

## SITE NAVIAGATION

אתרי אינטרנט מכילים לרוב יותר מדף אחד. ישנן מספר דרכים לעבור מדף אחד למשנהו. ישנן גם מספר שיטות להציג את הניווט באתר. בפרק זה נלמד איך לנווט באתר הצורה קלה ויעילה.

ראשית יש צורך להבין מהו URL = uniform resource locator

### Relative and absolute url

#### [01 URLs]

זוהי כתובת האינטרנט של הדף אותו רוצים להציג ניתן לעשות זאת ע"י הכתובת היחסית דהיינו בלי השם של האתר\השרת

למשל

Pag2.aspx or ..\..\dir2\page2.aspx

או ע"י הכתובת מהשורש של האתר

~\dir1\dir2.page.aspx

השיטה השניה עדיפה כי היא תעבוד גם כאשר מזיזים את הדף בו ההכוונה הזו נמצאת.

מציאת השורש :

~ רק לפקדי צד שרת

/ לפקדים של HTML

ניתן לשנות את השם של ספריית השורש בעזרת המאפיין virtual path של הפרוייקט.

ניתן כמובן לעבור גם דרך כתובת אבסולוטית למשל:

<http://localhost:51382/nir/dir2/dir3/Default3.aspx>

<http://weather.walla.co.il/?w=/1/402/w>

## מעבר מדף לדף

עד כה ראינו שניתן לעבור מדף אחד למשנהו דרך פקדי קישור עליהם המשתמש צריך ללחוץ אך ניתן לעבור מדף אחד לאחר גם דרך הקוד.

### [02 Redirect and Transfer]

השיטה הראשונה (והישנה) היא `Response.Redirect()` היא פשוט מקבלת את הלינק אותו הדפדן יבקש. יש לשים לב שהשיטה הזו יוצרת round trip נוסף. בעצם מהשרת נשלחת הודעה לדפדפן לבקש דף אחר. בשיטה זו ניתן לקרוא לדף באתר אחר. כתובת הדפדפן משתנה לדף החדש.

```
Response.Redirect("Target.aspx"); //GET
```

השיטה השנייה היא `Server.Transfer()` היא מקבלת את הלינק אותו הדפדן יבקש. יש לשים לב שהשיטה הזו לא יוצרת round trip נוסף, אלה פשוט מתוך השרת מחזירים כבר את הדף המבוקש. בשיטה זו לא ניתן לקרוא לדף באתר אחר. רק הדף המבוקש מיוצר ונשלח לדפדפן. (אך ניתן גם לגשת לדף הישן, נראה זאת מאוחר יותר). כתובת הדפדפן נשארת עם הדף הישן.

```
Server.Transfer("Target.aspx"); //POST
```

השיטה השלישית היא Cross-PagePostBack היא מבצעת קריאה לדף השני מתוך הדף הראשון כך שבכלל לא מגיעים לקוד של השרת בכדי לקרוא לדף השני. כל הקידוד הוא ב JS בתוך הקוד של הדף הראשון. דהיינו זה חוסך בקשה של הדף הראשון, זה גם חוסך ROUNDTRIP לדפדפן לעומת REDIRECT, ולעומת TRANSFER זה משנה את כתובת הדפדפן כך שהמשתמש יודע שהוא עבר לדף אחר. ניתן ליישם רק על פקדי שרת. (שמממשים `System.Web.UI.WebControls.IButtonControl` interface, בשפה אילו `Button`, `LinkButton`)

```
<asp:Button runat="server" ID="Button1" Text="move to target cross-page.aspx"
onclick="btnTargetCrossPage_Click" PostBackUrl="~/Target.aspx" />
```

שימו לב שהאירוע `btnTargetCrossPage_Click` לא יתבצע לעולם!

## העברת מידע בין דפים

### [03 QueryString and Context]

לאחר שראינו כיצד ניתן לעבור בין דפים, חשוב להבין כיצד להעביר ביניהם מידע. למשל דף ראשון מקבל מהמשתמש מידע כלשהוא ומעביר אותו לעיבוד והצגה לדף שני.

עבור השיטה הראשונה של Response.Redirect() אנחנו נשתמש ב QueryString בכדי להעביר את המידע לדף הבא. זוהי מחרוזת שמשורשרת לכתובת הדף עם הסימן ? לאחר מכן מגיעים ערכים של ערך=שם&ערך=שם וכו'

```
string url = "Target.aspx?";
url += "Name=" + txtName.Text;
url += "&Grade=" + txtGrade.Text;

Response.Redirect(url); //GET
```

הקריאה נעשית ע"י האינדקס בתוך האובייקט Request.QueryString

```
if (Request.QueryString.Count != 0)
{
    string name = Request.QueryString["Name"] ?? "";
    int grade = int.Parse( Request.QueryString["Grade"] ?? "0" );
}
```

