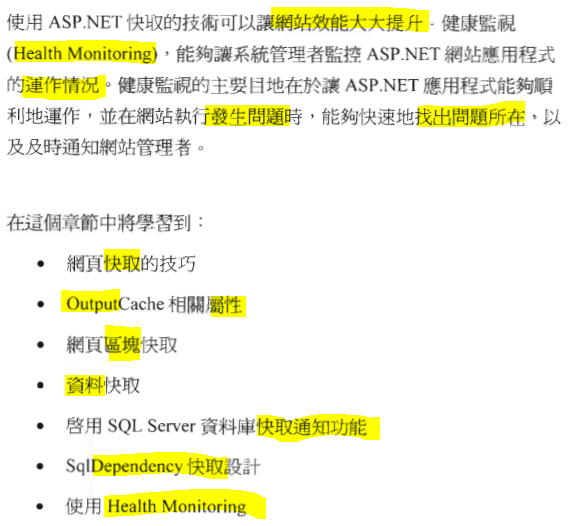
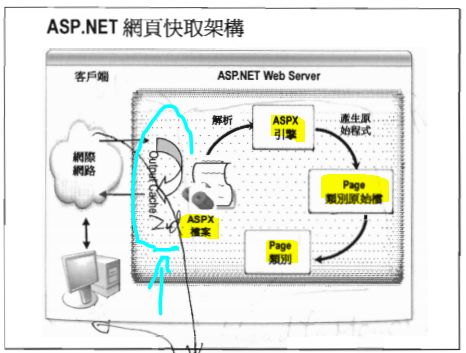
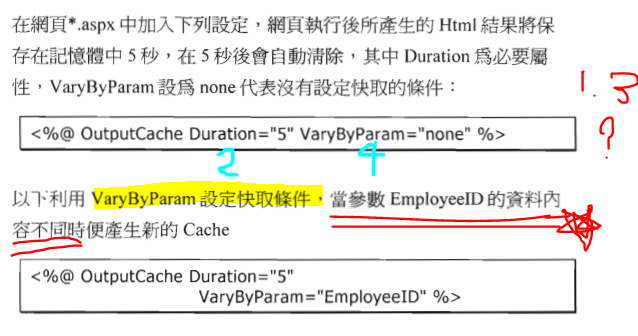
P 365

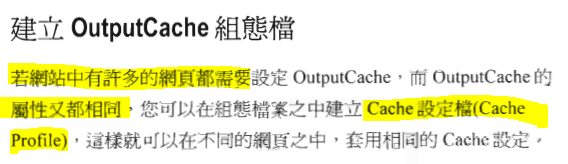


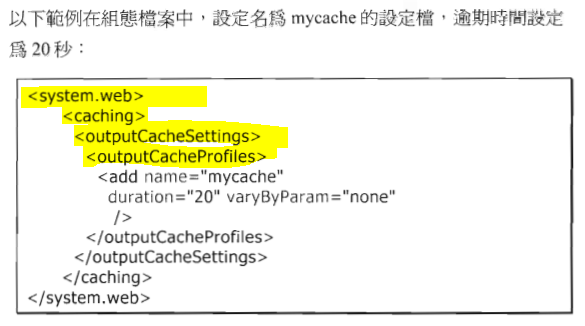
P 368

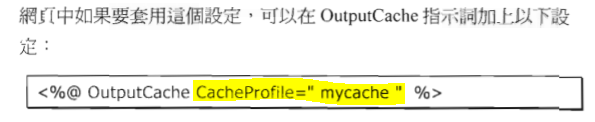




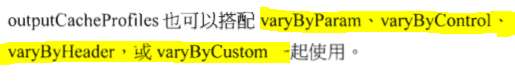
P370

 P 371

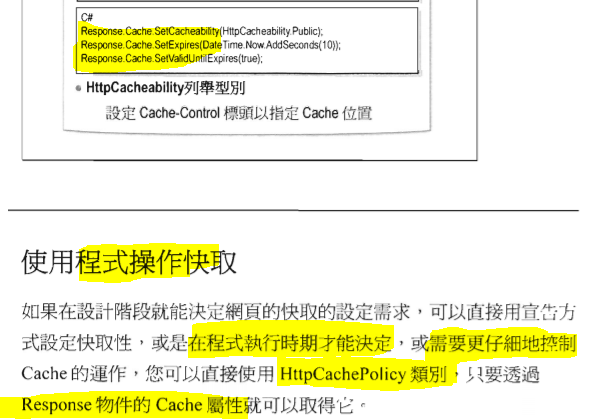


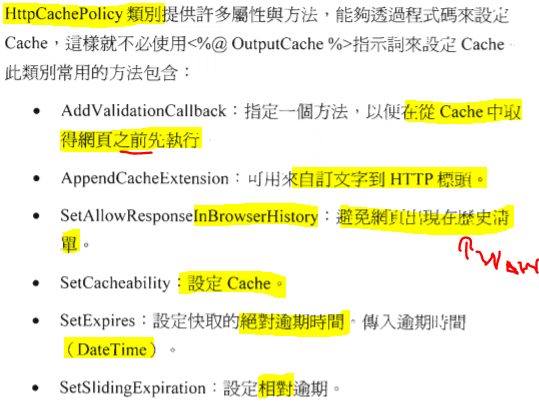


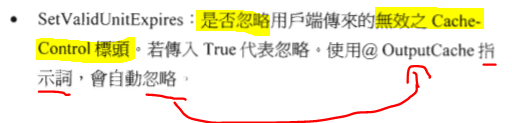
4 好像有很多選擇，但是 3 沒有出現過？



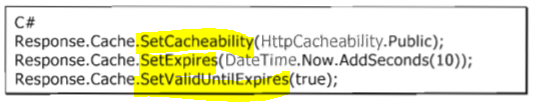
--------------------------------------------------方法二P 373----------------------------------------



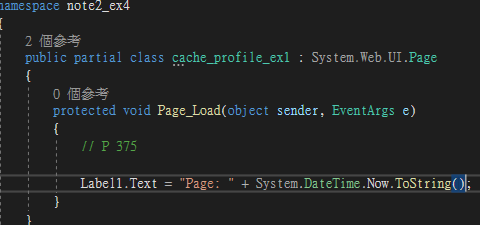


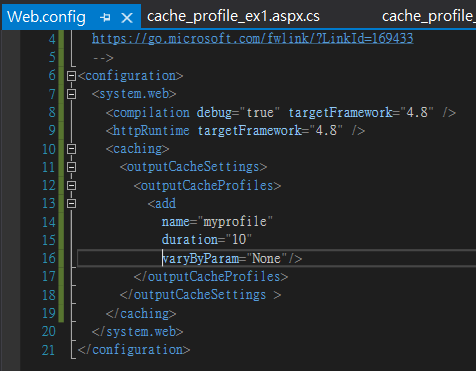






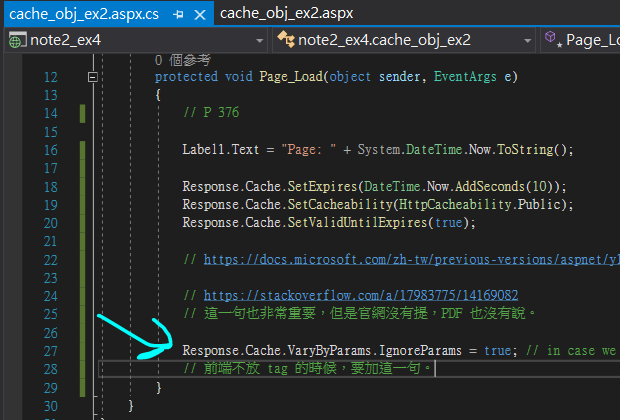
P 375 練習 note2\_ex4 <- 方案 + 專案 | cache\_profile\_ex1.aspx



