



## AJAX - תקשורת ישירה עם השרת ללא רענון

שושן (psclil)

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

אין להפיץ מסמך זה במדיה כל שהיא, ללא אישור מפורש מאת המחבר. מחבר המסמך אינו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן ונכונות התוכן של הנושאים המופיעים במסמך. עם זאת, המחבר שה את מירב המאמצים כדי לספק או המידע המדויק והמלא ביותר.

הזכויות למסמך שייכות ל**שושן (psclil)**. תגובות נא לשלוח ל-[psclil@gmail.com](mailto:psclil@gmail.com).

## תוכן עניינים

---

1.....	AJAX - תקשורת ישירה עם השרת ללא רענון
3.....	הקדמה
4.....	מתחילים! .....
5.....	דף תגובה
6.....	הקדמה לצד לקוח – האובייקט XMLHttpRequest
9.....	בניית דף צד לקוח .....
11.....	דוגמא שנייה – שימוש קצת יותר הגיוני .....
15.....	עוד הצעה טכנית קטנה .....
16.....	סיכום .....

## הקצנה

### קודם כל:

אם חלק מהמונחים או האובייקטים שיוזכרו פה לא מובנים לך, זה בסדר, אלו סתם שמות מפוצצים שכדאי להכיר, אבל בטח כבר התעסקת איתם בלי לדעת.

### מה זה AJAX ?

Asynchronous JavaScript and XML - AJAX, כלומר, זהו מונח שמתאר טכניקת פיתוח אתרים ליצירת דפים אינטראקטיביים תוך שימוש בקומבינציה של HTML, CSS, DOM, JavaScript ו-XMLHttpRequest, כדי לתקשר דו כיוונית עם השרת, למרות שיש גם דרכים אחרות לעשות זאת. בקיצור, זאת דרך מעולה לקבל תשובה מהשרת בלי לבקש דף חדש, או לרענן את הדף, דלי להסתיר מהמשתמש frames או יישומי Java.

### בעצם מה אני הולך/כת לעשות פה?

לבנות אפליקציה ממש פשוטה עם ASP ו-JavaScript רק כדי להסביר איך התקשורת עובדת. לשם ההסבר, יבואו 2-3 דוגמאות.

### באילו דפדפנים הסיפור הזה עובד?

אני לא טוען לשלמות של הסקריפט, אבל השתדלתי שיעבוד ברוב הדפדפנים. בחנתי אותו אישית בדפדפן IE6 ובדפדפן FF1.5, אבל אני יודע שהוא אמור לעבוד ברוב הדפדפנים הפופולאריים ללא קושי או שגיאות לא צפויות.

### אם אני לא יודע/ת X, אני לא צריכה להיות פה:

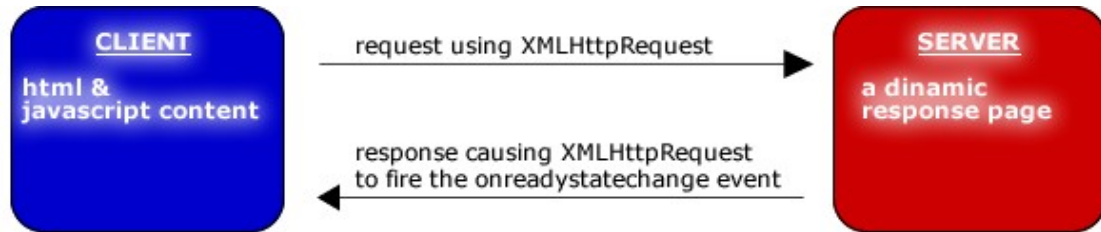
צריך לדעת HTML - בסיסי מאוד, JavaScript, ASP - בסיסי מאוד. - זהו, הכל ניתן ללימוד מהספרייה של underwar!

### רציתי לומר לך (הכותב) שטעית/יכולת להשתפר ב X.

ניתן לפנות אלי באימייל [psclil@gmail.com](mailto:psclil@gmail.com) או בפורום [underwar.co.il/forum](http://www.underwar.co.il/forum).

מתחילים!

הסבר תכני קצר על העמוד לקרות במסמך זה  
המבנה של האפליקציה יהי כזה:

© Shoshan Cohen, [psclil@gmail.com](mailto:psclil@gmail.com)

### הסבר הדיאגרמה:

צד לקוח שמקבל תוכן HTML ו-JavaScript - שולח בקשה לשרת בעזרת השימוש באובייקט XMLHttpRequest ומבקש דף דינאמי (במקרה שלנו ASP). השרת מעבד את הבקשה ומחזיר תגובה דרך האובייקט. לאחר קבלת התגובה ע"י האובייקט, נורה אירוע [אירוע כמו onclick רק שקוראים לו-] onreadystatechange ואז בהתאם לתוכן התגובה, מופעלת פעולת JavaScript אצל המשתמש.

