
# SessionState בקובץ web.config

ג'סטין-יוסף אנג'ל

מסמך זה הורד מהאתר http://www.underwar.co.il.

מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

הזכויות שמורות ל**ג'סטין יוסף אנג'ל**.

ג'סטין-יוסף אנג'ל

the_j_angel@hotmail.com

http://blogs.israelblog.co.il/justin_angel/

העשרה שבועית בנושא והפעם מאמר מקיף על האפשרויות של sessionState בתוך web.config. (נושא שנתתי הסבר מזורז עליו השבוע). (תודה לאוריאקס על הכותרת)

sessionState מאפשר זיהוי ומעקב על פני מספר webforms של משתמש מסויים. דוגמאות טובות הן: לעקוב אחרי הרשאות למערכת (למשל, Login), לעקוב אחרי שינויים במצב העבודה של המשתמש עם האפליקציה (למשל, עגלת קניות), לעקוב אחרי אופן הגלישה של משתמש באתר ומאוחר יותר לנתח את הממצאים כדי להראות הרגלי גלישה באתר.

ספציפית במאמר זה נתמקד באפשרויות השונות לשימוש ב-sessionState כפי שהן באות לכדי ביטוי ב-web.config ו-machine.config:
1. כבוי. אם לא משתמשים ב-sessionState, זה חסכון אדיר במשאבי מערכת שמכוונים את ה-sessionState לכבוי. ככה המערכת לא צריכה להעסיק סתם משאבים ולדאוג לזה. תתפלאו, אבל יש הרבה אפליקציות בסיסיות ואפילו כאלו מתקדמות שאין להן שום צורך ב-sessionState.

```
<configuration>
  <system.web>
    <sessionState mode="off">
    </sessionState>
  </system.web>
</configuration>
```

כמו כן, ישנה אפשרות שב-webform ספציפי באפליקציה לא עובדים בצורה ישירה עם sessionState. בהתאם לכך, נכבה אותו ספציפית לאותו webform ובאותו טופס נוסיף:

```
<%@ Page EnableSessionState="false" %>
```

אי-אפשר לציין ולהדגיש מספיק עד כמה זה חשוב במצב שבו לא משתמשים ב-sessionState לדאוג שהוא לא יהיה פעיל.

2. InProc - המצב הזה הוא המצב הראשון והבסיס ביותר של שימוש ב-sessionState. מה שלמעשה InProc אומר הוא שהכל התהליך של שמירת sessionState מתבצע In-process. הווה אומר, על הפרוסס\Appdomain של ה-ASP.net.
החסרונות - אין לו את היכולות של האפשרויות הבאות. בנוסף, חסרון חשוב הוא שב-machine.config נמצאת תגית processModel אשר בין השאר מציינת מתי יש לאתחל את הפרוסס של ה-ASP.net (כאשר אין תגובה לתקופת זמן, כאשר יש עומס על השרת, כאשר אין תגובה ברשת וכך הלאה) וכאשר התהליך מאתחל את עצמו אז גם נמחק ה-sessionState.

כדי ליישם את מצב זה יש לשנות את mode ל-InProc בweb.config או machine.config:

```
<configuration>
  <system.web>
    <sessionState mode="InProc">
    </sessionState>
  </system.web>
</configuration>
```

3. StateServer - הקוסנפט מאוד פשוט: בואו נשמור את הנתונים על ה-sessionState ב-windows service.
יתרונות:
א. הפרוסס נפרד לחלוטין מהפרוסס של ה-ASP.net, ככה שניתן לאתחל את אפליקציית ה-IIS ולא לאבד את מידע ה-sessionState. הווה אומר, באתחול כפוי (ע"ע processModel) או באתחול ידני (מתוך inetmgr) ה-sessionState נשמר.
ב. העובדה שהנתונים במצב זה נשמרים ב-Windows service מאפשרים גם חיבור לשרת מרוחק. הווה אומר, שרת\מחשב אחד אשר שומר נתוני sessionState לכמה שרתים. הדבר הזה נהדר לחוות שרתים אשר מריצות אותה אפליקציית רשת. דבר ראשון, אם מדובר באפליקציית רשת גדולה שדורשת יותר משרת אחד (למשל amazon.com) אז צריך שהמשתמש יוכל לעבוד על כל השרתים בלי להתחבר מחדש כל חמש שניות.
דבר שני, אם נופל שרת אפליקציה אחד ושרת אפליקציה אחר צריך לבוא ולמלות את המקום שלו אז לא נאבדים נתוני ה-sessionState.