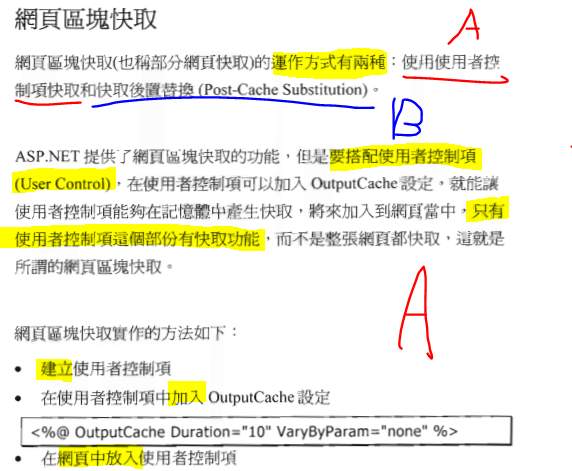


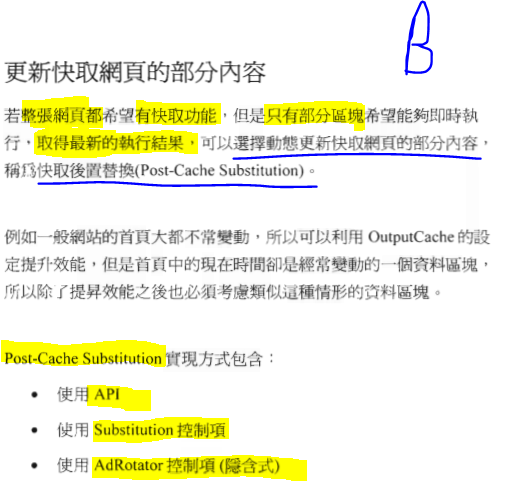


cache\_obj\_ex2.aspx

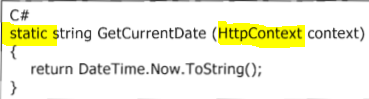


P 378 局部Cache

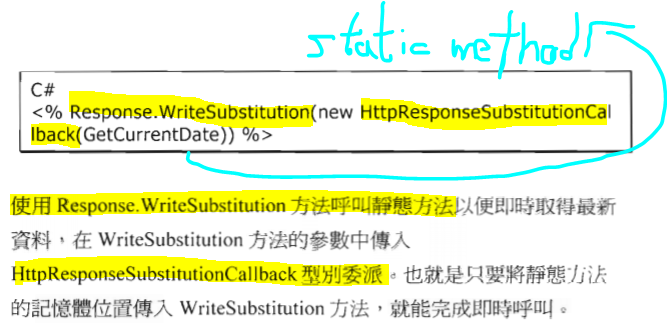


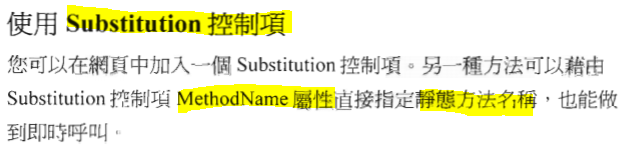


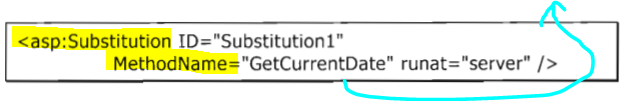




利用 API : P 381

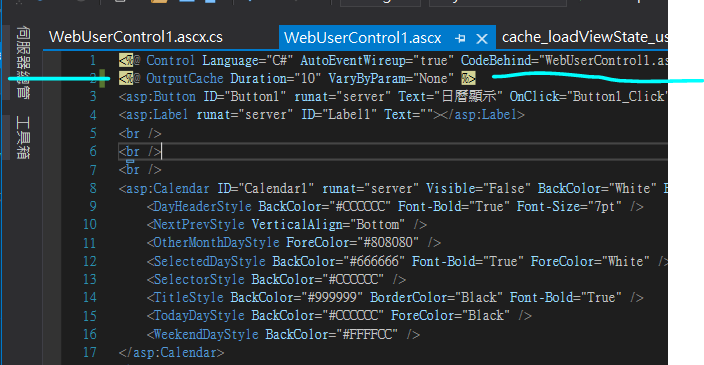






PDF 沒有實作， 要自己來

A : partly\_cache\_loadViewState\_userCtrl.aspx (引用之前的練習再改一下)

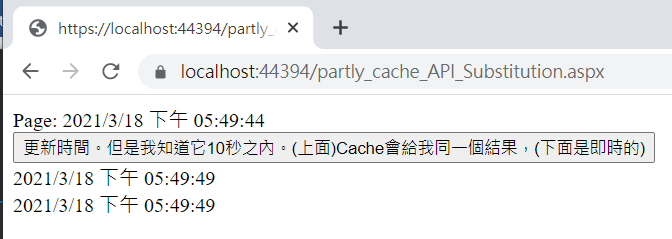


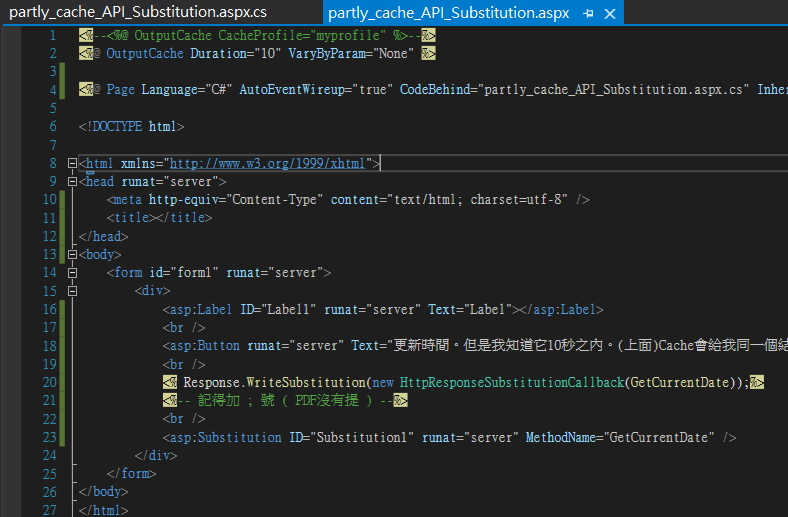


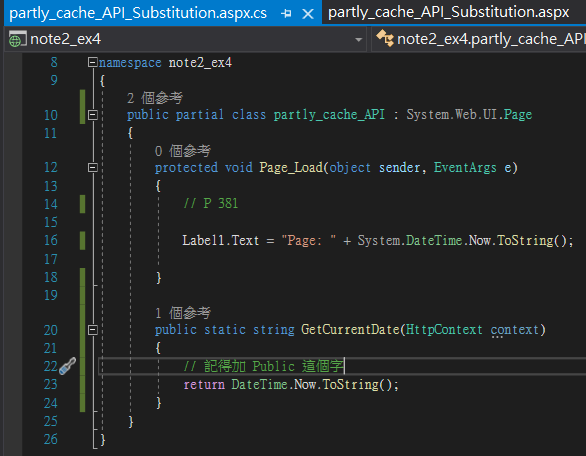
除了( 上上圖 )有指出來的那一句，

其他Code和 note2\_ex3 的 load\_ViewState\_userCtrl.aspx 沒有差

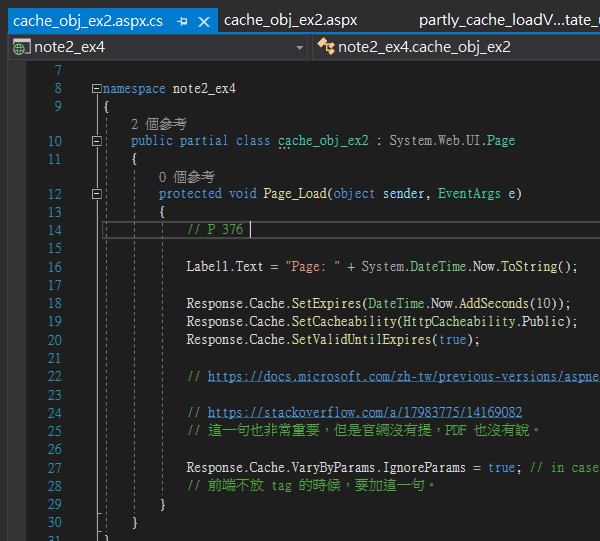
B1,2 :



