



ASP.NET 程式設計 輸出快取

講師：林賢達

Peter.lin@imestech.com

Microsoft
CERTIFIED
Professional

Microsoft
CERTIFIED
Solution Developer

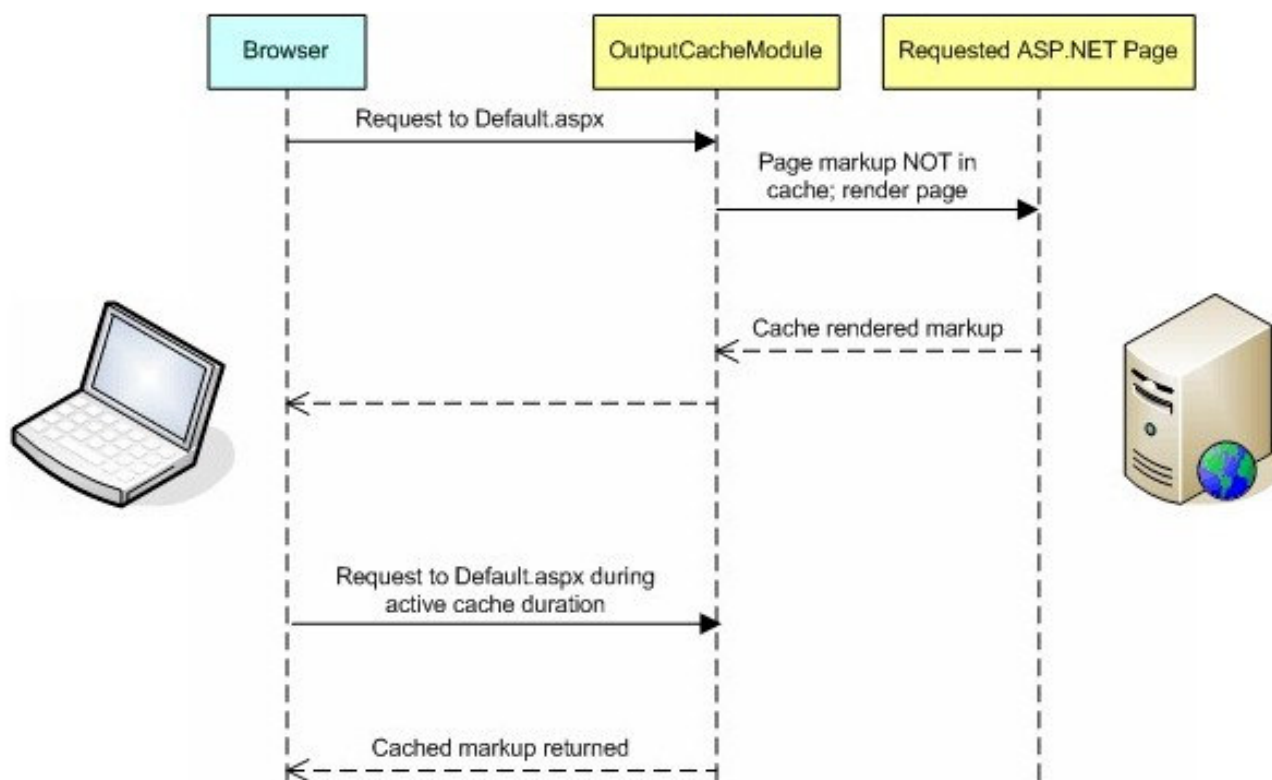
Microsoft
CERTIFIED
Application Developer

課程大綱

- ASP.NET快取簡介
- ASP.NET快取機制
- 頁面快取
- HttpCachePolicy類別和Cache物件
- 使用SqlDataSource控制項
- 使用Substitution控制項
- SQL快取

ASP.NET 快取簡介

- 當某個網頁頻繁被瀏覽時，可以快取整個網頁HTML輸出來提高執行效能



ASP.NET 快取簡介

■ ASP.NET 快取機制

- 頁面輸出快取(Page Output Caching)
- 頁面片段快取(Page Fragment Caching)，使用者 Web 控制項的輸出快取
- 頁面輸出快取 API – HttpCachePolicy 類別
- 使用 Cache 物件快取變數值

ASP.NET 快取簡介

- ASP.NET 2.0提供的快取機制
 - 使用DataSource控制項
 - 使用Substitution控制項
 - SQL快取(SQL Cache Invalidation)

頁面快取

- 在aspx檔案中使用@OutputCache指令以宣告方式設定頁面快取。當網頁在第一次被瀏覽時便會快取在記憶體中，直到過期為止
- 重要參數
 - Duration：設定快取的持續時間，以秒為單位（此為必要參數）
 - Location：設定快取的存放位置，其類型是一個OutputCacheLocation列舉值，包含Any（預設）、Client、Downstream（代理伺服器）、Server和None，不支援使用者Web控制項快取

頁面快取

■ 重要參數(續)

