STATE MANAGEMENT

כמו שכבר ראינו HTTP הוא פרוקטוקול שלא משמר את המצב של הדף או המשתמש בו. ראינו בפרק הקודם איך ניתן להעביר מידע בין דפים ועכשיו נלמד איך לשמר מידע עבור כל הדפים בבת אחת. אנחנו נבדיל בין שמירת נתונים עבור כל המשתמשים לעומת שמירת נתונים עבור לקוח בודד.

Application

[01 State Management/application/]

נתחיל עם אובייקט ה application שהוא אובייקט שמאפשר שמירת נתונים עבור כל המשתמשים באתר והוא נשמר בזכרון על השרת. האובייקט הזה כבר לא בשימוש ובמקומו משתמשים באובייקט ה cache אותו לא\כבר למדנו. ה - application תופס מקום על הזכרון של האתר עצמו ולכן הוא לא יעיל לעומת ה cache שהוא מהיר וכאשר אין מקום הוא נמחק. בכל מקרה רק נראה דוגמה למקרה ותתקלו בapplication.

```
Application["UName"] = txtName.Text;
Application["UGrade"] = int.Parse( txtGrade.Text);
...

string name = (string)Application["UName"];
int grade = (int)Application["UGrade"];
```

שימו לב שהמידע נשמר גם כאשר סוגרים ופותחים שוב את הדפדפן, כל עוד הdevelopment server לא נסגר כמובן. כמו כן אם נפתח מספר דפדפנים כולם רואים את אותו המידע.

ניתן להכניס ל application ערך ברירת מחדל דרך הפונקציה הבאה אשר נמצאת בקובץ global.asax

```
void Application_Start(object sender, EventArgs e)
{
    // Code that runs on application startup

}
```

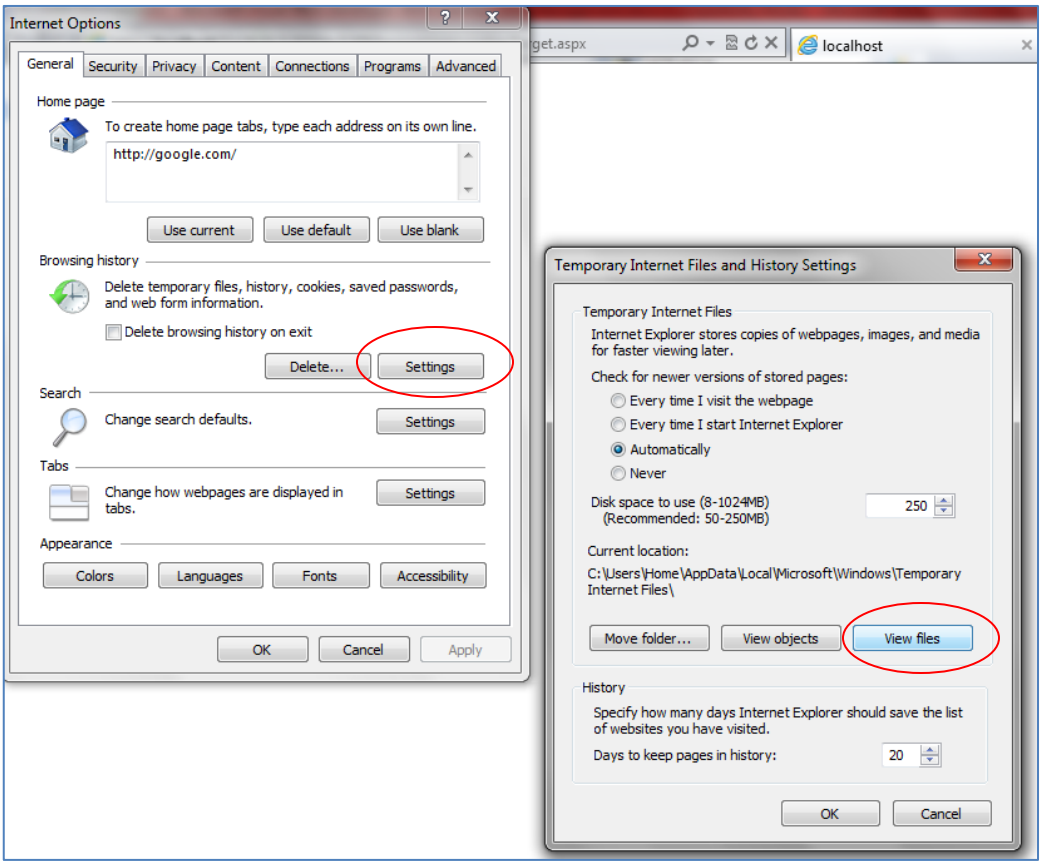
[01 State Management/cookie/]