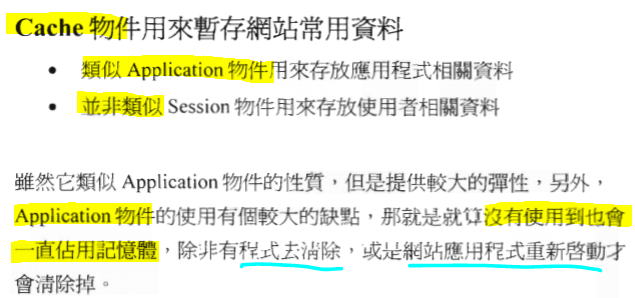


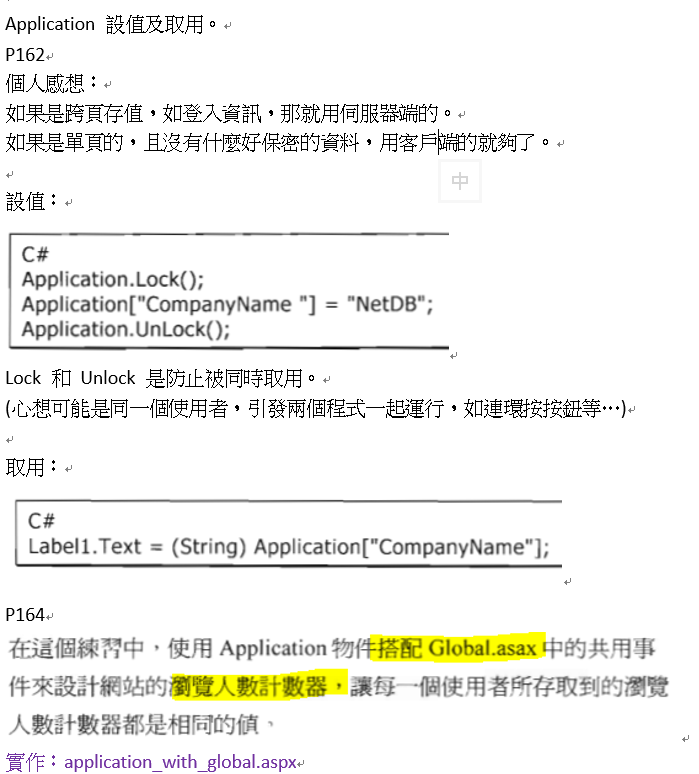


P 383 注意，它並不是這個「頁面」的Cache，而是這個「網站」的Cache



<-它並不是在說這個

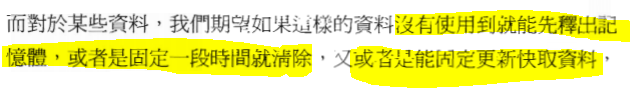


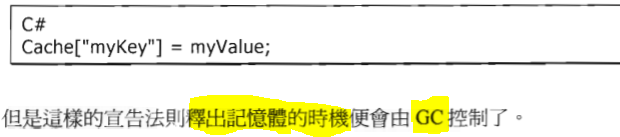


回顧 Application ︰

(四)狀態維護(網頁靜態變數).docx

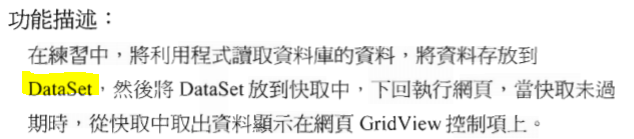
Cache 的特點︰

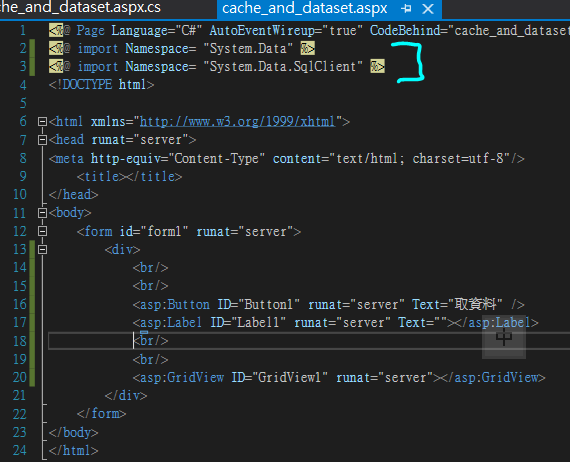
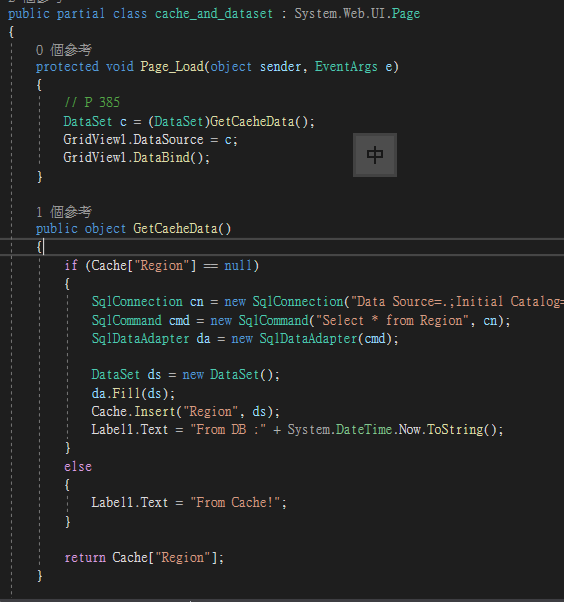


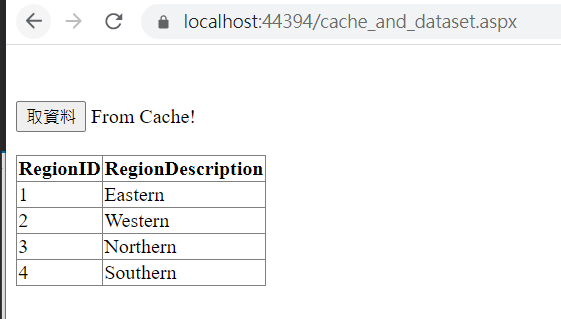
Garbage Collection

在練習之後，這一些時間設定等有再細說。

P 385 練習 cache\_and\_dataset.aspx







但是因為沒有設下期限，

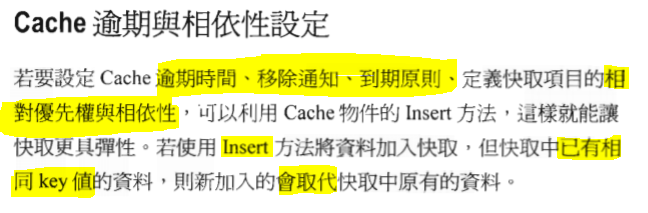
所以它即使是回伺服器去取的。

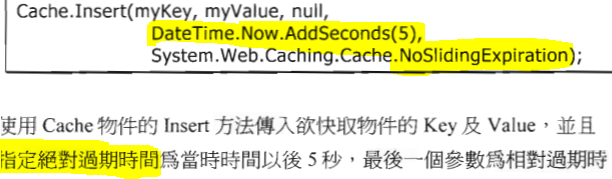
它永遠只會回傳第一次生成網頁的搜尋結果。

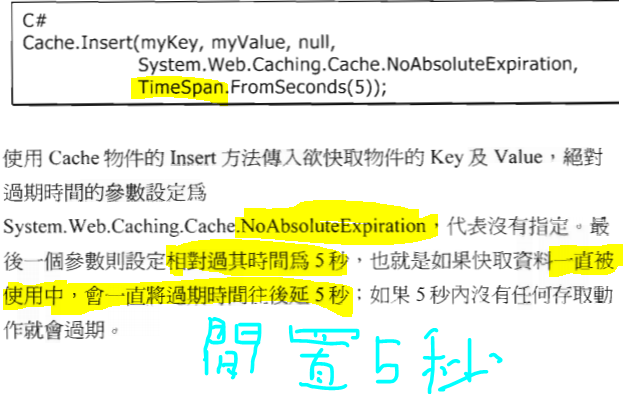
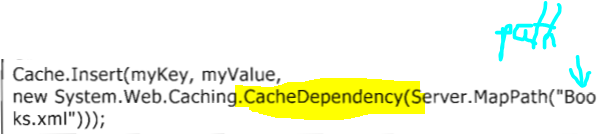
好處是︰相比於Application，它在這一頁關閉之後，就會清理掉了。