- Shared：使用者Web控制項的輸出是否可被多個頁面所共用
- VaryByParam：設定快取是否相依於參數，根據不同的參數值建立不同的快取版本 (此為必要參數)，參數之間用分號(;)隔開
- VaryByControl：設定快取是否相依於控制項，快取是否因某個控制項的值改變而改變
- VaryByCustom="browserlstring"，根據不同的瀏覽器或者任意的字串建立不同的快取版本

頁面快取

- 在組態檔中加入快取設定檔，設定應用程式級別的快取策略，然後在aspx檔案中使用@OutputCache指定的CacheProfile參數指定使用哪一個快取設定檔

```
<system.web>  
  <caching>  
    <outputCacheSettings>  
      <outputCacheProfiles>  
        <add name="CachePolicy1" enabled="true" duration="30"/>  
      </outputCacheProfiles>  
    </outputCacheSettings>  
  </caching>  
</system.web>
```

```
<%@ OutputCache CacheProfile = "CachePolicy1" %>
```


HttpCachePolicy類別

- 提供控制ASP.NET網頁輸出快取的方法，等同於使用@OutputCache指令
- 使用Response.Cache參考System.Web.HttpCachePolicy實體
- 重要成員
 - SetExpires方法：設定快取過期的絕對時間
Response.Cache.SetExpires(Now.AddSeconds(60))
 - SetSlidingExpiration方法：是否將快取過期的絕對時間改成相對時間

HttpCachePolicy類別

- 設定頁面快取相依於某個檔案，當檔案內容發生改變時，頁面快取便過期

```
Dim file1 As String=Server.MapPath("file1.txt")
```

```
Dim file2 As String=Server.MapPath("file2.txt")
```

```
Dim files() As String = New String{file1, file2}
```

```
Response.AddFileDependencies(files)
```

```
Cache.Insert ("Item1", 123, New  
CacheDependency(Server.MapPath("file1.txt")))
```

Cache類別

- ASP.NET使用Cache物件來快取變數值，其生命週期與應用程式相同
- 使用Cache關鍵字參考到System.Web.Caching.Cache實體

'賦值

```
Cache("Var1") = "abc" 或者 Cache.Insert ("Var1", "abc")
```

'取值

```
Dim MyString As String
```

```
If Not Cache("Var1") Is Nothing Then
```

```
    MyString = CType(Cache("Var1"), String )
```

```
End If
```

SqlDataSource控制項

- 使用DataSource控制項快取資料來源
- 快取的相關屬性
 - EnableCaching：是否啟用快取
 - CacheDuration：指定快取過期期限(以秒為單位)
 - CacheExpirationPolicy：指定快取的期限策略是絕對(absolute)或者相對(Sliding)
 - CacheKeyDependency：快取相依的索引鍵，當鍵值改變時，快取便過期
 - SqlCacheDependency：快取相依的資料表，當資料表的資料改變時，快取便過期

Substitution控制項

- 使用Substitution控制項，在快取頁面中嵌入動態內容

- 定義一個用於傳回動態內容的**回呼靜態方法**，傳入參數類型必須是HttpContext，且傳回字串

```
Shared Function GetRealTime(context As HttpContext) As String  
    Return DateTime.Now.ToString()
```

```
End Sub
```

- 將MethodName屬性設定成回呼靜態方法名稱，
MethodName="GetRealTime"

或者使用

```
Response.WriteSubstitution(New  
    HttpResponseSubstitutionCallback(AddressOf GetRealtime))
```

SQL快取

- 當SQL Server的資料發生變動時，自動通知Web應用程式，立刻更新快取，而不需要等到快取過期
- 僅支援SQL Server 7.0以上的版本
- 開發步驟
 - 設定SQL Server支援SQL Cache Invalidation
 - 設定web.config
 - 使用SQL Cache Invalidation

SQL快取

■ 啟用SQL快取相依性的資料庫

```
aspnet_regsql -S [SQL Server執行實體] -E -d [資料庫名稱] -ed
```

■ 啟用SQL快取相依性的資料表

```
aspnet_regsql -S [SQL Server執行實體] -E -d [資料庫名稱]  
-t [資料表名稱] -et
```

■ 在相依性的資料庫新增下列物件

- AspNet_SqlCacheTablesForChangeNotification 資料表
- AspNet_SqlCacheNotification_Trigger 觸發程序
- AspNet_SqlCachePollingStoredProcedure... 等五個預存程序

SQL快取

- 在組態檔中使用sqlCacheDependency元素設定SQL快取的輪詢間隔時間

```
<キャッシング>  
  <sqlCacheDependency enabled="true" pollTime="1000" >  
    <databases>  
      <add name="testdb"  
        connectionStringName="testdbConnectionString" />  
    </databases>  
  </sqlCacheDependency>  
</キャッシング>
```

- 在aspx頁面中加入

資料表名稱的大小寫有別

```
<%@ OutputCache Duration="600" sqldependency="testdb:employee" %>
```


清除快取

- 當發生以下狀況，系統會自動清除快取
 - 當伺服器記憶體不足時，開始一個清除快取的過程，首先會清除一段時間不再使用或優先等級較低的快取項目
 - 當快取過期時
 - 當相依物件發生變化時