HTTP是 \*\*Stateless(沒有狀態)\*\* 的通訊協定(Protocol)，Stateless(沒有狀態)含有以下特色:

1. 每個Request 都是唯一且獨立的

2. 不會記住之前的連線

也就是說，每次頁面載入彼此之間都沒有關係。但...這樣太沒感情了!!(不是啦!)，這樣有時候其實不太方便。

為了蝦皮雙十一積了半年的購物車，如果從購物車頁面跳轉到結帳頁面，它就像失憶了一樣，完全不記得你的購物車內容!!!!!瞬間是不是非常崩潰!!!!

那有哪些方式可以儲存資訊，讓頁面間彼此共享?本文將為大家介紹Application、Session、Cookie、Cache以及ViewState。

==============================================================

# Application

- \*\*介紹\*\*

- Application存放在server的記憶體中，當IIS或網頁伺服器重新啟動、修改Global.asax、修改 web.config時， Application的值便會遺失。

- 因為儲存在伺服器的記憶體中，因此與在資料庫中存儲和檢索資訊相比，它的執行速度更快。

- Application 的內容所有user都能夠存取，且都存取同一個，是所有user```共用```的，全域變數的概念，因此可以用來存放```少量經常使用，但不會因為使用者不同而變更的資料```。

- 存放的資料內容為Object資料型別，讀取時需要轉換為適當的型別使用。例如:Conver.ToInt32(Application["名稱"])。

- 資料量大小無限制，以server的記憶體大小而定

- 使用頻率較低，在Asp.Net中類似的資料可以儲存在Web.config檔案中

- \*\*範例\*\*:

網頁瀏覽次數，無論是誰造訪這個網頁都共同存取同一個變數並+1

- \*\*相關語法\*\*

由於大家都共用這個變數，避免同一時間有多人同時存取，所以存取Application 狀態資料前，須將Application鎖定，修改後再打開

```

Application.Lock();//先把Application固定

Application["yoyo"]="大家都看得到這段字喔!";//對Application賦值

Application.UnLock();//把Application打開

string yoyo =Convert.ToString( Application["yoyo"]);//存取時要記得轉型

```

==============================================================

# Session

- \*\*介紹\*\*

- 存放在server的記憶體中

- Session的內容鎖定同一瀏覽器，並不是所有user都共用。你打開第一次chrome使用這個網頁，跟把它關掉再用Chrome再開一次，對Session還說是不同user。或者你同時開了Chrome還有IE造訪同一個網頁，也是代表不同user。

- 預設生命週期為20分鐘(若是持續使用不閒置，可以一直存在)，或是直到瀏覽器被關閉才消失。```同一瀏覽器只是關掉分頁Session不會消失```，換頁面也不會消失，故類似跨網頁的全域變數

- 存放的資料內容為Object資料型別，讀取時需要轉換為適當的型別使用。例如:Conver.ToInt32(Session["名稱"])。

- 資料量大小無限制，以server的記憶體大小而定

-\*\*範例\*\*:

傳統的購物車網站傳遞資料的方法。可以把選好的紀錄存在session裡，當跳到結帳頁面時也不會消失。

- \*\*相關語法\*\*

```

Session["yoyo"]="Hiyoyo";//賦值Session["yoyo"]

Session.Timeout = 30;//更改Session過期時間

Session.Remove("yoyo");//移除 Session["yoyo"]的值

Session.RemoveAll();//移除所有Session變數

Session.Clear(); //移除所有Session變數

Session.Abandon();//移除 session 的所有變數，且會觸發 Session\_End 事件，把 Session["yoyo"]整個砍了

```

==============================================================

# Cookie

- \*\*介紹\*\*

- 是放在Web Server和瀏覽器之間，伴隨request和網頁傳送的一小段文字

- 存放在使用者電腦

- cookies存在時間通常為1000分鐘，也可以自行設定

- 資料量大小最大為4096 bytes，實際一個瀏覽器可以儲存的cookie數量及大小受限於瀏覽器設定

- 每個Cookie需要有唯一的名稱，若名稱相同會覆蓋過去

- String格式存放

- "每次"發請求都會夾帶 Cookie，會影響訪問網站的速度

- \*\*相關語法\*\*

```

HttpCookie yoCookie = new HttpCookie("test");//用new宣告一個型別為HttpCookie的yocookie

yoCookie.Value = Server.UrlEncode("我是內容");//通常寫入cookie的值會加密，所以加密後賦值

yoCookie.Expires = DateTime.Now.AddDays(2);//設定2天後過期

Response.Cookies.Add(yoCookie);//寫到用戶端

HttpCookie readCookie = Request.Cookies["test"];//用request讀取

Response.Write("cookie內容 : " + Server.UrlDecode(readCookie.Value));