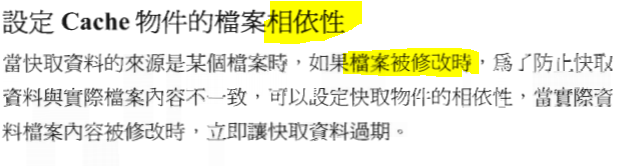
(Application則會一直累加，多個用戶都會拿到同一個結果。就算用不到都不會消失。)

P 388



絕對

閒置

 修改內容

?? 但是沒有練習？ 那我只好又自己來了。修改做過的cache\_and\_dataset.aspx

* cache\_and\_dataset\_abs\_time.aspx

Cache.Insert("Region", ds, null, DateTime.Now.AddSeconds(5), System.Web.Caching.Cache.NoSlidingExpiration);

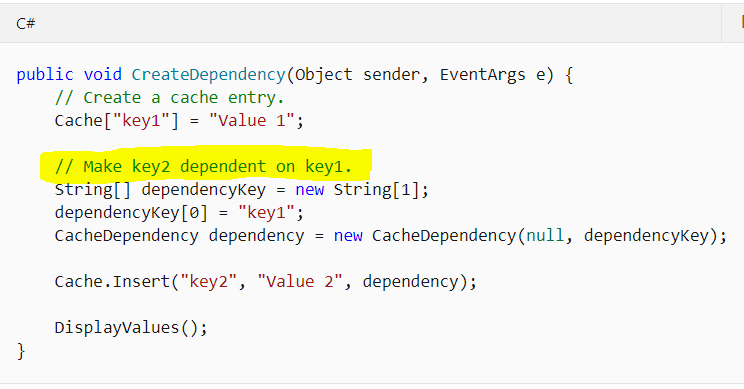
* cache\_and\_dataset\_still\_time.aspx

Cache.Insert("Region", ds, null, System.Web.Caching.Cache.NoAbsoluteExpiration, TimeSpan.FromSeconds(5));

Cachedependency的補充︰

<https://docs.microsoft.com/zh-tw/dotnet/api/system.web.caching.cachedependency?view=netframework-4.8>

<https://docs.microsoft.com/zh-tw/dotnet/api/system.web.caching.cachedependency.-ctor?view=netframework-4.8>



這個滿有趣的，也不用多建一個檔案和寫入檔案的方法，所以我實作這一個。

public partial class cache\_and\_dataset\_do : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

if (!Page.IsPostBack)

{

Cache["key1"] = DateTime.Now.ToString("G");

Label2.Text = Cache["key1"].ToString();

}

DataSet c = (DataSet)GetCaeheData();

GridView1.DataSource = c;

GridView1.DataBind();

}

public object GetCaeheData()

{

if (Cache["Region"] == null)

{

SqlConnection cn = new SqlConnection("Data Source=.;Initial Catalog=Northwind;Integrated Security=True");

SqlCommand cmd = new SqlCommand("Select \* from Region", cn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds);

string[] dependencyKey = new string[1];

dependencyKey[0] = "key1";

CacheDependency dependency = new CacheDependency(null, dependencyKey);

Cache.Insert("Region", ds, dependency);

Label1.Text = "From DB :" + System.DateTime.Now.ToString();

}

else

{

Label1.Text = "From Cache!";

}

return Cache["Region"];

}

/// <summary>

/// Key 1

/// </summary>

protected void Button2\_Click(object sender, EventArgs e)

{

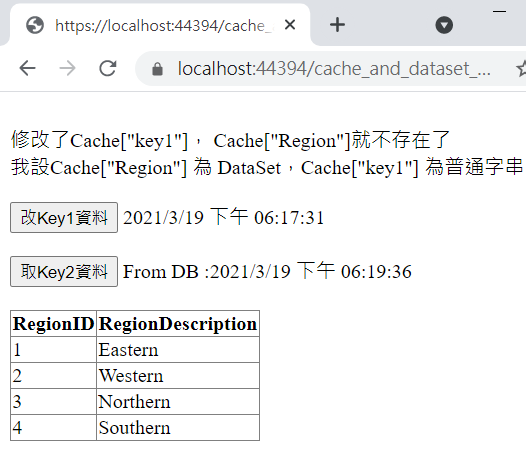
Cache["key1"] = DateTime.Now.ToString("G");

Label2.Text = Cache["key1"].ToString();

Label1.Text = Cache["Region"] == null ? "Cache['Region']為 NULL " : Cache["Region"].ToString();

