

Application、Session物件與Cookies 的處理

資科系
林偉川

Web應用程式的基礎

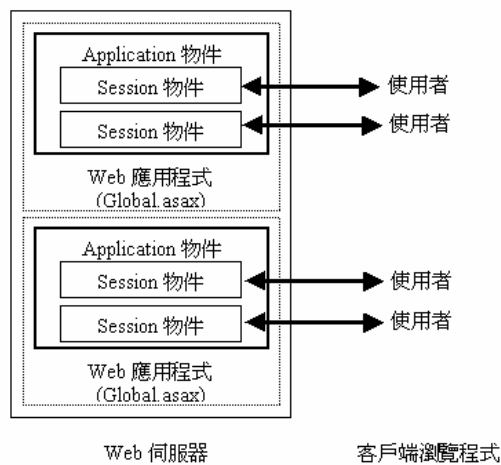
- Web應用程式是指網站目錄和其子目錄執行檔案的能力，IIS支援ASP.NET、ASP、ISAPI、CGI、IDC與SSI應用程式，能夠設定應用程式的執行權限，也就是檔案所在目錄的權限，以Web應用程式來說是【指令】和【執行】權限。

ASP.NET應用程式

- ASP.NET技術的Web應用程式就是建立ASP.NET應用程式，屬於一種.NET Framework應用程式。
- 這是一種完全在伺服器端執行的應用程式，ASP.NET應用程式是由一些程式檔案組成，當我們在Web伺服器的虛擬目錄建立ASP.NET程式檔案和設定檔案Global.asax，就是在建立ASP.NET應用程式。

3

Web應用程式的基礎



4

IIS建立Web應用程式

- Windows 2000/XP Professional的IIS能夠在同一個網站建立多個Web應用程式，在IIS建立Web應用程式的方法，因目錄種類而有不同的建立方法，如下所示：
 - **網站主目錄**：在IIS新增網站時，主目錄預設是Web應用程式。
 - **網站的虛擬目錄**：只需在新增Web共用的虛擬目錄時，設定【指令碼】權限，即可以在目錄建立Web應用程式。
 - **主目錄下實際路徑的子目錄**：預設與上一層主目錄屬於同一個Web應用程式。

5

網頁間的資料分享

- ASP.NET使用HTTP通訊協定，其特性就是ASP.NET程式並不是在客戶端的電腦執行，而是在伺服器端執行，客戶端主要的目的只是讀取網頁內容或提供使用者輸入資料的介面。
- 在Web伺服器的伺服器端，事實上，對於同時連線網站的使用者幾乎無法獲得任何使用者狀態，因為網站各網頁間並沒有什麼關連，ASP.NET程式碼的變數只限在同一個ASP.NET程式存取，而無法橫越其它的ASP.NET程式。

6

在網頁間分享資料

- 網站的分享資料可以分為2種，如下所示：
 - 分享給網站所有的使用者：在ASP.NET程式就是所謂的Application變數，例如：目前線上的使用者數。
 - 針對每位使用者的專屬資料：在ASP.NET程式就是Session變數，例如：登入使用者的權限。

7

網頁間資料分享的方法

- ASP.NET程式在網頁間傳遞資料的方法有很多種，筆者整理如下所示：
 - Web表單將表單處理的ASP.NET程式指定成其它ASP.NET程式，如此就可以將欄位資料傳遞到其它ASP.NET程式。
 - 第10章Request物件配合表單或網址URL參數，將欄位資料和參數傳到其它的ASP.NET程式。
 - 第11-3和11-4節的Application和Session物件。
 - 第11-6節的Cookies直接保留在客戶端電腦，以便ASP.NET程式取得所需的資訊。