עבור השיטה השניה נשתמש באובייקט CONTEXT להחזקת הערכים. שימו לב שהוא מחזיק סוגים שונים של ערכים ולא רק מחרוזת. ה context הוא אובייקט שקיים עבור כל request. שמו המלא הוא HttpContext. ז"א שעבור ה request הבא אין לנו את הערכים מהrequest הקודם. ולכן השימוש בו הוא בעיקר עבור server.Transfer

```
Context.Items["Name"] = txtName.Text;
Context.Items["Grade"] = int.Parse( txtGrade.Text );

Server.Transfer("Target.aspx"); //POST
```

והקריאה

```
string name = (string)Context.Items["Name"] ?? "";
int grade = (int)Context.Items["Grade"];
```

בכדי להגיע לקונטרולים של הדף ניתן להשתמש גם ב:

```
string str = Request.Form["txtName"];
```

עבור השיטה השלישית (ניתן גם ליישם עבור השיטה השנייה שגם בה משתמשים ב POST) אנחנו ניקרא את המידע מהדף הראשון ע"י אובייקט PreviousPage. ניתן לעשות זאת ע"י המרה מפורשת בצורה הבאה:

```
if (PreviousPage != null)
{
    if (PreviousPage.IsCrossPagePostBack)
    {
        //option 1:
        //for the case with the directive
        //casting in order to get the new properties Name and Grade
        //pay attention without the directive
        //"<%@ PreviousPageType VirtualPath="~/Source.aspx" %>" there is no
        intellisense on the Source page

        Source p = (Source)PreviousPage;
        string name = p.Name;
        int grade = p.Grade;

        lblResult.Text = string.Format("CROSS PAGE(POST) without
        directive<br>Your name is: {0} and your garde is: {1}", name, grade);
    }
}
```

או ע"י שימוש ב directive

```
<%@ PreviousPageType VirtualPath="~/Source.aspx" %>
```

והקוד הבא

```
string name2 = PreviousPage.Name;
int garde2 = PreviousPage.Grade;

lblResult.Text = string.Format("CROSS PAGE(POST) with directive<br>Your name is:
{0} and your garde is: {1}", name2, garde2);
```

שימו לב שהדף PreviousPage נוצר רק כאשר נקרא לו במפורש מהדף עליו הגענו.

IsCrossPagePostBack – משמש בכדי לדעת האם היתה קריאה רגילה לדף או האם מתבצעת קריאה ע"י מנגנון ה cross page. בעיקר יהיה לזה שימוש בדף ה source.aspx בכדי לא לבצע פעולות לחינם.

## URL – Rewriting

### [05 URL Rewriting]

ע"י שימוש בשורות הבאות ניתן יהיה למפות כתובת שמתבקשת מהדפדפן לדף מסויים דבר שנותן גמישות לגבי השמות של הכתובות. אפשר לתת כתובות קלות יותר למשתמש.

```
<configuration>
  <system.web>
    <urlMappings >
      <add url="~/Page2.aspx" mappedUrl="~/dir1/dir2/Page2.aspx"/>
```

יפנה את הכתובת הפשוטה הזו `~/Page2.aspx` לכתובת המסובכת הזו `~/dir1/dir2/Page2.aspx`

ניתן אפילו לעשות שימוש בQueryString למשל

```
<add url="~/nir.aspx" mappedUrl="~/Default2.aspx?Name=NIR"/>
```

## Menus

### [04 Menus]

עכשיו נלמד על שלוש מחלקות לבניית תפריטים עבור הדפים באתר.

לשם כך עלינו להכיר את האובייקט SiteMap – ניתן להגדיר אותו ע"י

Add New Item → Site Map

זהו קובץ XML אשר יכיל את בסיס הנתונים עבור האובייקט של התפריט. הקובץ מכיל טאגים של `siteMapNode` אשר ניתן לקונן לפי ההיררכיה הרצויה. כל טאג מכיל את ה URL של הדף הרצוי, הכותרת של הלינק שלו באתר ותיאור שיופיע ב TOOLTIP. אין קשר בין ההיררכיה הפיזית של הקבצים לבין ההיררכיה של התפריט.

שלושת סוגי התפריטים שנכיר הם:

- `SiteMapPath` זהו תפריט שמראה את המיקום הנוכחי באתר והוא נותן את האפשרות לעלות במעלה ההיררכיה בחזרה עד לתפריט הראשי. אובייקט זה מקושר באופן אוטומטי ל `SiteMap`.

- **Menu \ TreeView** אילו אותם תפריטים בעלי צורה שונה. כל אחד נותן את האפשרות לעבור בין כל דפי האתר גם במורד ההיררכיה ולא רק במעלה ההיררכיה כמו התפריט הקודם שראינו. אנחנו נקשר אותו גם כן ל SiteMap למרות שניתן לחברו למקורות מידע אחרים גם כן.