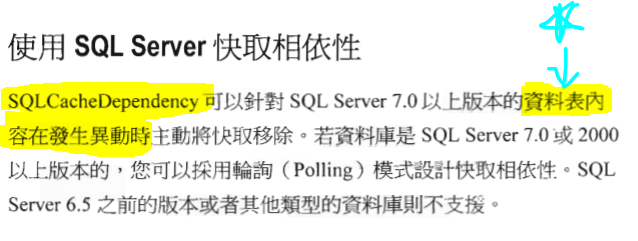
}

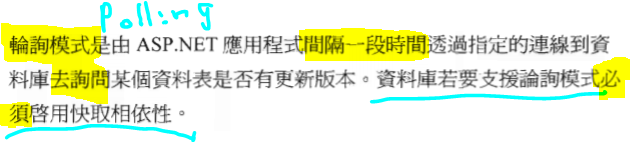
}

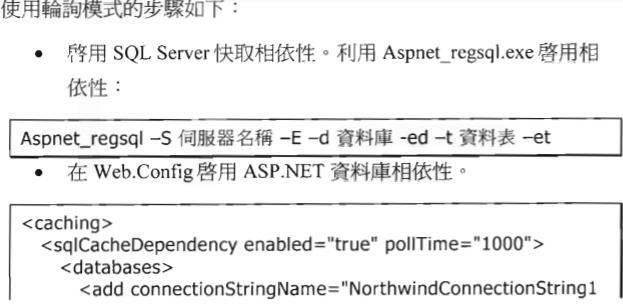




P390

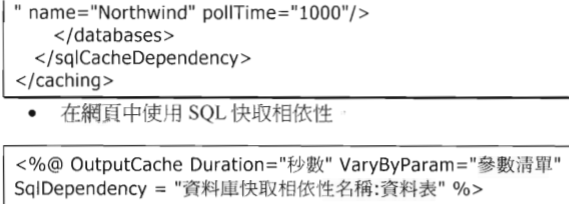




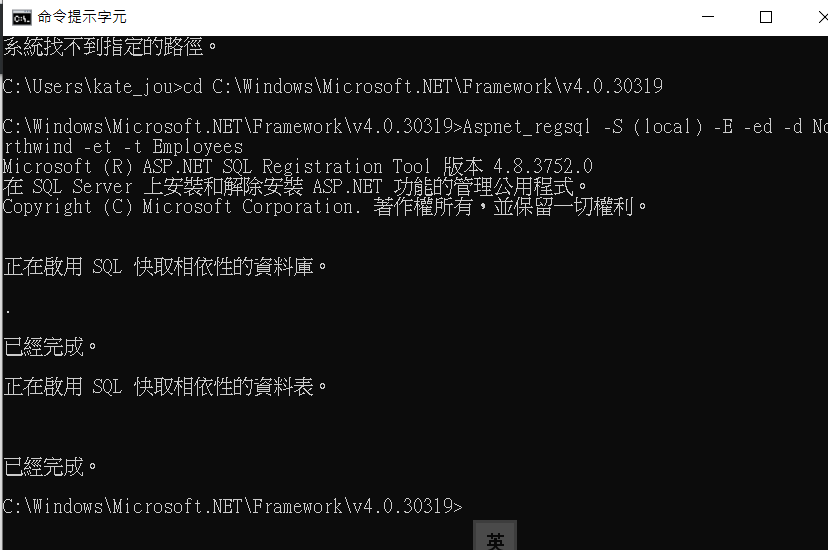


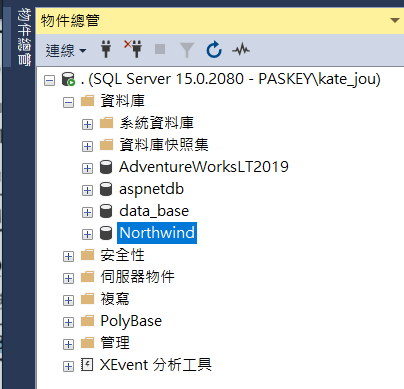
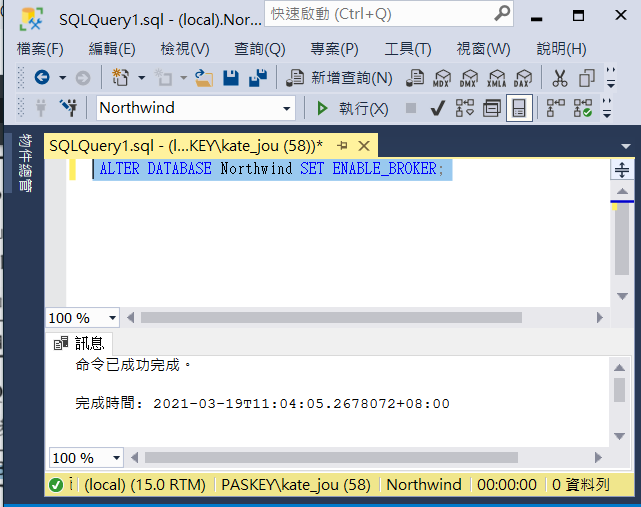
Cmd

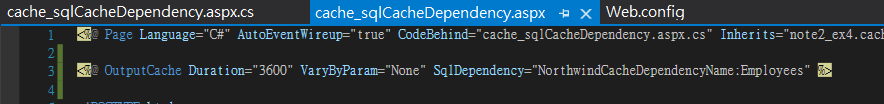
cd c:/xx/ run ….