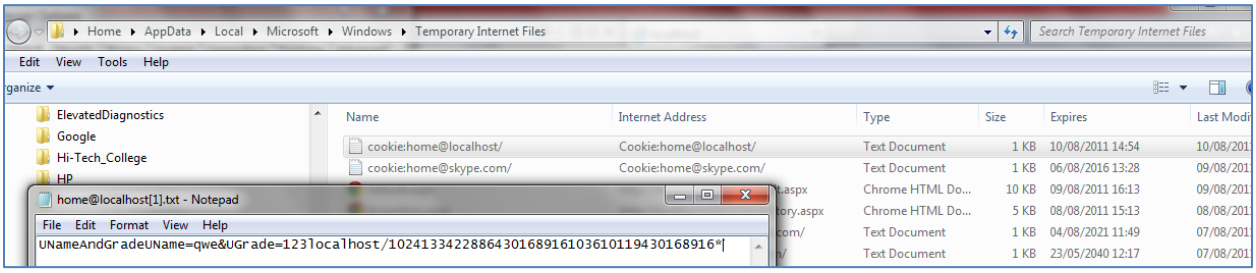
Cookie

עוגיה היא בעצם קובץ אשר נשלח ביחד עם הדף מהשרת ללקוח וחוזר ביחד עם הrequest של הלקוח בחזרה לשרת. על הקובץ הזה ניתן לאחסן מידע בצורת string. חסרון אחד הוא שה cookie יכול להכיל עד 4kb של מידע והוא נשמר גם לאחר סגירת הדפדפן אצל הלקוח. ברירת המחדל היא שהעוגיה תמחק כאשר סוגרים את הדפדפן אך ניתן להגדיר את המאפיין Expires לחיות גם לאחר שהדפדפן נסגר. כאשר נבקש שוב את האתר תישלח הcookie ונוכל לקרוא את תוכנה. חסרון שני הוא שניתן להגדיר בדפדפן את מניעת העבודה עם cookie.

```
HttpCookie myCookie = new HttpCookie("UNameAndGrade");
myCookie.Values.Add("UName", txtName.Text);
myCookie.Values.Add("UGrade", txtGrade.Text);
myCookie.Expires = DateTime.Now.AddSeconds(20);
Response.Cookies.Add(myCookie);

//for a single value in the cookie
//HttpCookie myCookie2 = new HttpCookie("UName");
//myCookie2.Value = txtName.Text;
//myCookie2.Expires = DateTime.Now.AddSeconds(20);
//Response.Cookies.Add(myCookie2);
...

if (Request.Cookies["UNameAndGrade"] != null)
{
    string name = Request.Cookies["UNameAndGrade"].Values["UName"];
    string gradeString = Request.Cookies["UNameAndGrade"].Values["UGrade"];
    res.Text = name + " - " + gradeString;
}

//for a single value in the cookie
//if (Request.Cookies["UName"] != null)
//{
//    string name = Request.Cookies["UName"].Value;
//    res.Text = name ;
//}
```

כך ניתן לראות את הcookie :





Session

[01 State Management/session/]

האובייקט הזה יודע לשמור מידע עבור כל משתמש בנפרד כך שהמידע נשמר כל עוד הדפדפן לא נסגר.
אובייקט זה גם הוא עובד בשיטת ה key\value. המידע נשמר על הזכרון של השרת.

```
Session["UName"] = txtName.Text;
Session["UGrade"] = int.Parse( txtGrade.Text);
...
string name = (string)Session["UName"] ?? "no name";
int grade = -1;
if (Session["UGrade"] != null)
{
    grade = (int)Session["UGrade"];
}
```

עכשיו ניתן לראות שכל משתמש (חלון של דפדפן) מוקצת עבורו יחידת זכרון יחודית וכל אחד יכול לראות רק את המידע שלו. ברגע שסוגרים את הדפדפן המידע נמחק!

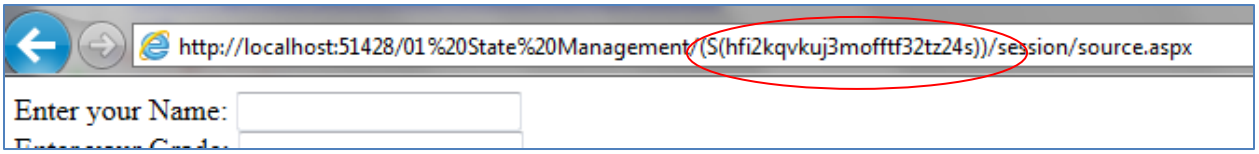
ניתן להכניס לsession ערך ברירת מחדל דרך הפונקציה הבאה אשר נמצאת בקובץ global.asax

```
void Session_Start(object sender, EventArgs e)
{
    // Code that runs when a new session is started

}
```

והשאלה היא כמובן איך המערכת מזהה מיהו כל משתמש?

