Response.redirect או Server.Transfer?

ג'סטין-יוסף אנג'ל

ג'סטין-יוסף אנג'ל

the_j_angel@hotmail.com

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

לא פעם נתקלתי עם מפתחים חדשים ומנוסים כאחד שלא בטוחים במספר סוגיות בתחום היותר אפור של הפן הטכני בתכנות דוט נט. פעם בשבוע אפרסם נושא אחד עם מאמרים הדנים בו.

והפעם: **Response.redirect או Server.Transfer?**

מאמר #1: סקירה כללית על הנושא - http://www.developer.com/net/asp/article.php/3299641
מאמר #2: פירוט על נושא ה-flow - http://haacked.com/archive/2004/10/06/1308.aspx
מאמר #3: סיכום והרחבה - http://www.csharpfriends.com/Articles/getArticle.aspx?articleID=15

סיכום של הנקודות שיש לזכור:
Response.recdirect ו-Server.transfer לכאורה מבצעות אותה פעולה, אך יש הבדלים מהותיים בעבודה של שני המתודות הללו. לא מדובר רק על העברה בין דף לדף, אלא על היכן העברה מתרחשת, מה היכולות שלה וכיצד המשתמש יקלוט אותה.

1. תחבירית:

```
// Server.Transfer
// # First Overload:
Server.Transfer(string path)
Server.Transfer("myPage.Aspx")
// # Second Overload:
Server.Transfer(string path, bool PreserveForm)
Server.Transfer("myPage.Aspx",true)


// Response.Redirect
// # First Overload:
Response.redirect(string URL)
Response.redirect("myPage.aspx")
// # Second Overload:
Response.redirect(string URL, bool endResponse)
Response.redirect("myPage.aspx", false)
```

2. הבדלים בקונספט: Response.Redirect מיועד לשימוש לצד לקוח, Server.Transfer מיועד לשימוש בצד שרת. כל שאר ההבדלים נובעים מיכולותיו ומגבלותיו של כל צד.

3. שימור נתוני טופס: בעזרת Server.Transfer ניתן לשמר את המצב של הטופס הנוכחי לדף אליו מעבירים. הווה אומר, האוספים Request.Form ו-Request.QueryString. ניתן להשיג זאת בכך שמכוונים את המשתנה preserveForm ל-true. היתרון של הדבר שלמעשה כל העבודה שנעשתה עד כה בדף המפנה תהיה זמינה לשימוש בדף אשר אליו מפנים. החסרון של הדבר היא ששימור מצב הטופס דורש משאבי מערכת וכך הופך את התוכנית לפחות יעילה. חשוב לציין ש-viewState הרבה יותר מאובטח ומכיל את היכולת הזאת, אך גם יקר יותר בביצועים.

4. העברה בצד לקוח: כאשר אנו מבצעים Response.redirect מה שקורה למעשה הוא שהשרת שולח ללקוח הודעה שהדף עבר למקום אחר והדפדפן של הלקוח פונה לאותה הכתובת.

```
HTTP/1.1 302 Object moved
Server: Microsoft-IIS/5.0
Location: somewhere/newlocation.aspx
```

ניתן לראות את ההודעה אשר נשלחת ללקוח. החסרון בכך שמדובר בעוד PostBack לשרת.

5. העברה בצד שרת: כאשר אנו מבצעים Server.Transfer השרת למעשה מבצע עיבוד של webform אחר במקום זה שהלקוח ביקש. וכל זה, בלי שהלקוח יודע כלום. היתרון בכך שנחסך PostBack. החסרון הוא בכך שהדפדפן של הלקוח אינו מודע לכל התהליך הזה. הכתובת ב-address bar תישאר זהה לכתובת המבוקשת המקורית ולא לכתובת אשר אליה הפננו. באם ירצה הלקוח לשים סימניה על הדף הזה, הוא למעשה יסמן את הדף הקודם.

6. תכלס', חסכון: Server.Transfer מוריד את העומס על השרת והלקוח כאחד.

7. העברה הודעות: כאשר מבצעים העברה בעזרת Server.Transfer ניתן להעביר הערות\פרמטרים\הודעות בעזרת אסופת Content.