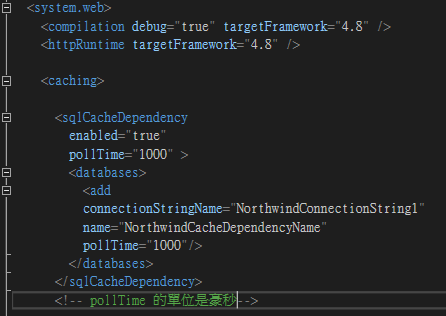


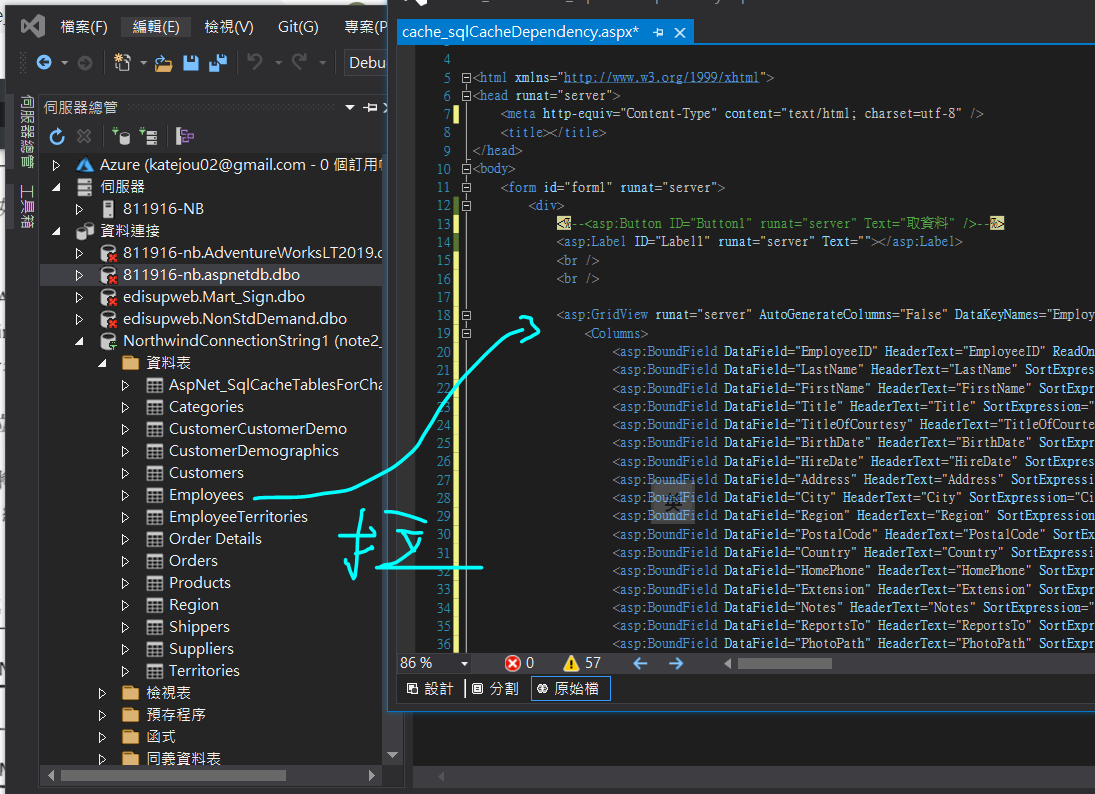
P 392 練習



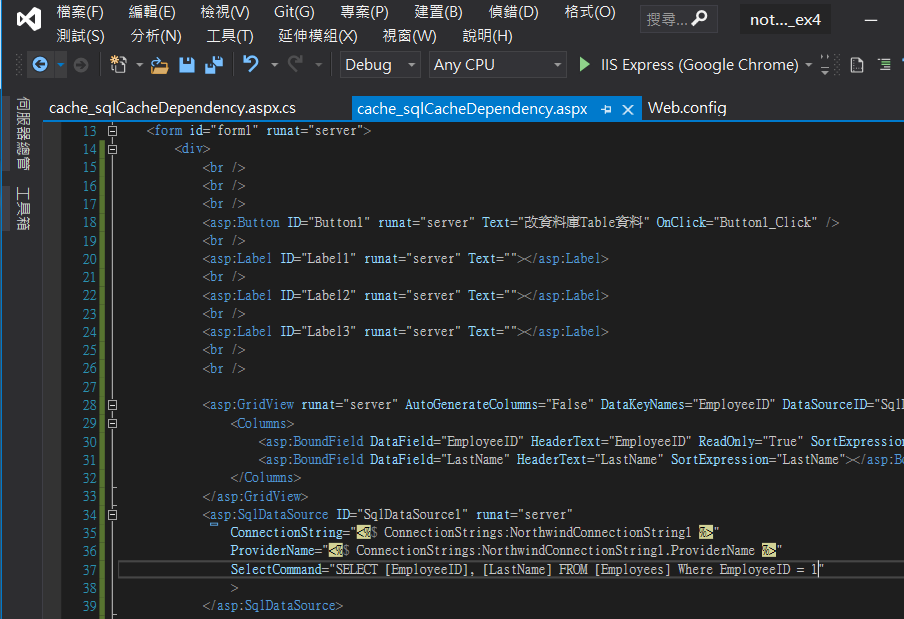


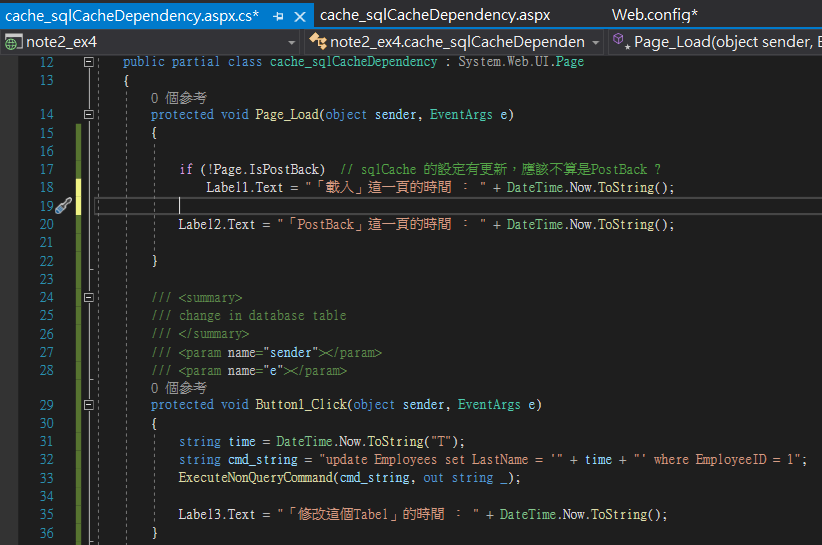




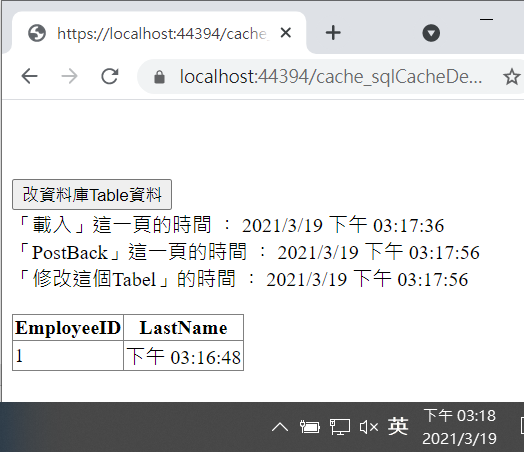


自動產生 GridView 和 sqlDataSource

刪減後



結果︰…..

這一頁的Cache

是設成了3600秒。

Polling 是 1 秒。

我按下「改資料庫」的時候，

這一頁的Cache就失效了。

所以3600秒Cache 對這個 button是沒有用的

但是︰

前端繫結的 Gridview 沒有任何變化

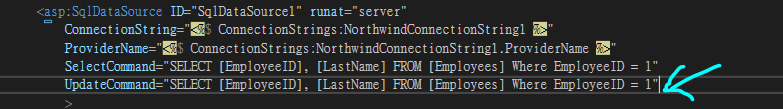
我本來預期它會自動polling 出新結果

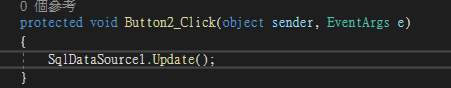
但是它維持在「初次載入」的狀態。

所以我猜3600秒 Cache 對他有效。

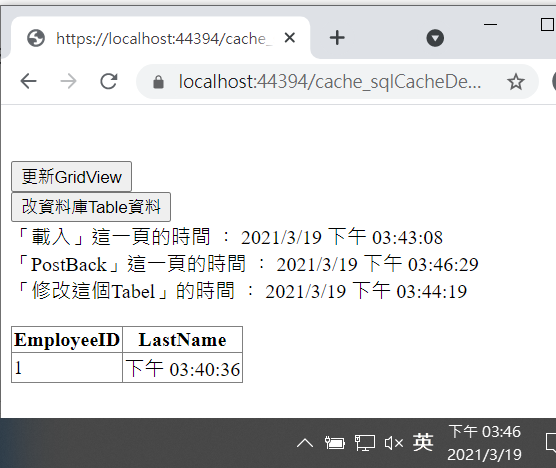
我減少 3600秒 的Cache 到 10 秒，加一個可以更新DataBinding 的按鈕給GridView。

看看會怎麼樣。





結果︰Update 的指令是不對的。它並沒有重新載入GridView資料。



.Update()的真正指令作用是︰

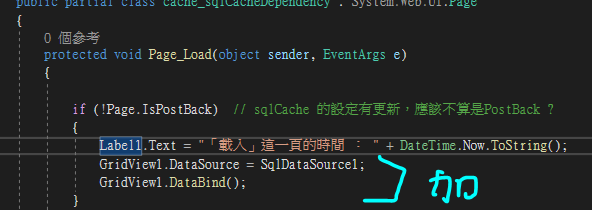
<https://docs.microsoft.com/zh-tw/dotnet/api/system.web.ui.webcontrols.sqldatasource.update?view=netframework-4.8>

正確的做法是︰

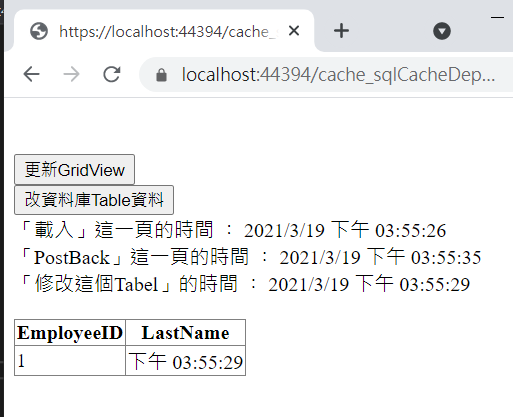
GridView1.DataSource = SqlDataSource1;

GridView1.DataBind();

所以也不能用 DataSourceID 的屬性來進行第一次載入時的綁定︰



**成功了！**



GridView的結果

如果不動資料庫，

它10秒可更新一次。

但如果資料庫有動，

它就會一直可以更新。

但是我發現，Cache好像有Disable Button 的功能。

如果我先按了「更新GV」，「改資料庫」的Button會被封鎖10秒。

結論︰

不要把它們兩個按鈕放在同一頁，使用者會吐血的。

A 頁

只放「更新GV」在查詢頁

= > 每 X 秒檢查是否有新資料

有 => cache失效，要新資料

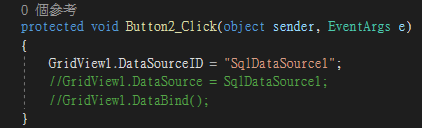
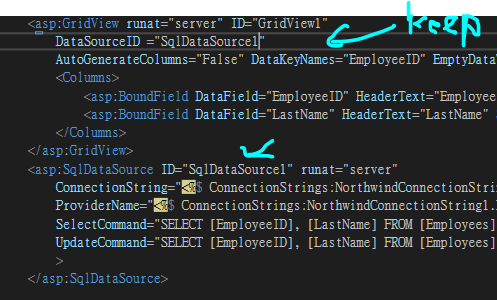
沒有 => 回傳這一頁保留的Cache