ישנן שתי שיטות שבהן המערכת יכולה להשתמש:

1. יצירת cookie שמכיל ID יחודי שמתאים ליחידת הזכרון הספציפית עבור אותו ID. ה cookie נשלח הלוך וחזור בין הלקוח לשרת וכך ניתן לזהות את המשתמש כל עוד הוא לא סוגר את הדפדפן. – יש בעיה אם הדפדפן מוגדר כ disable ל cookie. נשמר כברירת מחדל עד 20 דקות אך ניתן להעריך את הזמן הזה.
2. URL Mangling – ה ID ישמר ממש כחלק מה URL



יש לשים לב לשימוש בלינקים רלטיווים ולא להשתמש באבסולוטיים!

ניתן להגדיר את האפשרויות דרך הweb.config

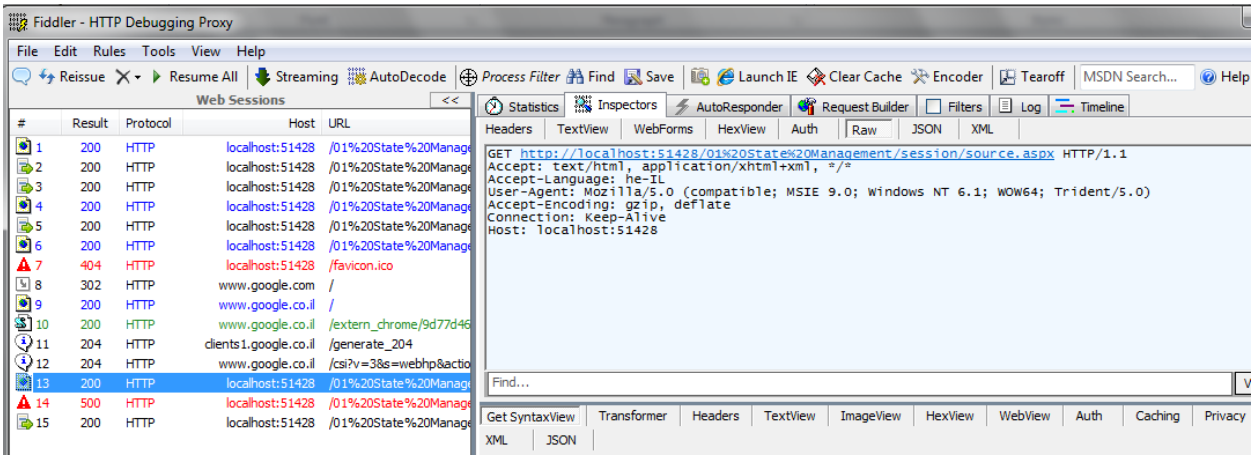
```
<system.web>
    <sessionState cookieless="true" />
```

: Cookieless

- True = להשתמש ב URL MANGLING
- cookie = False
- autoDetect = אם יש cookie אז להשתמש, אם אין אז URL

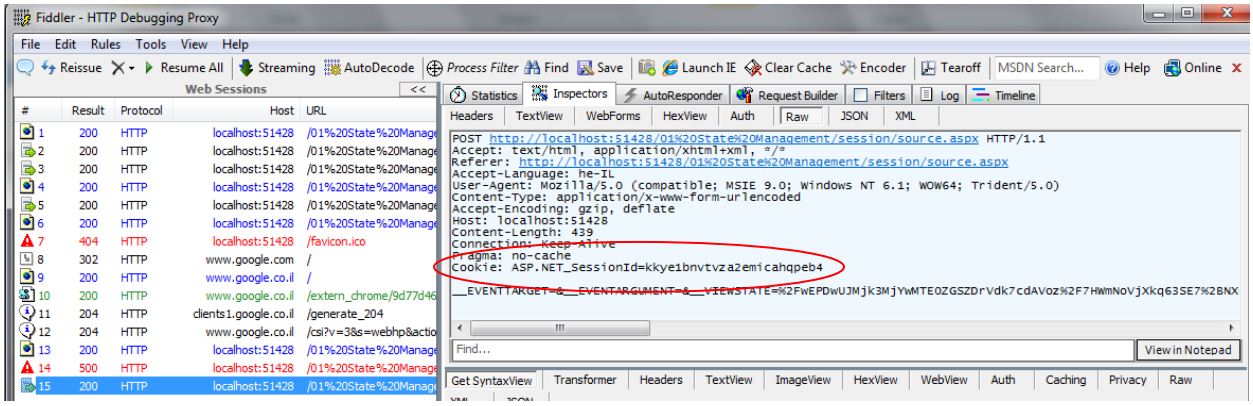
בכדי לראות את ה cookie של ה session, ניתן להוריד את התוכנה הבאה שנותנת את האפשרות לעקוב אחרי הנעשה ברשת

http://www.fiddler2.com/fiddler2/version.asp



ניתן לראות שאין cookie בגישה הראשונה ל source.aspx (נכון עבור IE, עבור CHROMEנשלח cookie גם בפעם הראשונה??)

