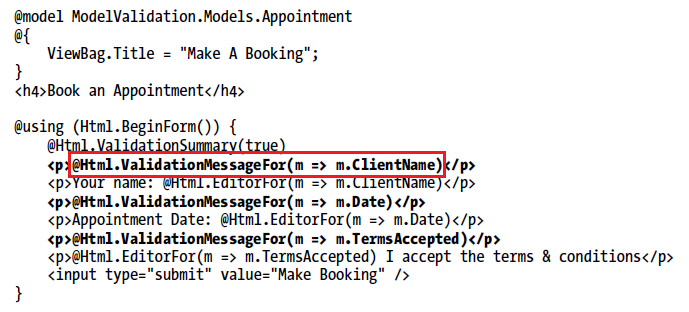
* + 1. 錯誤訊息分為Model-Level以及Property-Level，第一個圖的訊息屬於Property-Level，因為在呼叫AddModelError傳入非空字串的值(ClientName,Date這種的)，而且這本來就是針對物件的某個屬性值去做判斷。若要提供Model-Level的訊息，傳入空字串即可，如下圖



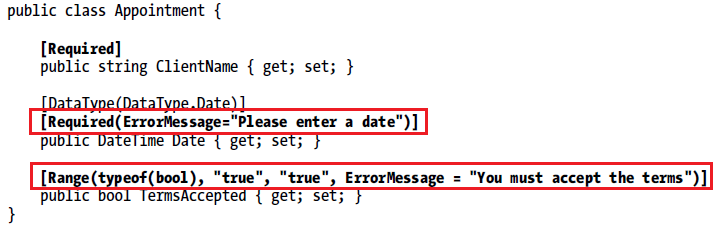
* + 1. ValidationSummary預設是顯示所有種類的訊息，其他Overload的方法如下



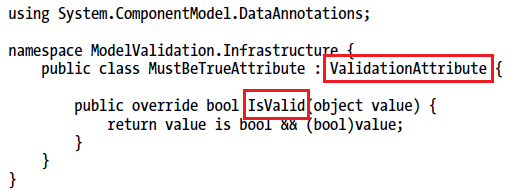
* + 1. 要顯示Property-Level的訊息也可以用以下寫法，Html Helper的ValidationMessageFor

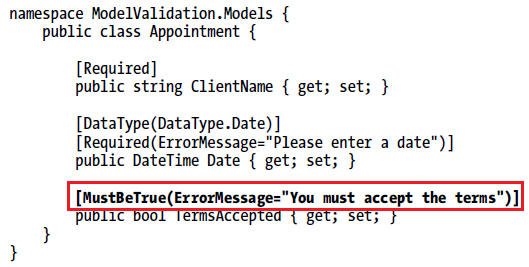


1. **Using Alternative Validation Techniques**
   1. Specifyring Validation Rules Using Metadata (可以在Model透過Attribute的方式加入一些限制)

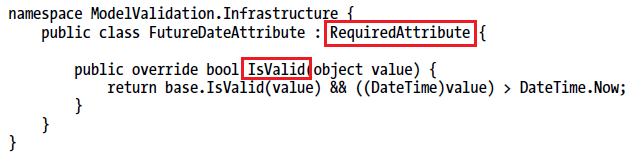


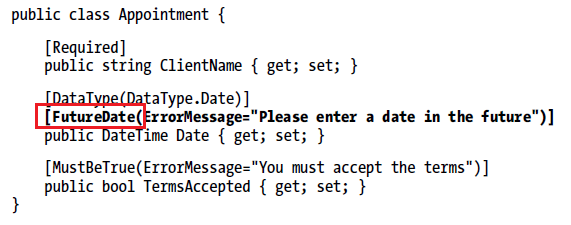
* 1. Creating a Custom Property Validation Attribute
     1. 主要繼承ValidationAttribute，實作IsValid方法