B 頁

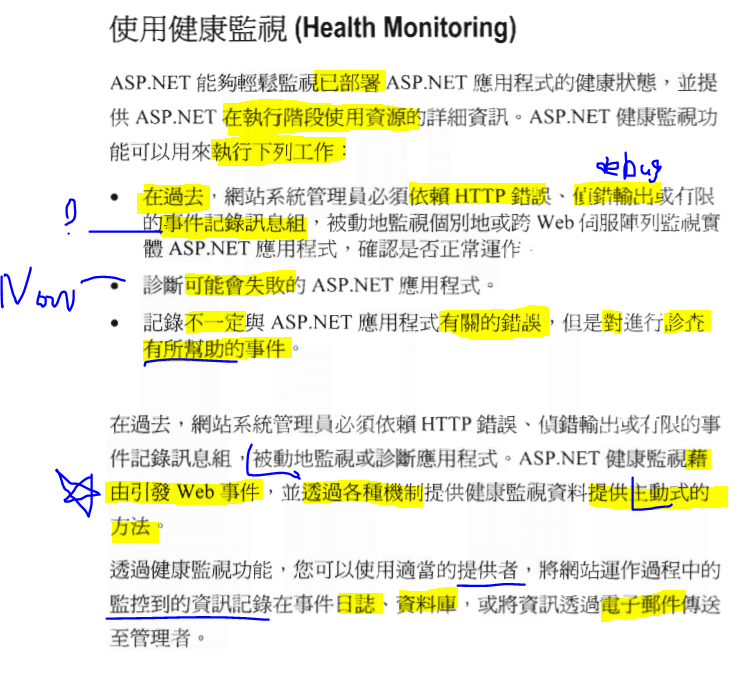
只放「更改資料庫」同時顯示Table

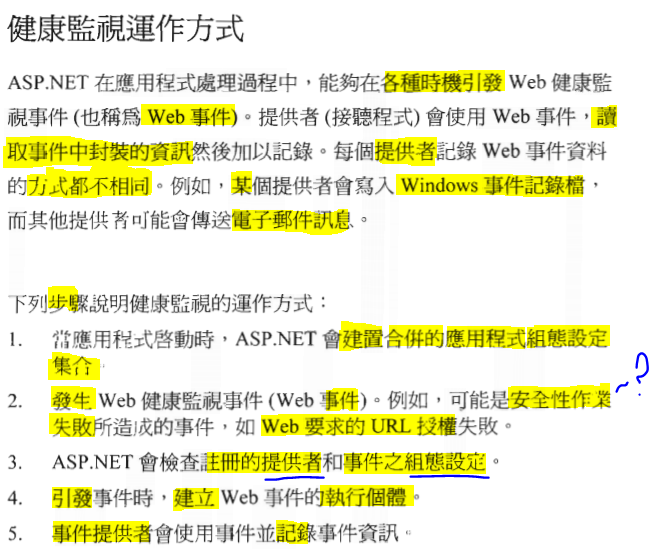
= > 每次按下去 都得到最新的結果，完全無視Cache

補充︰「更新GV」其實也可以這樣寫 ( 用 DataSourceID 屬性 )︰

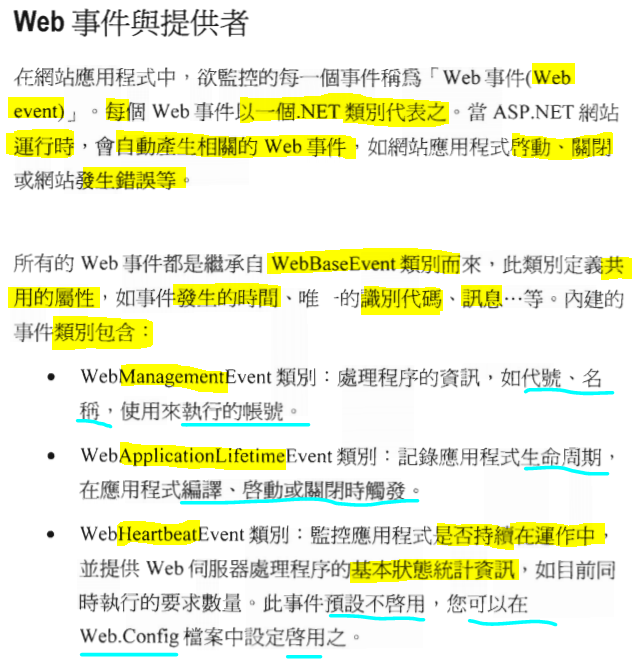


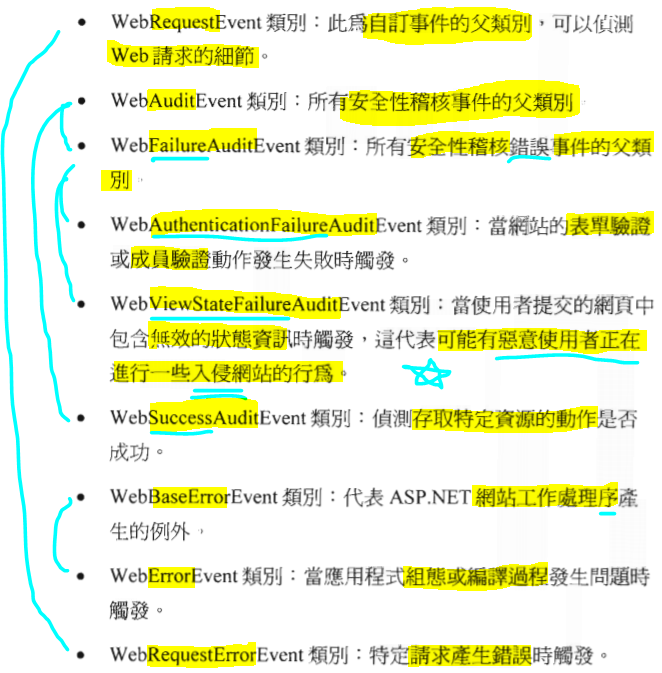
P 395 Health Monitoring



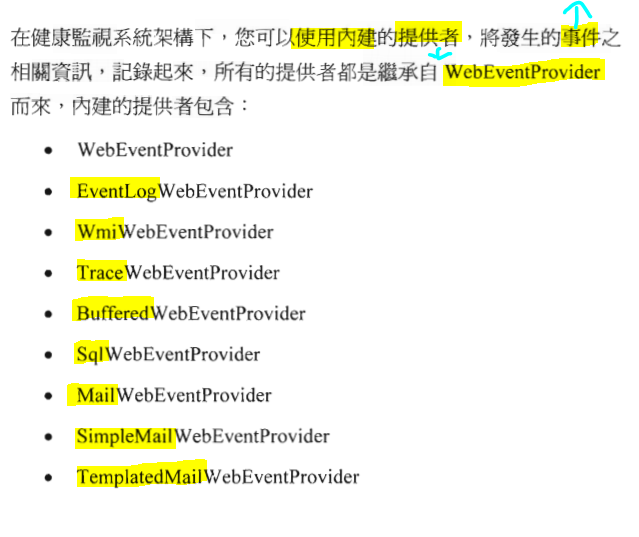


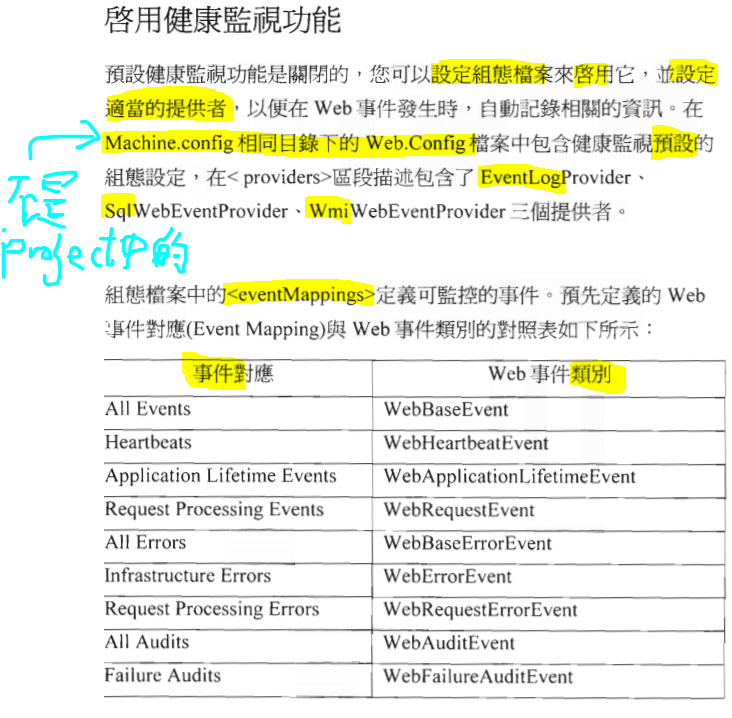
P 397





P 398





SuccessAudits | WebSuccessAuditEvent

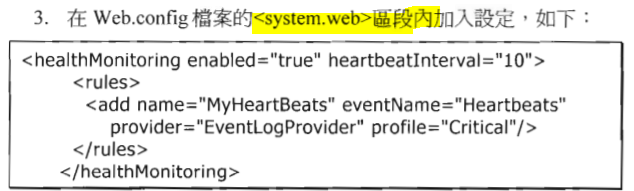


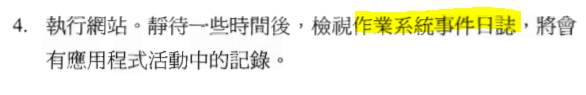
P 400

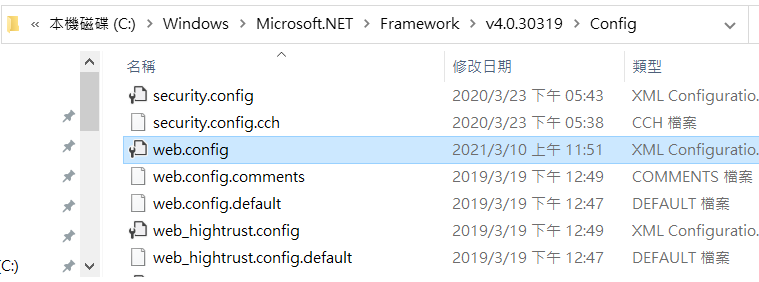