בגישה השניה כבר נראה את הcookie



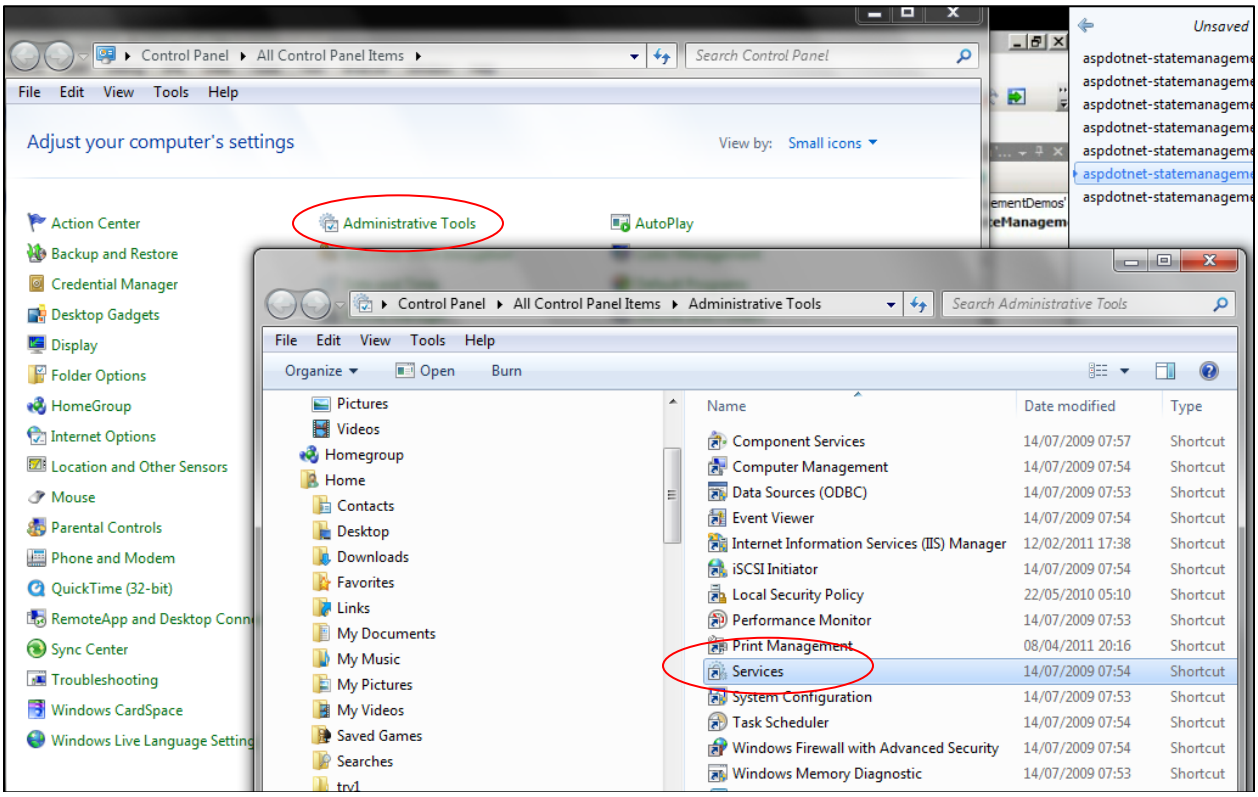
למדנו שהמידע עבור ה session מוחזק בזכרון, אבל מה קורה עם ה server נפל (או אם מחזיקים את האתר בחוות שרתים)? נאבד את המידע? איך נתמודד עם זה?