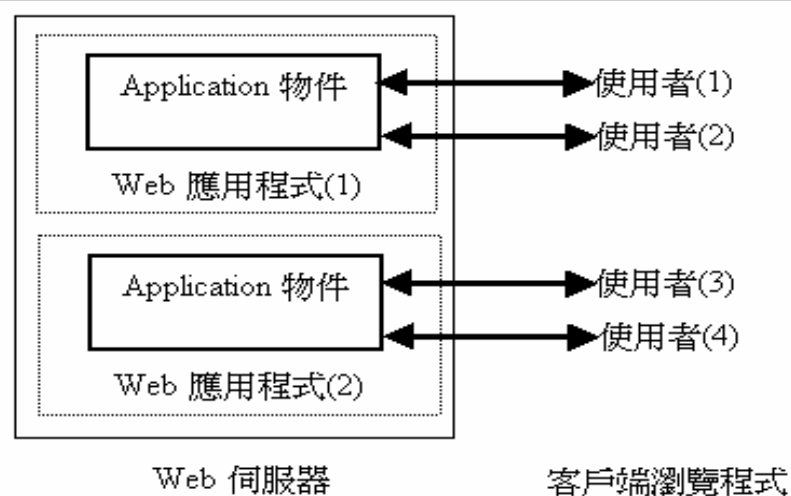
8

Application物件的基礎

- Application物件最主要的目的是建立Application變數，這和一般的程式變數不同，它是一個集合物件，Application變數提供進入Web應用程式的每位使用者一個分享資料的管道，因為Application變數的內容允許網站的每位使用者存取其值。

9

Application物件的基礎



10

Application變數的使用-基本格式

- 不論網站的Web應用程式同時有多少位使用者進入，對於每一個Application變數，伺服器都只會在記憶體保留一份資料，變數的基本格式如下：

`Application("Page_Counter")=0`

- 變數名稱和一般ASP.NET程式使用的變數不太一樣，這是一個字串"Page_Counter"，加上Application()表示為集合物件，上述程式碼將Application變數Page_Counter設為零。

11

Application變數的使用

方法	說明
Lock()	禁止其它使用者修改 Application 變數
Unlock()	允許其它使用者修改 Application 變數

12

Application變數的使用

- 如果2位使用者同時讀取Application變數並沒有關係，如果一位更改，一位是讀取Application變數，在這種情況下衝突就會發生，為了避免這種情況，Application的2個方法可以保障在同一時間內只允許一位使用者存取Application變數，如下所示：

```
Application.Lock()
```

```
Application("Page_Counter")= _
```

```
Application("Page_Counter")+1
```

```
Application.Unlock()
```

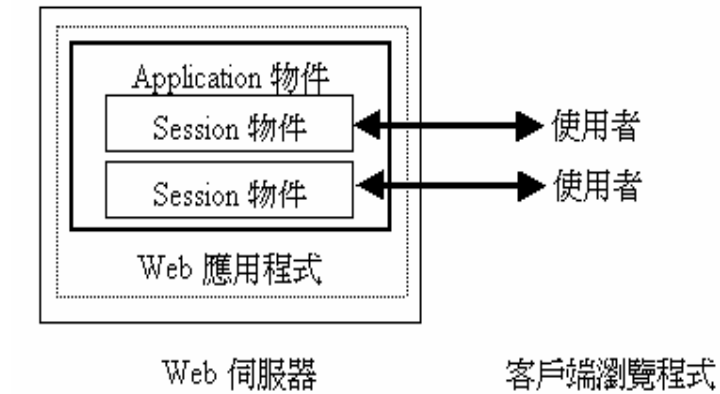
13

Session物件的基礎

- 當一位尚未建立Session物件的使用者執行Web應用程式的ASP.NET程式，伺服器就會自動建立一個新的Session物件，並且指定唯一的Session ID編號。
- 如果同時有多位使用者進入網站，每位使用者都會指定一個專屬的Session ID編號，以便建立不同的Session物件。

14

Session物件的基礎



15

Session變數的使用

- Session變數是附屬在進入Web應用程式使用者的Session期間，每一位使用者都擁有一組專屬的Session變數，雖然每位使用者的Session變數名稱相同，但是內容可能完全不同，而且只有該使用者執行的ASP.NET程式才能夠存取自己專屬的Session變數。

16

Session變數的使用

方法	說明
Abandon()	使用者建立的 Session 變數都會被清除掉，也就是說再也不能存取 Session 變數值
Remove()	刪除指定的 Session 變數，參數是 Session 變數的名稱字串

屬性	說明
TimeOut	設定每一個 Session 期間的持續時間，以分鐘計，如果超過時間，Session 變數將自動刪除
SessionID	取得使用者唯一的 Session 編號，這是一個唯讀屬性

17

Session變數的使用-存取

- 在ASP.NET程式設定Session變數的程式碼，如下所示：`Session("User_Pages")=Session("User_Pages")+1`
- 程式碼將Session變數User_Pages加一，Session變數並不需要考慮資料衝突的問題，因為它是附屬於使用者的Session期間，我們可以使用下列程式碼取得使用者目前的Session ID，如下所示：