### מבלבלים!?

שוב אני חוזר אין בעיה עם בלבול משום שההסבר כרגע הוא על הרקע הטכני ודברים שחוזרים על עצמם ללא לוגיקה מיוחדת בכל פעם שרוצים לממש קטע מהאתר בעזרת Ajax, ניתן לבצע זאת גם ללא הבנה או רענון הידע.

3 התגובה

נתחיל עם הדף הדינאמי בצד השרת (דף התגובה) – אני אעבוד עם asp אבל אפשר גם php, asp.net, Html, קובץ txt וכו' קבצי Html ו-Txt היו רק דוגמא. אנו זקוקים לדף דינאמי שיחזיר לנו תשובה דינאמית בהתאם לבקשה שלנו. בעקרון, הרעיון הוא להחזיר תוכן לפי בהקשה. את הבקשה נקבל בעזרת Query String. נתחיל עם דף פשוט שמחזיר "hello" אם הפקודה היא "h" ואחרת "unknown command".

**File: AJAX\_response.asp**

```
<%
Select Case Request.QueryString("action")
  Case "h"
    response.write("hello world")
  Case Else
    Response.write("unknown command")
End select
%>
```

זה אמנם דף פשוט, אבל לא הולכים להיות דפים הרבה יותר מסובכים מזה.

**לדוגמא,**

כרגע אם נבקש את מהשרת את הקובץ AJAX\_response.asp נקבל את הפלט "unknown command" ואם נבקש את AJAX\_response.asp?action=h נקבל "hello world".

**חשוב לדעת ש:**

כדי שהשרת לא ישלח "תשובה שגויה", חשוב שלא יכתב טקסט לפני או אחרי הקוד, אפילו לא רווחים. עלינו לשלוט בתגובת השרת ביד נוקשה.

## הקדמה לזן האובייקט XMLHttpRequest

**הקדמה**  
האובייקט XMLHttpRequest נועד כדי לבקש דפים מהשרת. בדפדפנים שונים מכריזים עליו בדרכים שונות.

### Internet Explorer 5.0+

```
Var http = new ActiveXObject("Microsoft.XMLHTTP");
```

### Mozilla & FireFox 1.0+ & Safari 1.2+ וכנראה עוד כמה:

```
Var http = new XMLHttpRequest();
```

### טבלת השיטות של האובייקט:

שיטה	תיאור
<code>abort()</code>	עוצרת את הבקשה האחרונה
<code>getAllResponseHeaders()</code>	מחזירה את כל הכותרות של הבקשה (Headers) בתור מחרוזת
<code>getResponseHeader("headerLabel")</code>	מחזירה את הערך של כותרת אחת - headerLabel
<code>open("method", "URL" [, asyncFlag [, "userName" [, "password" ]]])</code>	ממנה כתובת בקשה, שיטה ועוד מספר תכונות אפשריות. **שיטה - "Get" או "Post"
<code>send(content)</code>	שולחת את הבקשה. אפשרות לשליחת מידע נוסף.
<code>setRequestHeader("label", "value")</code>	מוסיפה/קובעת כותרת ו/או ערך לכותרות שישלחו בעת בקשת הדף (Http Headers)

## כעת, נגדיר פונקציה שתחזיר לנו אובייקט מאותחל

```
function createRequestObject() {
  var ro;
  if (window.XMLHttpRequest) {
    try {
      ro = new XMLHttpRequest();
    } catch(e) {
      //some kind of a weird mistake...you choose what you want to do
      ro = false;
    }
  } else if(window.ActiveXObject) {
    try{
      ro = new ActiveXObject("Msxml2.XMLHTTP");
    } catch(e) {
      try{
        ro=new ActiveXObject("Microsoft.XMLHTTP");
      } catch(e) {
        //don't know what to do...you choose
        ro = false;
      }
    }
  }
  return ro;
}

// דוגמא לשימוש
var http = createRequestObject();
```

## ועכשיו, עוד תוספת להקדמה, רשימת התכונות והאירועים של האובייקט:

תכונה	תיאור
<b>onreadystatechange</b>	אירוע שמתקיים בכלא פעם שהמצב של האובייקט משתנה
<b>readyState</b>	מצב האובייקט - מספר שלם: לא מאותחל = 0 טוען = 1 נטען = 2 אינטראקטיבי = 3 סוים = 4
<b>responseText</b>	גרסת מחרוזת של התגובה מהשרת
<b>responseXML</b>	גרסת XML של התגובה מהשרת
<b>status</b>	קוד מצב מספרי כמו 404 בשביל "Not Found" ו-200 בשביל "OK"
<b>statusText</b>	מחרוזת המתלווה לקוד המצב

נשתמש באירוע onreadystatechange ובתכונה readyState כדי לדעת מתי סיימנו לקבל את התגובה מהשרת ועלינו להתחיל לפעול בצד הלקוח.