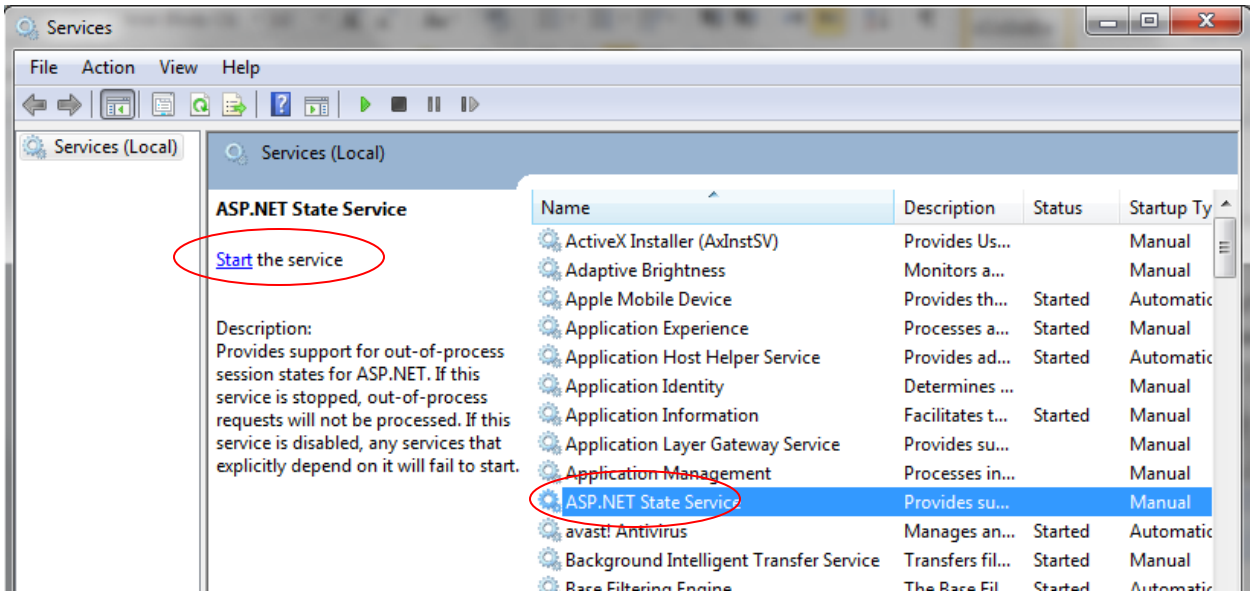
אז הפתרון לזה הוא שנחזיק את המידע עצמו על state server שהוא בעצם אפליקציה ניפרדת כך שאם האפליקציה של האתר נופלת עדין יש את המידע. (בנוסף ניתן להריץ את הstate server על מחשב ניפרד ולגשת עליו מחוות שרתים שלמה).

אז איך זה נעשה?

קודם כל נשנה את המאפין בweb.config מברירת המחדל שלו (`mode="InProc"`) ל

```
<sessionState cookieless="false" mode="StateServer"
                stateConnectionString="tcpip=127.0.0.1:42424" />
```





עכשיו לאחר שנריץ, גם אם נוריד את הdevelopment server (אבל לא נסגור את הדפדפן) ונרים אותו שוב דרך הVS, נוכל לראות שאכן הנתונים נשמרו לא על השרת של האתר (אותו הרי הורדנו ועכשיו אנחנו משתמשים באחד חדש).

ניתן גם להשתמש בSQL server במקום בstate server אבל זה מעבר לחומר.

בפועל בכל בקשה לדף נישלחת בקשה לקבל את הנתונים ואז עוד בקשה לעדכנם מתוך בסיס הנתונים שמחזיק את נתוני הsession.

בכדי למנוע roudtrip מיותרים ניתן להשתמש ב

```
<%@ Page Language="C#" EnableSessionState="True" %>

<%@ Page Langauge="C#" EnableSessionState="False" %>

<%@ Page Language="C#" EnableSessionState="ReadOnly" %>
```

דפים שלא כוללים session בכלל – False

דפים שרק קוראים מידע מהsession – ReadOnly

לשאר – True

Profile

מנגנון ששומר מידע על SQL Server עבור כל משתמש בנפרד והוא גם strongly typed , דהיינו המידע לא נישמר כסתם object אלה ממש לפי ה type שלו כך שלא צריך לעשות casting בעצמנו.

יתרון נוסף הוא שעבור משתמשים רשומים ה IDשלהם יהיה השם משתמש שלהם וכך אין צורך להמציא להם מספר יחודי כלשהוא.

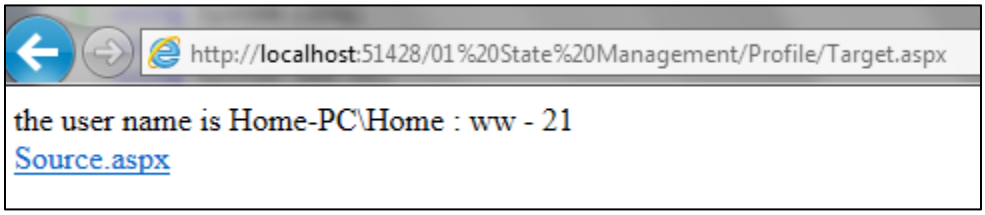
יש להגדיר את שמות וסוגי המשתנים ב:web.config

```
<system.web>
    <profile enabled="true">
        <properties>
            <add name="name" type="string" defaultValue="no name" allowAnonymous="true"/>
            <add name="grade" type="int" defaultValue="-1" allowAnonymous="true"/>
        </properties>
    </profile>
    <anonymousIdentification enabled="true"/>
```