使用者的Session ID : `<%=Session.SessionID%>
`

18

使用Session變數傳遞資料

- Session變數主要的用途是在網頁間分享資料，換句話說。例如：儲存使用者資訊的名稱和密碼，首先是使用者hueyan的Session變數，如下：

```
Session("UserName") = "hueyan"
```

```
Session("UserPassword") = "1234"
```

- 只要使用者沒有超過Session期間或執行Abandon()方法，在期間內執行其它ASP.NET程式都可以存取上述Session變數的值，如下所示：

```
name = Session("UserName")
```

```
password = Session("UserPassword")
```

19

Session變數的使用限制

- 在Session變數使用上的一些限制，如下：
 - 如果瀏覽程式為Internet Explorer 4.0x前的版本，需要開啟Cookies功能。
 - ASP.NET程式執行轉向指令，只限同一個Web應用程式，不能轉向到其它的Web伺服器。
 - 如果是十分龐大的Web應用程式，擁有多個子目錄，子目錄ASP.NET程式所建立的Session變數是無法在其它子目錄中存取。
 - Session變數的使用上比較浪費記憶體。

20

Global.asax檔案的使用

- Global.asax是一個選擇性檔案，在ASP.NET的Web應用程式並不一定需要此檔案，除非Web應用程式需要使用到Application和Session物件的事件，因為這些事件處理程序是位在Global.asax檔案。

21

Global.asax檔案的架構

- Global.asax檔案定義應用程式Application_Start、Application_End、Session_Start和Session_End的事件處理程序，檔案架構如下所示：

```
<script language="VB" runat="Server">  
Sub Application_Start(sender As Object, e As EventArgs)  
End Sub  
Sub Application_End(sender As Object, e As EventArgs)  
End Sub  
Sub Session_Start(sender As Object, e As EventArgs)  
End Sub  
Sub Session_End(Sender As Object, E As EventArgs)  
End Sub  
</script>
```

22

Application與Session物件的事件處理程序

事件處理程序	說明
Application_Start	當第一位使用者進入 ASP.NET 程式時，Application_Start 的事件就觸發，在觸發後就算有成千上萬位使用者進入網站都不會重新觸發，除非 Web 伺服器關機。程序內容通常是屬於系統設定的 Application 變數，例如：初始的訪客計數
Application_End	當 Web 伺服器關機，此事件就會觸發
Session_Start	每一位使用者在執行 ASP.NET 程式前，就觸發此事件，如果有 50 位使用者，就觸發 50 次事件處理程式，每個事件是獨立觸發，如同瀏覽網頁一般，並不會互相影響，程序內容通常是一些使用者專屬 Session 變數的初始設定
Session_End	使用者在預設的 20 分鐘內沒有進入其它的 ASP.NET 程式，就會觸發此事件，這個時間是由 TimeOut 屬性設定，程序內容並沒有辦法處理已經離開網站的使用者，通常屬於一些善後用途的程式碼，例如：將 Session 變數存入資料庫或文字檔案

23

Global.asax檔案的使用

- 在IIS的Web伺服器（Cassini Personal Web Server的每一個應用程式目錄可以擁有一個Global.asax檔案），Global.asax檔案能夠放在2個地方，如下所示：
 - 網站的主目錄：如果將Global.asax檔案放在此目錄（預設是Web應用程式），不論目錄本身，實際或其虛擬子目錄都可以存取檔案內建立的Application和Session變數。
 - 虛擬應用程式的主目錄：如果已經將虛擬目錄設定成Web應用程式，虛擬目錄的根目錄也允許擁有Global.asax檔案。

24

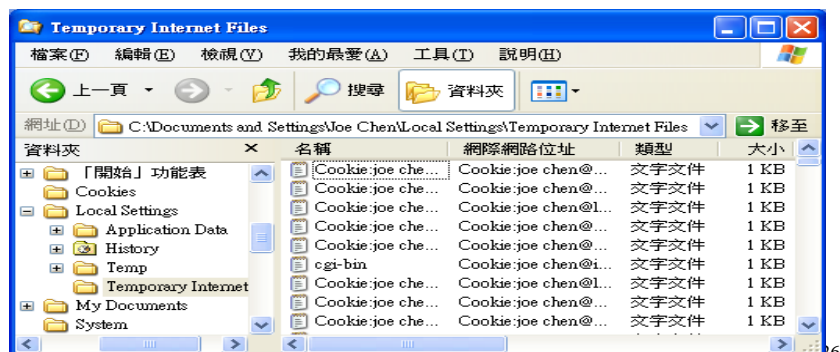
Cookies的基礎

- Cookies的英文意義是小餅乾，源於這些儲存在客戶端電腦的檔案尺寸都很小，Cookies儲存在瀏覽程式所在的電腦，所以並不會浪費伺服器的資源，只需執行ASP.NET程式進入Web應用程式時，檢查客戶端是否儲存Cookie，在取得Cookie保留的資訊後，就可以輕鬆建立複雜的Web應用程式。