למעשה מה שאנו עושים זה ליצור שרת שאחת ממטרותיו היא לשמור לכל חוות השרתים את נתוני ה-sessionState. אבל בד בבד מומלץ בתהליך זה גם לשרתים בדדים, ולא רק לחוות שרתים.

חסרונות:
א. ככל שעולים יותר ויותר בפתרונות ל-sessionState מבחינת גמישות ואפשרויות ככה גם צריך יותר RAM\זכרון על השרת.
ב. כל דבר אשר שומרים ב-sessionState חייבת להיות אפשרות לעשות לו serialize. הווה אומר, כל אובייקט אשר לא ניתן לעשות לו serialize או שיזרוק שגיאה כאשר נכניס אותו ל-sessionState או פשוט לא ישמר.
ג. אין לאפשרות זאת כתיבה על הדיסק הקשיח. ככה שאתחול השרת אשר מפעיל את ה-state server עדיין ימחוק את כל נתוני ה-sessionState. אין שום דרך לגבות את הנתונים לדיסק ולטעון אותם מאוחר יותר.

איך ליישם:
א. Administrative Tools / Services --> להפעיל את ה- ASP.NET State Service

ב. יש לשנות את המode בתוך קובץ ה-web.config (או ה-machine.config) ל-"StateServer". כמו כן יש להוסיף את הנתונים הבאים על השרת אשר מריץ את ה-windows service: כתובת IP ודרך איזה פורט מתחברים. את שני הנתונים הללו נשים ב-stateConnectionString. בנוסף ניתן לקבוע משתנה stateNetworkTimeout, אשר קובע תוך כמה שניות מתרחש timeout בפרוטוקול ה-tcp/ip. חשוב לציין שאירוע ה-timeout נרשם בלוגים של השרת.

```
<configuration>
  <system.web>
    <sessionState
     mode="StateServer"
     stateConnectionString="tcpip=127.0.0.1:42424"
     stateNetworkTimeout=15>
    </sessionState>
  </system.web>
</configuration>
```

(בחיבור למעלה מבצע חיבור דרך Loopback. הווה אומר, השרת מתחבר לעצמו דרך 127.0.0.1. בנוסף באם אין תגובה מה-windows service תוך 15 שניות יתרחש timeout.)

4. SQLserver - נתוני ה-sessionState ישמרו בתוך מסד נתונים SQL server. למעשה מה שקורה הוא שיש ליצור מסד נתונים חדש בשם ASPstate, ובתוכו נתוני ה-sessionState נשמרים כנתונים מסוג BLOB (קובץ בינארי גדול).

יתרונות:
א. שרידות נתוני sessionState אדירה. אפשר לאתחל את ה-IIS, את תהליך ה-SQLserver, את שרת ה-SQL server עצמו, ושום דבר חוץ מלירות בהארד-דיסק לא יגרום לאיבוד נתונים.
ב. כמו ב-stateServer יש יתרון גדול בשימוש בחוות שרתים. רק צריך לזכור (זה תקף גם ל-stateServer) שבכדי שיהיה sessionState משותף לאותה אפליקציה על שני שרתים שמחוברים לאותו SQLserver/StateServer צריך אותו נתיב לאפליקציה.