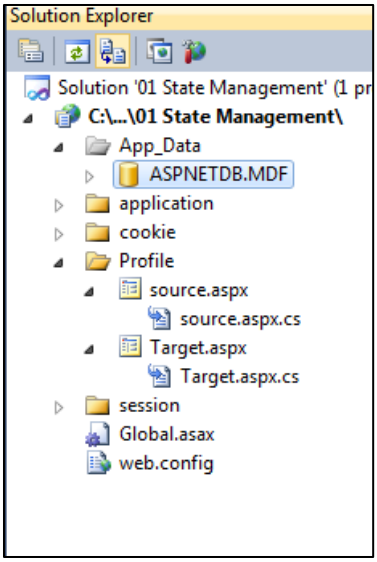
לאחר מכם יש לנו profile שהוא strongly typed

```
Profile.name = txtName.Text;
Profile.grade = int.Parse(txtGrade.Text);
...

res.Text = "the user name is " +  Profile.UserName +" : " +
                 Profile.name + " - " + Profile.grade.ToString();
```

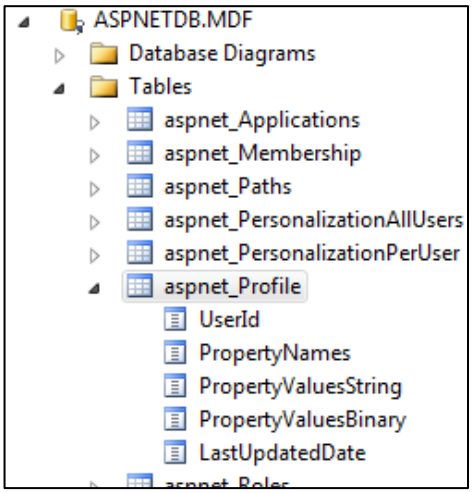
הנה התוצאה:



בבקשה הראשונה עבור דף שימו לב שלוקח די הרבה זמן וזה בגלל ש ASP בונה בסיס נתונים עבור המנגנון