25

Cookies的基礎-儲存位置

- 在「C:\Documents and Settings\Joe Chen\Local Settings\Temporary Internet Files」的快取資料夾，可以看到一大堆網站保留在客戶端電腦的Cookie檔案，如下圖所示：



26

新增Cookie

- Cookie是Response物件的Cookie集合物件，可以使用名稱存取Cookie值，新增Cookie是使用Response物件的Cookies屬性來新增Cookie，其程式碼如下所示：

```
Response.Cookies("UserName").Value = "陳會安"
```

- 在新增Cookie後還需要設定Cookie的Expires屬性，如下所示：

```
Response.Cookies("UserName").Expires = _  
    DateAdd("D", 10, Today)
```

- 程式碼使用DateAdd()函數加上10天，計算出Cookie過期天數，當日期到後Cookie就會刪除。

27

取得Cookie的值

- 如果客戶端電腦擁有Cookie，我們可以使用Request物件的Cookies集合物件取得指定Cookie的值，其程式碼如下所示：

```
name =Request.Cookies("UserName").Value
```

- 程式碼取得Cookie名稱UserName的值，將它指定給變數name。

28

刪除Cookie

- 如果客戶端的Cookie不再需要，我們可以使用程式碼刪除Cookie，刪除方式只需將Expires屬性設定成過期即可，其程式碼如下所示：

```
Response.Cookies("UserName").Expires = DateAdd("D", -365, Today)
```

- 程式碼將有效期限設定為一年前，因為Cookie已經過期，換句話說，就是刪除Cookie。

29

多鍵Cookie的使用-建立

- Cookie是一種目錄結構的集合物件，在同一個名稱Cookie下允許擁有不同的Key鍵名，換句話說，同一個Cookie名稱就可以儲存多個資料，如下：

```
Response.Cookies("User")("UserName") = "陳會安"
```

```
Response.Cookies("User")("Password") = "1234"
```

```
Response.Cookies("User").Expires = _  
DateAdd("D", 10, Today)
```

- 程式碼新增名為User的Cookie，這個Cookie擁有2個鍵名UserName和Password，分別儲存使用者名稱和密碼，最後設定Cookie期限屬性Expires時只需使用Cookie名稱。

30

多鍵Cookie的使用-取出與刪除

- 多鍵Cookie如同一個陣列，取得Cookie值時不只需要指定名稱，還需要指定鍵名，如下所示：

```
name = Request.Cookies("User")("UserName")
```

```
pass = Request.Cookies("User")("Password")
```

- 多鍵Cookie的刪除只需使用Cookie名稱設定Expires屬性就可以刪除，如果只刪除Cookie其中的一個鍵名，只需將該鍵名設成空字串，如下所示：

```
Response.Cookies("User")("Password") = ""
```

31

Cookies的基本應用

- Cookies在網站實作上的應用相當多，使用Cookies保留的資料大都屬於幾個方面，如下所示：
 - 個人資訊：使用Cookies保留個人的資訊，例如：姓名、地址、時區、帳號和造訪網站的次數。
 - 個人化的內容：Cookies可以定製個人化的網站外觀和個人偏好的網站內容，或者只提供使用者有興趣的資訊。
 - 網站購物車：線上購物車需要保留使用者選擇的商品，Cookies就可以用來記錄這些選購的商品。

32

儲存使用者的瀏覽記錄

- 如果使用者在瀏覽網站時，一進入網頁就馬上顯示歡迎再次到訪的訊息，使用者可能會驚訝於這個網站怎麼如此Smart，事實上，該網站早已經在您第一次進入網頁時，就已經在電腦留下到訪的記錄，這就是Cookies的功能。
- ASP.NET程式檢查客戶端的電腦是否已經留下使用者名稱的Cookie，其程式碼如下所示：
`name = Request.Cookies("UserName").Value`
- 如果變數name有內容，表示已經建立Cookies，就直接顯示歡迎使用者的訊息，否則顯示Web表單輸入使用者名稱，然後建立Cookie儲存使用者名稱。

33