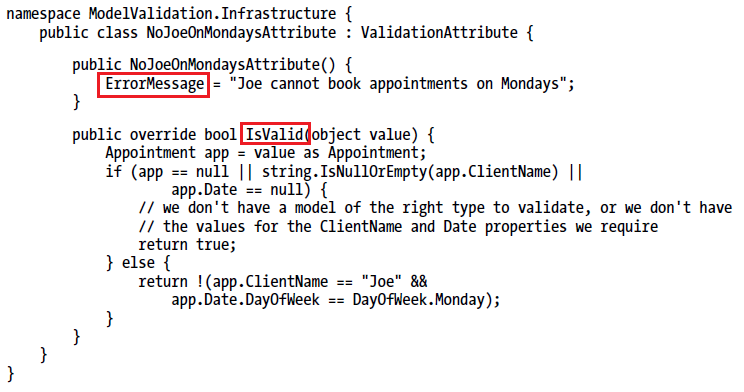


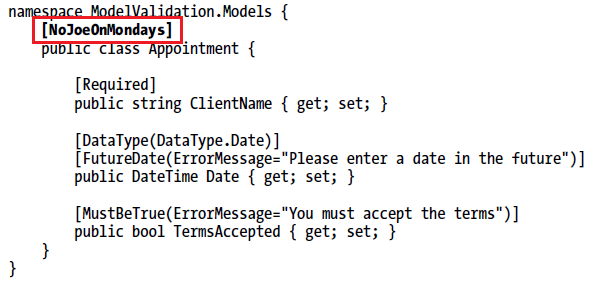
或是直接繼承內建的類別後再作修改，如下圖



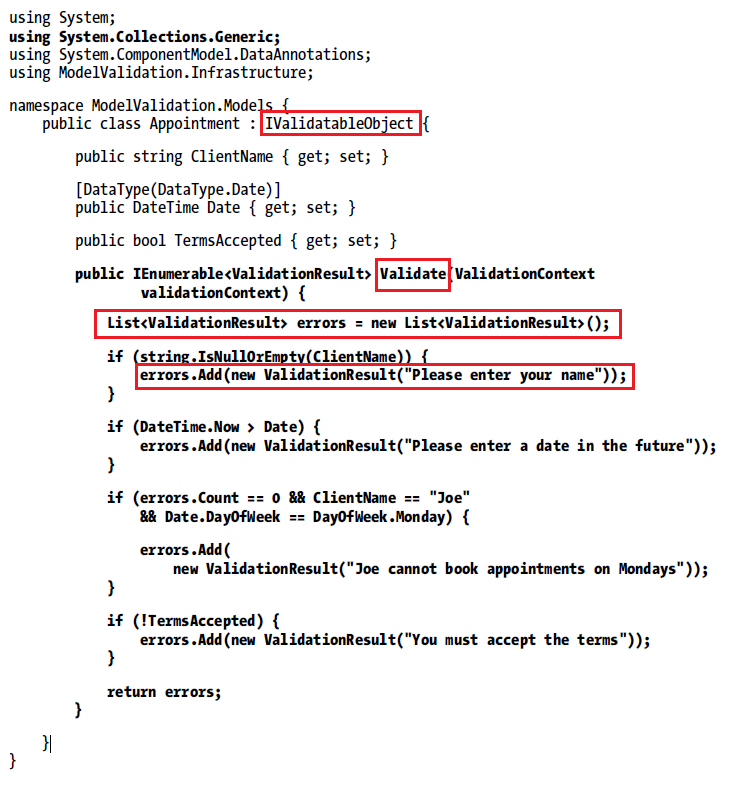


* 1. Creatring a Model Validation Attribute
     1. 一樣是建立ValidationAttribute的物件，只不過是套用在Model上面，如下圖

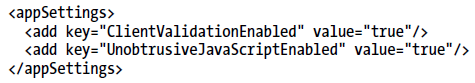




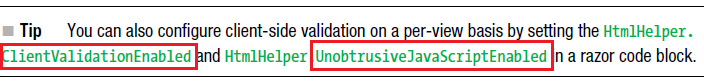
* + 1. Defining Self-Validating Models
       - 直接實作介面IValidatableObject，把相關的資料檢驗邏輯寫在Validate方法



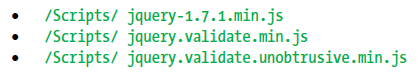
1. **Performing Client-Side Validation**
   1. Enabling and Disabling Client-Side Validation
      1. 透過Web.config裡面的設定值 (必須都為true)



也可以針對單一個View改設定



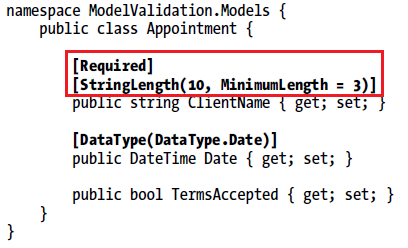
* + 1. View裡面要輸出以下三個script



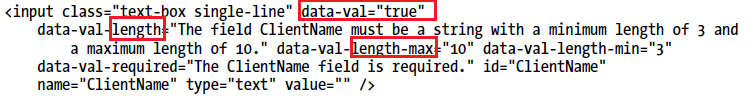
或是透過Html Helper的方法

4.PNG

* 1. Using Client-Side Validation
     1. 直接用一般的Attribute去設定Model就可以

`

* 1. Understanding How Client-Side Validation Works
     1. 跟驗證有關的資料都是以data-val做為開頭，如下圖



1. **Performing Remote Validation (呼叫Server上面的方法，在ASP.NET時代就有了)**
   1. 建立用來檢驗資料的方法，然後在Model進行綁定