ניתן לראות את הטבלה שמכילה את פרטי המידע

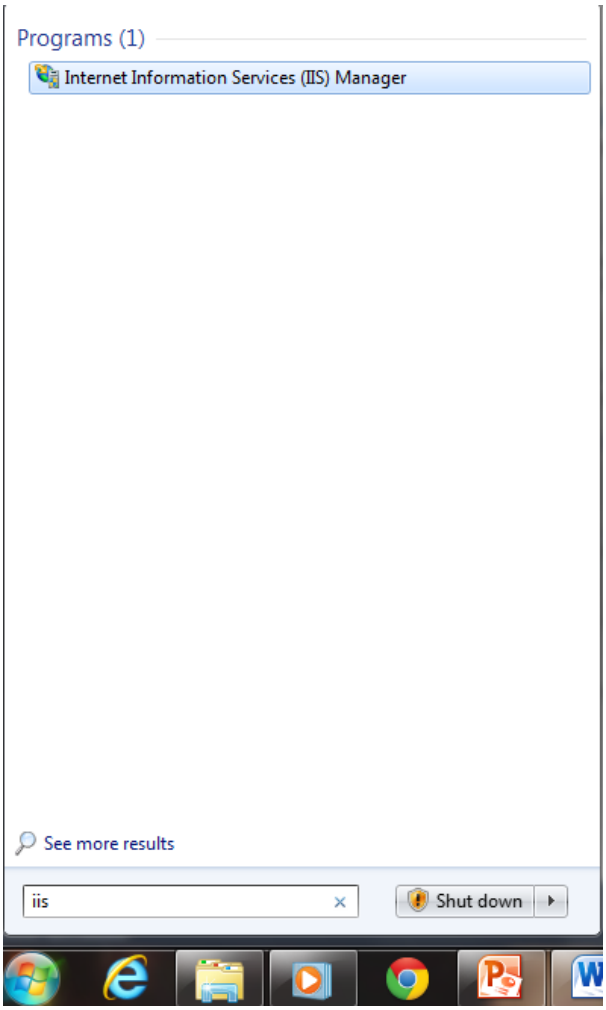


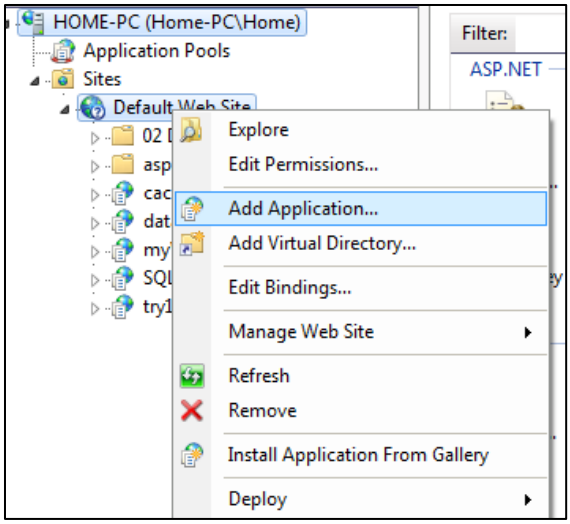
וגם את פרטי המידע עצמם

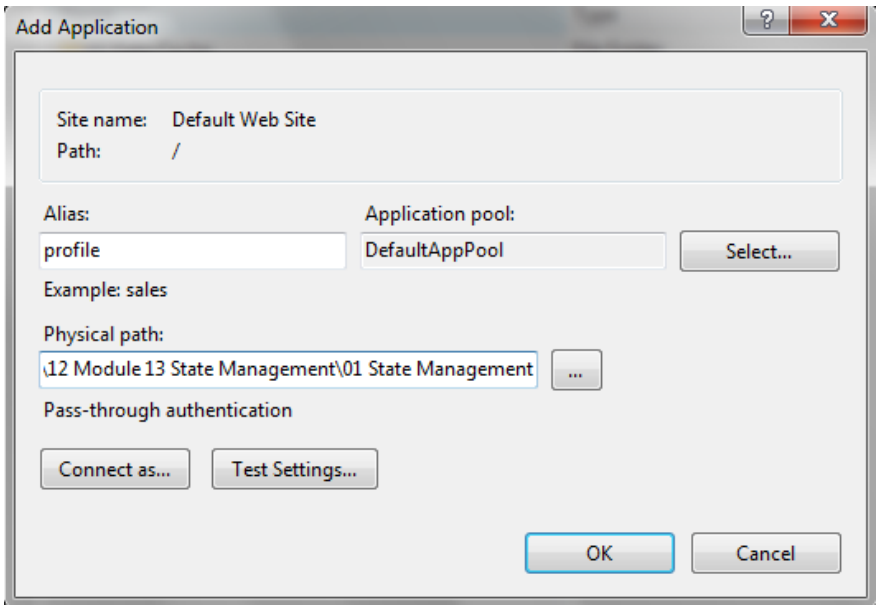
aspnet_Profile:...ATA\ASPNETDB.MDF) | web.config | application/source.aspx.cs | session/Target.aspx.cs | Profile/Target.aspx.cs

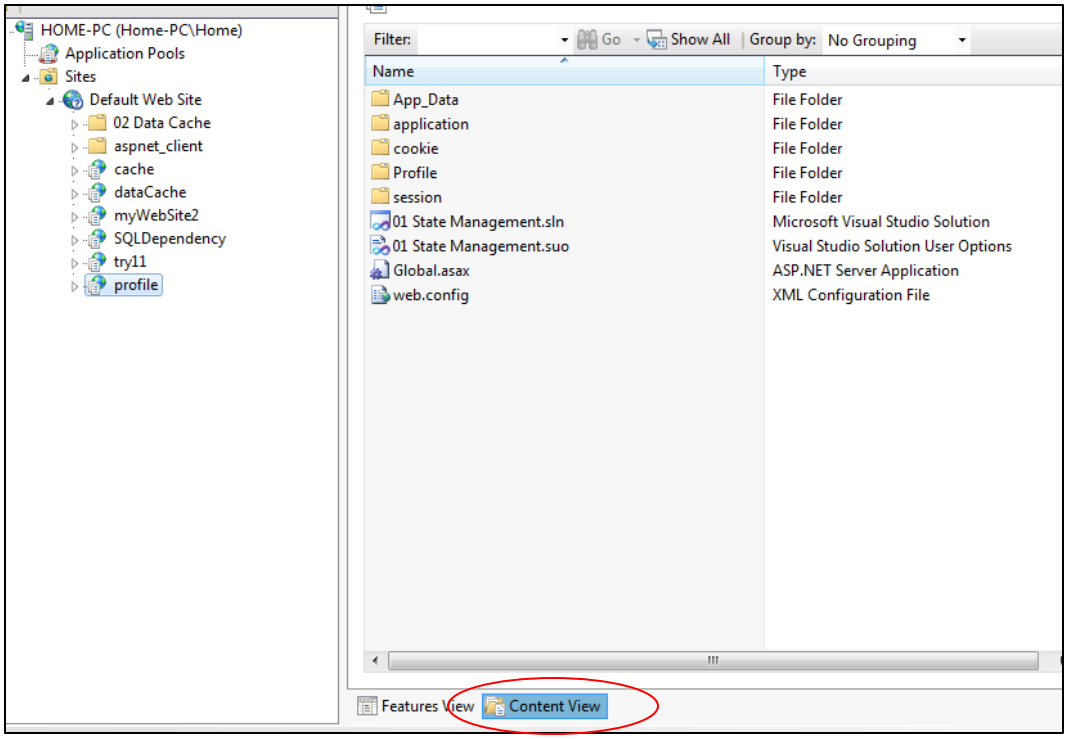
UserId	PropertyNames	PropertyValuesString	PropertyValue...	LastUpdatedDate
500-5da599af223b	grade:S:0:3:name:S:3:3:	333zzz	<Binary data>	10/08/2011 15:00:31
f6158db6-e5a3-41fc-a117-...	name:S:0:2:grade:S:2:2:	ww21	<Binary data>	10/08/2011 15:07:54
NULL	NULL	NULL	NULL	NULL