```

也可以在單一cookie中儲存多的名稱/值組。增加設定方便性以及減少cookie的數量及大小

```

//方法一

Response.Cookies["test"]["username"] = "I'm Cynthia";

Response.Cookies["test"]["gender"] = "女";

//方法二

HttpCookie yoCookie = new HttpCookie("test");

yoCookie.Values["username"] = "I'm Cynthia";

yoCookie.Values["gender"] = "女";

```

==============================================================

# Cache

- \*\*介紹\*\*

- 儲存在伺服器記憶體

- 所有瀏覽者在同一時間共用同一個變數

- Cache自己設定時間，時間到消失，不受網頁關閉或關機影響

- 通常應用在靜態網頁快取以及入口型網站

- 目的是希望可以減少對資料庫的存取

-\*\*範例\*\*:

當有一個入口網站，內容不會頻繁變動，但是頁面上有許多資訊需要從資料庫抓取，每個人點進來就要重新抓取並載入一次，其實非常消耗流量跟資源。於是我們將輸出結果存到Cache裡，並設定6小時或合適的時間。當第一個人進到頁面時，(被迫)佛心的幫大家把結果載完並存到cache，接下來6個小時內進來這個頁面的人都不用重新載入就能看到跟第一個人一模一樣的頁面。一直到6小時時間到，下一個人再繼續扮演(被迫)佛心的好孩子。~~有客怨也只有第一個人會抱怨呵呵~~

- \*\*相關語法\*\*

```

//存放資訊

Cache["yoyo"] = "1000";

//存放資訊

Cache.Insert("yoyo", "1000");

//讀取資訊

String UserNameID = Cache["yoyo"].ToString();

```

==============================================================

# ViewState

- \*\*介紹\*\*

- ViewState 屬性提供一個字典物件，用於在對同一頁的多個請求(request)之間保留值。

- 用來存放ASP.NET頁面上控制項的狀態值給兩個連續的postback使用

- 要傳遞的變數會存放在隱藏表單欄位（hidden field）的值（Value）屬性中( \_ VIEWSTATE)

```

<input Type="Hidden"...>

```

- 不需要伺服器資源，ViewState包含在頁面程式碼的結構中

- ViewState的值會經過雜湊、壓縮，並且透過Base-64編碼，安全性較高

- 只能用在同一頁面裡，無法跨網頁

- 可存放可序列化的Object

- 不是用存放大量資料，網頁載入可能會太久

- 若要關閉ViewState，需關閉 @page 裡面設定EnableViewState=false

- \*\*相關語法\*\*

```

ViewState["Text"] = "123";//賦值

string myValue = ViewState["Text"].ToString();//取值

```

==============================================================

# 比較表

| |變數 | 變數儲存位置| 生命週期| 資料量大小

------------- | -------------| -------------| -------------| -------------

Application | 所有瀏覽者共用同一個| 伺服器記憶體| 不關伺服器就一直存在| 任意

Session | 一個瀏覽器一個| 伺服器記憶體| 關掉瀏覽器就消失(關掉分頁不會)。也可設定存活時間(預設為20分鐘)| 小量

Cookie | 一個瀏覽者一個 | 使用者電腦| 通常存在1000分鐘，也可以自己設時間| 小量

Page.Cache | 所有瀏覽者在同一時間共用同一個 | 伺服器記憶體| 自己設定時間，時間到消失，不受網頁關閉或關機影響| 任意

ViewState | 一個頁面(aspx)同一個 | 用戶端| 該頁面的生命週期| 小量

==============================================================

# 參考資料

[[ASP.NET]ViewState、Session、Cookies、Application State比較](https://dotblogs.com.tw/hatelove/2009/06/28/viewstate-session-cache-cookies-application-of-user-state)

[[.NET] ASP.NET 狀態管理（State Management）：ViewState](https://dotblogs.com.tw/marcus116/2011/05/24/25923)

[Application, Session, Cookie, Viewstate, Cache物件用法和區別](https://www.itread01.com/content/1548041225.html)

[91之ASP.NET由淺入深 不負責講座 Day3 - ViewState](https://ithelp.ithome.com.tw/articles/10051189)

[Session、Cache、Cookie、ViewState簡略說明](https://dotblogs.com.tw/jamessever/2011/06/29/30509)

[[ASP.NET] Application、Cookie、Session和Cache](https://sophieprofession.pixnet.net/blog/post/17325503-%5Basp.net%5D-application%E3%80%81cookie%E3%80%81session%E5%92%8Ccache-)

https://www.itread01.com/p/1392783.html

https://www.cnblogs.com/qiantuwuliang/archive/2009/12/27/1633110.html

https://www.csie.ntu.edu.tw/~r93057/aspnet134/ch1.pdf

https://dotblogs.com.tw/jimmyyu/2010/10/16/difference-between-stateful-and-stateless