חסרונות:
א. ככל שעולים יותר ויותר בפתרונות ל-sessionState מבחינת גמישות ואפשרויות ככה גם צריך יותר RAM\זכרון על השרת.
ב. כל דבר אשר שומרים ב-sessionState חייבת להיות אפשרות לעשות לו serialize. הווה אומר, כל אובייקט אשר לא ניתן לעשות לו serialize או שיזרוק שגיאה כאשר נכניס אותו ל-sessionState או פשוט לא ישמר.

ג. חובה שיהיה מסד נתונים SQL server. מדובר בחסרון אם האפליקציה עצמה עובדת עם מסד נתונים אחר, ולכן צריך לתחזק מסד נתונים נוסף.

כיצד מיישמים:
1. מריצים על שרת ה-SQLserver את InstallSqlState.sql.
2. אם עובדים ב-trusted connections עם שרת ה-SQLserver צריך לשנות את הבעלות על מסד ה-ASPstate.
3. אם עובדים עם SQL authentication, צריך ליצור משתמש שיהיה לו גישה למסד ASPstate (כולל הרצת פרוצדורות).
4. יש לשנות את mode בתוך קובץ ה-web.config (או machine.config) ל-"SQLServer". כמו כן יש לציין מחרוזת חיבור למסד (שם המסד, שם משתמש וסיסמא) במשתנה sqlConnectionString. בנוסף, כמו ב-stateServer ניתן לציין Timeout בנתון ה- stateNetworkTimeout.

```
<configuration>
  <system.web>
    <sessionState
     mode="SQLServer"
     sqlConnectionString="data source=server_name;user
id=user_id;password=password"
     stateNetworkTimeout=15>
    </sessionState>
  </system.web>
</configuration>
```

בבחירת מצב ה-sessionState יש לשקלל את היתרונות של כל מצב (שרידות, נגישות ואפשרויות אשר נפתחות) מול החסרונות של שימוש רב יותר של משאבי מערכת.

בנוסף, קיימים שני משתנים נוספים בתוך תגית ה<sessionState>.
1. cookieless - מציין אם ה-sessionState עובד עם או בלי עוגיות בצד לקוח.

```
<configuration>
  <system.web>
    <sessionState cookieless="false">
    </sessionState>
  </system.web>
</configuration>
```

אם בחרנו לעבוד בלי עוגיות הנתיב היחסי של פנייה בין webforms ישתנה.
למשל:

http://localhost/(lit3py55t21z5v55vlm25s55)/Application/SessionState.aspx

במקום:

http://localhost/Application/SessionState.aspx

ואת זאת מיישמים בכך שנכוון את משתנה ה-cookieless ל-true (קרי, לא להשתמש בעוגיות צד-לקוח).

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

2. נתון ה-timeout. בנתון זה נרשום כמה דקות לקוח יכול להיות בלתי-פעיל (בלי בקשות לשרת) לפני שהשרת זורק את ה-sessionState שלו. באפלקיציות מאובטחות יש טעם בלקצר תקופה זו, לעומת זאת באפליקציות בעלות גישה חופשית נהוג לכוון גישה זו למספר שעות.

```
<configuration>
  <system.web>
    <sessionState timeout="20">
    <!-- for 20 minutes -->
    <sessionState timeout="300">
   <!-- for 5 hours -->
    </sessionState>
  </system.web>
</configuration>
```

מקווה כי למדתם והשכלתם ממאמר זה, אשמח לענות על שאלות הבהרה ונוספות. כמו כן אני ממליץ שבאם נתקלתם ביישום SQLserver תפנו למאמר המצורף השלישי.

בביליוגרפיה עיקרית זמינה בקישורים.

שבוע טוב,
ג׳סטין-יוסף אנג׳ל

כתובות אינטרנט נילוות:


קישור #1: ASP.NET Session State ב-MSDN -
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspnet/html/asp12282000.asp

קישור #2: Performance ASP.NET Session State: Architectural and Considerations - http://blogs.msdn.com/tims/archive/2003/11/21/57453.aspx

קישור #3: Using SQL Server for asp.net session state -
http://idunno.org/dotNet/sessionState.aspx