ASP.NET程式設計
ASP.NET狀態管理

講師：林賢達

Peter.lin@imestech.com

課程大綱

- ASP.NET狀態管理簡介
- 瀏覽器端的狀態管理
- 伺服器端的狀態管理
- 跨頁提交

ASP.NET狀態管理簡介

- 每次對Web伺服器的請求都是無狀態的，對Web伺服器而言，每一次請求都是新的請求
- ASP.NET狀態管理確保每一次請求之間的狀態不會遺失
- ASP.NET狀態可分成瀏覽器端與伺服器端的狀態

瀏覽器端的狀態管理

- 在瀏覽器端儲存狀態資訊，不會佔用伺服器端的資源
- 儲存位置
  - 檢視狀態(ViewState)
  - 控制項狀態(ControlState)
  - 隱藏欄位(HiddenField)
  - Cookie
  - 查詢字串(QueryString)

檢視狀態

- 將同一個網頁多次請求之間的狀態，執行hash加密後儲存在一個或多個隱藏欄位中
- 所有ASP.NET伺服器控制項都具有ViewState屬性
- 當網頁回傳到Web伺服器時，在初始化階段，便會解析隱藏欄位的字串，還原網頁和控制項屬性的值
- ViewState物件一個Key-Value Pair結構

```
ViewState("Key")=Value
Dim MyValue As Object = ViewState("Key")
```

檢視狀態

- 使用EnableViewState屬性是否啟用檢視狀態
- 檢視狀態支援的儲存類型，包括String, Integer, Boolean, Color, Array, ArrayList, Unit和上述類型的HashTable物件
- 使用Page.MaxPageStateFieldLenght屬性設定檢視狀態的最大長度(Byte)，將傳送到瀏覽器端的檢視狀態分成多個隱藏欄位

檢視狀態

- 重要特性
  - 檢視狀態是ASP.NET專有的狀態管理技術
  - 適合儲存小量的資料
  - 支援複雜的資料類型
  - 提供較高的安全性要求
  - ViewState可被關閉

控制項狀態

- 使用ControlState屬性在隱藏欄位中儲存控制項的屬性資訊
- 首先必須在初始化控制項時，呼叫RegisterRequiresControlState方法，然後覆寫SaveControlState和LoadControlState方法來儲存狀態
- 重要特性
  - 可控制如何儲存控制項狀態
  - ControlState無法被關閉

隱藏欄位

- 使用HiddenField控制項儲存網頁狀態
- 使用Request物件的Form屬性取出隱藏欄位值

  Request.Form("fieldID")
- 重要特性
  - HiddenField控制項是標準的HTML欄位，大部分的瀏覽器和移動設備均支援隱藏欄位
  - 資料未經加密，可被檢視和遭竄改的潛在風險
  - 只能儲存字串資料

Cookie

- 將資料儲存在瀏覽器端的文字檔，網站本身只認得自己的Cookie
- Cookie物件是簡單的key-value pair結構
- 當請求某個網頁時，瀏覽器會將 Cookie隨著請求傳送到伺服器端，可使用Request物件讀取Cookie資料

  Request.Cookies("key").Value
- Cookie會隨網頁輸出到瀏覽器端，可使用Response物件寫入Cookie資料

  Response.Cookies("key").Value

Cookie

- 清除Cookie

  Response.Cookies("key").Value =Nothing

  Response.Cookies("key").Expires = Now

- 重要屬性
  - Value：Cookie儲存的字串值
  - Doman：指定哪一個網域可以存取Cookie
  - Path：指定哪一個路徑的網頁可以存取Cookie
  - Expires：指定Cookie的過期時間

Cookie

- 重要特性
  - 適用於儲存小量的字串資料
  - Cookie可以是暫時(具有過期時間)或者永久保存
  - 大部分瀏覽器的Cookie最大長度是4096 Byte
  - 使用瀏覽器設定可以關閉Cookie功能

查詢字串

- 將資料儲存在URL後面，例如

  http://www.site.com/default.aspx?category=book
- 使用查詢字串，可以將資料從一個網頁傳送到另一個網頁
- 使用Request物件的QueryString或Params屬性取出查詢字串的參數值

  Request.QueryString("category")

  或者

  Request.Params("category")

查詢字串

- 重要特性
  - 大部分瀏覽器的查詢字串最大長度是2083 Byte
  - 資料顯示在網址，可直接檢視和被竄改的風險

使用時機

- 均僅適用於儲存小量的資料

	資料類型	安全性	時間	相同網頁	不同網頁
檢視狀態	Object	高	短	是	否
控制項狀態	String	低	短	是	否
隱藏欄位	String	低	短	是	是
Cookie	String	低	長	是	是
查詢字串	String	低	短	是	是