איך שולחים בקשה ?

```
function sndReq(action) {
    http.open('get', 'AJAX_response.asp?action='+action,true);
    http.onreadystatechange = handleResponse; // מוגדרת לא עדיין
    http.send(null);
}
```

איך מטפלים בתגובה ?

כמובן, אין דרך אחת תטפל בתגובה, תלוי מה היא מייצגת ואיך נודה הדף תכנן את פעולתו. בדוגמא הראשונה, דרך הטיפול תהיה הצגת הטקסט בתוך תגית קיימת – זוהי דוגמא פשוטה.

```
function handleResponse() {
    if(http.readyState == 4){
        var response = http.responseText;
        document.getElementById("output").innerHTML=response
        //כשתראי/ה את הקוד המלא תבין לאיזו תגית אנחנו ניגשים
    }
}
```

מה הקוד המלא ?  
בעמוד הבא.



בניית דף לז' לקוח

\*רק לשם נוחות, אני אשתמש בדף asp. אפשר באותה מידה להשתמש בקובץ asp שנמצא באותו השרת.

File: AJAX\_request.asp

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1256"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1256" />
<title>AJAX try...</title>
<script type="text/JavaScript">

function createRequestObject() {
    var ro;
    if (window.XMLHttpRequest) {
        try {
            ro = new XMLHttpRequest();
        } catch(e) {
            //some kind of a weird mistake...you choose what you want to do
            ro = false;
        }
    } else if(window.ActiveXObject) {
        try{
            ro = new ActiveXObject("Msxml2.XMLHTTP");
        } catch(e) {
            try{
                ro=new ActiveXObject("Microsoft.XMLHTTP");
            } catch(e) {
                //don't know what to do...you choose
                ro = false;
            }
        }
    }
    return ro;
}

// דוגמא לשימוש
var http = createRequestObject();

function sndReq(action) {
    http.open('get', 'AJAX_response.asp?action='+action,true);
    http.onreadystatechange = handleResponse; // עדיין לא מוגדרת
    http.send(null);
}

function handleResponse() {
    if(http.readyState == 4){
        var response = http.responseText;
        document.getElementById("output").innerHTML=response
        //כשתראי/ה את הקוד המלא תבין לאיזו תגית אנחנו ניגשים
    }
}

</script>
<!-- BODY START -->
```

```
<body>
<h1>Questions:</h1>
<a href="javascript:sndReq('h');">what is the most stupid sentence
ever?</a>
<a href="javascript:sndReq('what is a toast');">how smart are you?</a><br
/>
<br />output:<br />
<pre id="output"></pre><!--this is the output tag... -->
</body>
</html>
```

### זהו, סיימנו עם הדוגמא הראשונה!

בעצם, מה שאמור לקרות זה שכשהמשתמש לוחץ על הקישור, נשלחת בקשה לשרת. כשמוחזרת תשובה, התוכן שלה מוכנס לתגית pre שה-id שלה הוא content. כשהמשתמש לוחץ על "what is the most...the most" התשובה שמוחזרת היא "hello world" וכשהוא לוחץ על "how smart...", מתקבלת התשובה "unknown command"

## זאגמא אַייה - אינאלי יתיר היינטיג

### מה הסיפור הפעם?

אוקיי, עד עכשיו הבנו את הטכניקה בערך, עכשיו נבין למה באמת כדי להשתמש בה ובמה היא מקלה/עוזרת. הדוגמא: ישנו טופס הרשמה לדואר אלקטרוני באתר השירותים [k.co.il](http://k.co.il). ישנם בערך שמונה פרטים שעל המשתמש למסור. ברצוננו לתת למשתמש לדעת בזמן הקצר ביותר, ובלי טרחה של הגשת מאות טפסים (10) אם שם שמשתמש פנוי, ואם לא אילו שמות קרובים פנויים.

### הקוד של הלקוח ואחריו הסברים:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1256"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1256" />
<title>AJAX try...</title>
<style type="text/css">
h1{
font-family:Verdana, Arial, Helvetica, sans-serif;
font-size:18px;
font-weight:bold;
margin:0;
padding-bottom:5px;
}
body,td,th{
font-family:Verdana, Arial, Helvetica, sans-serif;
font-size:11px;
}
</style>
<script type="text/javascript">
function createRequestObject() {
    var ro;
    var browser = navigator.appName;
    if(browser == "Microsoft Internet Explorer"){
        ro = new ActiveXObject("Microsoft.XMLHTTP");
    }else{
        ro = new XMLHttpRequest();
    }
    return ro;
}

var http = createRequestObject();

function sndReq(action) {
    http.open('get', 'AJAX_response.asp?action='+action,true);
    http.onreadystatechange = handleResponse;
    http.setRequestHeader("User-Agent","shoshan bot")
    http.send(null);
}

function handleResponse() {
    if(http.readyState == 4){
        var response = http.responseText;
        eval(response);
    }
}

function checkUser(name) {
    http.open('get', 'AJAX_response.asp?action=checkUser&name='+name,true);
    http.onreadystatechange = handleResponse;
    http.send(null);
}
```

```

function checkName () {
c=document.daForm.daName;
if (c.value.length<3){
alert('name should be 3 chars long')
return;
}
//name is ok...now lets check...
checkUser(c.value);
}

function showBadUser (users) {
//show message
d=document.getElementById("badUser");
d.style.visibility="visible";
d.style.display="block";
//options
s=document.daForm.nameList;
o=s.options;
while (o.length>0){
o[0]=null;
}
users=users.split("|");
for (i=0;i<users.length;i++){
o[i]=new Option(users[i]);
}
//hide other message
d=document.getElementById("goodUser");
d.style.visibility="hidden";
d.style.display="none";
}

function showGoodUser () {
//show message
d=document.getElementById("goodUser");
d.style.visibility="visible";
d.style.display="block";
//hide other message
d=document.getElementById("badUser");
d.style.visibility="hidden";
d.style.display="none";
}

function showSelectedName () {
t=document.daForm.daName;
s=document.daForm.nameList;
o=s.options[s.selectedIndex];
t.value=o.text;
}
</script>
</head>
<body>
<form action="" method="post" name="daForm">
<h1>Choose User Name:</h1>
User:
<input name="daName" type="text" size="20" maxlength="20" />
<a href="#" onclick="checkName()">Check</a><br />
<div id="goodUser" style="display:none;visibility:hidden;"><br />
user name is good
</div>
<div id="badUser" style="display:none;visibility:hidden;"><br />
The name is not good.<br />
you can choose another name:<br />
<select name="nameList" size="4" onchange="showSelectedName()">
<option>psclil</option>
<option>nir</option>
<option>someone</option>
</select>
</div>
</form>
</body>
</html>