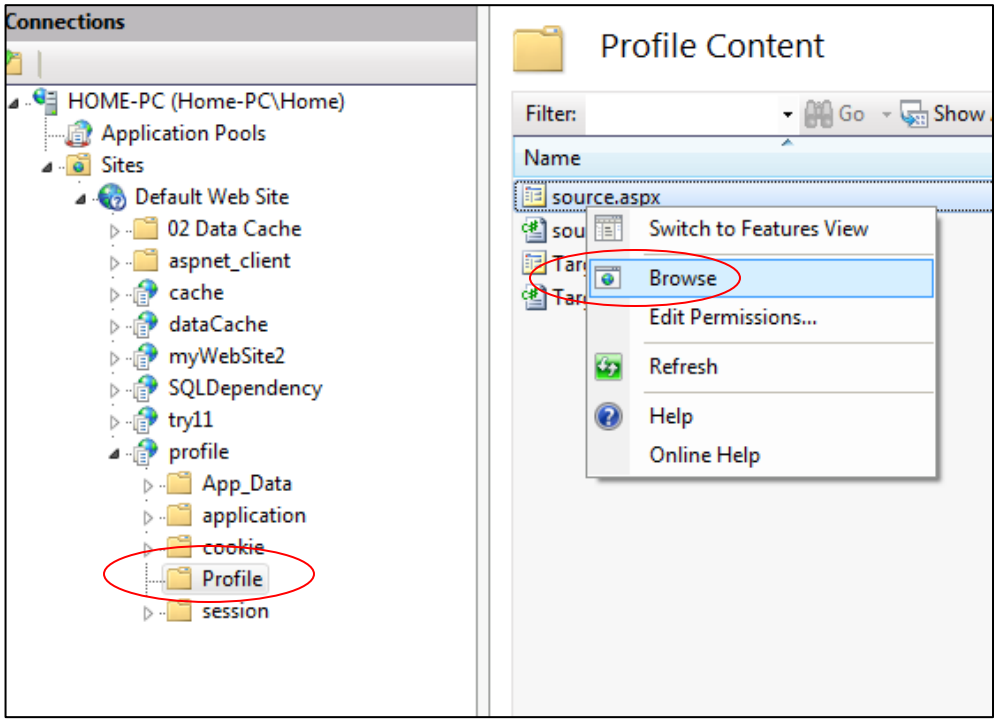
בכדי לראות שם משתמש שונה מזה של המערכת הפעלה איתו אנחנו עובדים גם על ה VSאנחנו צריכים להריץ ממש דרך ה IIS, וככה זה עובד:




HOME-PC (Home-PC\Home)
Application Pools
Sites
Default Web Site
Filter:
ASP.NET
Explore
Edit Permissions...
Add Application...
Add Virtual Directory...
Edit Bindings...
Manage Web Site
Refresh
Remove
Install Application From Gallery
Deploy

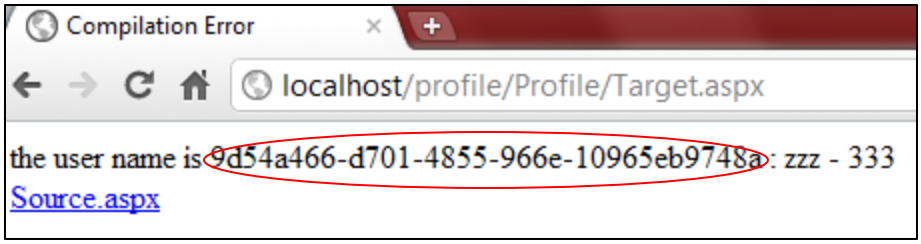

Add Application
Site name: Default Web Site
Path: /
Alias:
profile
Application pool:
DefaultAppPool
Select...
Example: sales
Physical path:
\12 Module 13 State Management\01 State Management
...
Pass-through authentication
Connect as...
Test Settings...
OK
Cancel





שימו לב שצריך להוריד את ההגדרות של `//using System.Linq;`
וגם ב web.config

```
<!--<compilation debug="true" targetFramework="4.0"/>-->
```



[אם יש בעיות ניתן למחוק אותו מה VS, ואז לייצר אותו דרך ה IIS ע"י הרצה ראשונית מה IIS. ורק אז להריץ מה VS(אולי לאחר dbClick על ה MDF)]

ניתן למחוק את המידע עבור מתשמשים שלא היו אקטיביים בX זמן האחרון ע"י

```
ProfileManager.DeleteInactiveProfiles(ProfileAuthenticationOption.All,DateTime.Now.AddMin
utes(-10));
```

ניתן לשנות את הבסיס נתונים בו המידע מוחזק אבל זה לא בחומר הנלמד.