伺服器端的狀態管理

- 在伺服器端儲存狀態資訊，可減少資料的傳輸量，安全性也較高，但會增加伺服器的工作負荷
- 實現方式
  - Application狀態
  - Session狀態
  - 個人設定檔(Profile)
  - 後端資料庫

Application狀態

- Application狀態提供應用程式級別的儲存機制，應用程式的所有網頁均可存取
- ASP.NET使用Application物件來管理應用程式狀態，它是HttpApplicationState類別的實體，當應用程式啟動時，便會自動建立；停止IIS服務便被摧毀

```
Application.Lock()
Application("CustomerNum") = value
Application.UnLock()
Dim MyValue As Object = Application("CustomerNum")
```

Application狀態

- Application狀態是儲存在伺服器端的記憶體
- 常用於儲存可被多個使用者共用而且不會經常變動的資料
- 若是單純的字串資料，可考慮使用組態檔的應用程式設定(appSettings)替代Application物件儲存狀態

Session狀態

- Session狀態提供使用者連線級別的儲存機制，同一個使用者連線均可存取
- 當使用者第一次瀏覽網站時，伺服器會給予此連線唯一的SessionID，來判斷網頁請求是否屬於同一個使用者
- ASP.NET使用Session物件來管理Session狀態。它是一個HttpSessionState類別的實體。當建立一個新的使用者連線時，便會自動產生；當離開網站、逾時或者呼叫Abandon 方法，便被摧毀
- 每個使用者有自己的Session物件

```
Session("Key") = Value
Dim MyValue As Object = Session("Key")
```

Session狀態

- 使用時機
  - 購物車，使用者在網站上的購物清單
  - 使用者訊息，使用者姓名、喜好和瀏覽紀錄
  - 個人化設定
- 儲存位置
  - Cookie，瀏覽器端的文字檔
  - Web伺服器的記憶體

```
<sessionState mode="InProc" cookieless="false"
timeout="20" />
```

Session狀態

- 儲存位置(續)
  - 狀態伺服器，使用netstat –a或者工作管理員檢查是否啟動ASP.NET狀態服務，然後設定sessionState如下

    <sessionState mode="StateServer"
    stateConnectionString="tcpip=localhost:42424" />
  - 資料庫伺服器，適合Web Farm部署環境，提高系統的延展性，首先安裝aspstate資料庫

    aspnet_regsql.exe –S local –U sa –ssadd –sstype p

    然後設定sessionState組態

    <sessionState mode= "SQLServer"
    sqlConnectionString="Data Source=(local);uid=sa" />

Session狀態

- 重要成員
  - TimeOut屬性，設定使用者連線在幾分鐘內沒有任何請求，則自動清除Session狀態。預設時間為20分鐘
  - Abandon方法：中斷連線，清除Session狀態
  - Clear方法：清除Session狀態，但不中斷連線

個人設定檔

- 允許使用者定義自己的設定，並在中斷使用者連線後，自動儲存設定。當使用者下次再瀏覽網站時，直接套用原來的設定
- 重要特性
  - 具有型別
  - 永續性儲存
  - 支援儲存匿名者的個人設定

個人設定檔

- 首先在組態檔中定義profile的properties屬性

```
<profile>
  <properties>
    <add name="PreferredColor" allowAnonymous="true"
     type="System.Drawing.Color" SerializeAs="Binary" />
  </properties>
</profile>
```

- profile的properties屬性的預設類型是字串，也支援其他的資料類型，例如System.Int32, System.Drawing.Color, System.Collection.ArrayList …等

個人設定檔

- 使用個人設定

```
Profile.PreferredColor=System.Drawing.Green

Dim Preferred As
System.Drawing.Color=Profile.PreferredColor
```

資料庫

- 使用資料庫來儲存應用程式的狀態，具有安全性、個人化、永久保存和可靠性等優點
- 適合於儲存大量的交易資料，當應用程式或IIS重新啟動後，仍可保留原有的資料

跨頁提交

- 跨頁提交(CrossPagePostBack)，實作IButtonControl介面的控制項，例如Button控制項，透過設定PostBackUrl屬性，將來源網頁的Web表單資料提交到目標網頁
- 常用於使用多個網頁逐步來蒐集使用者的輸入資料
- 取出來源網頁的控制項狀態
  - 在目標網頁中，使用Page.PreviousPage屬性參考到來源網頁實體
  - 然後使用來源網頁實體的FindControl方法取出某個控制項的狀態

跨頁提交

- 取出來源網頁的控制項狀態(續)
  - 如果要取出容器控制項內的控制項狀態，首先必須先取出容器控制項實體，然後從該容器取出控制項狀態
  - 在來源網頁定義公開成員，然後在目標網頁的aspx檔案加入PreviousPageType指令，並設定VirtualPath參數為來源網頁網址，接著便可在aspx.vb檔案取出來源網頁的強類型參考，存取定義在來源網頁的公開成員