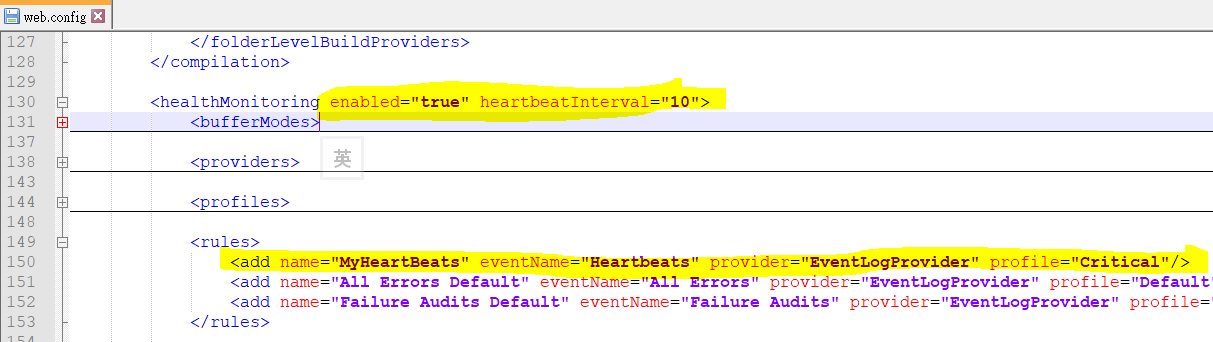
這個Web.config是 和 Machine.config 同目錄的那一個

P 402練習說︰



這個東東又是在哪裡？

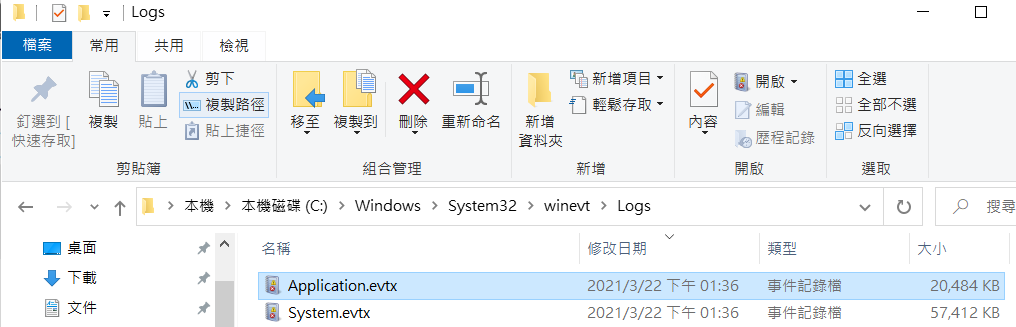


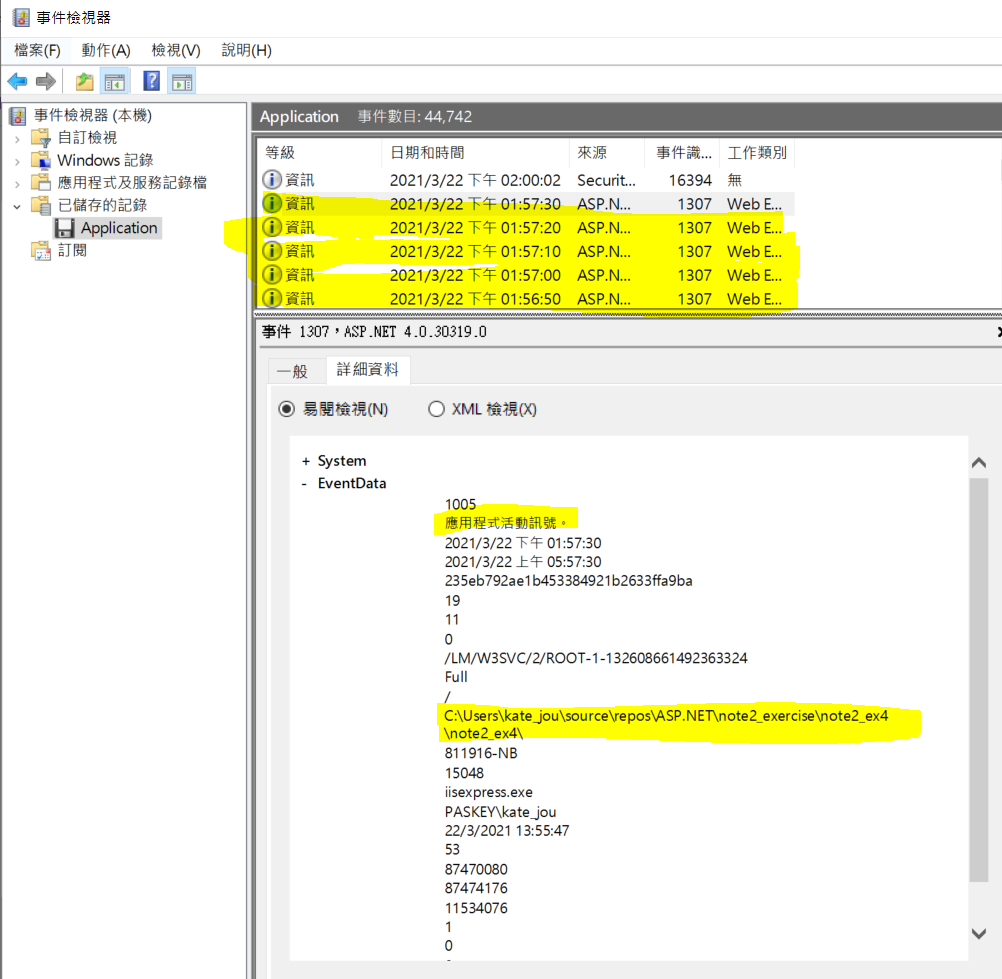


<healthMonitoring enabled="true" heartbeatInterval="10">

<add name="MyHeartBeats" eventName="Heartbeats" provider="EventLogProvider" profile="Critical"/>

我改好了，也隨便跑了一個網頁多於 10 秒。但是我找不到作業系統事件日誌在哪裡？





我已經把Web.config 回複正常，我感覺以後不太會用到？

回為我還沒有這麼高的層次？

也許以後網站真的有問題，我會來這裡看看有沒有紀錄。

作業系統事件日誌的地點，我是從這裡找到的︰

<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/444383/>

P 403 Finish