```

## הסברים

הפונקציה `handleResponse` השתנתה כך שהיא מריצה את קוד ה-`JavaScript` שמתקבל מהשרת (ראה עוד עצה טכנית קטנה), בהתחלה, ה-`div`-ים שה-`id` שלהם הוא `goodUser` ו-`badUser` הם מוסתרים. הפונקציה `showBadUser(users)` מקבלת מחרוזת עם רשימה של שמות שמוצעים למשתמש במקום, מופרדים ב-`"|"`, ומציגה אותה בתוך תיבת בחירת שנמצאת ב-`div "badUser"`, כשהמשתמש בוחר שם, תיבת הטקסט `"daName"` משתנה לערך שנבחר.

עכשיו, דף התגובה-

```

<%
set cnn = server.createObject("ADODB.Connection")
Set Rs = Server.CreateObject("ADODB.RecordSet")
cnn.open "DRIVER={Microsoft Access Driver
(*.mdb)};DBQ="+server.MapPath(".")+"\users.mdb"
function userNotExists(nm)
    set Rs=cnn.Execute("SELECT * FROM users WHERE name='&nm&'")
    userNotExists=Rs.eof
end function
    select case request.QueryString("action")
        case "checkUser"
            dim n
            n=Request.QueryString("name")
            replace n,"'",""
            n=trim(n)
            IF not(userNotExists(n)) THEN
                response.Write("showBadUser('")
                for i=0 to 9
                    if userNotExists(n&i) then
                        response.write(n&i&"|")
                    end if
                next
                response.write("Choose Another Name")
                response.write("'");")
            ELSE
                response.Write("showGoodUser();")
            END IF
        case else
            response.write("alert('unknown command');")
    end select
%>

```

## הסבר:

בעצם, בודקים במסד נתונים אם השם (והשמות החלופיים) נמצאים, ושולחים תגובה בהתאם.

רעיונות נוספים לשימוש:

1. בהגשת טופס, ישנו תהליך ארוך בצד שרת. ולא רוצים שהמשתמש יגיש טופס ויאלץ לחכות בלי שום פלט לגבי ההתקדמות.
2. ביצוע פעולות על מסדים גדולים יחסית מבלי לרענן את כל הדף, אלא רק ע"י עדכון החלקים בטבלה שהשתנו, וגם זאת ע"י JavaScript, כלומר יותר מהר ובלי רענון – זוג מנצח.
3. הנה דוגמא נחמדה – בדף חיפוש במשתמשים בפורום web wiz, תוך כדי שהמשתמש מקליד את טקסט החיפוש, מופיעה רשימת משתמשים שאולי יתאימו לחיפוש, והמשתמש יכול לבחור אחד מהם מן הרשימה. זאת ממש לא פעולה מסובכת – משתמשים בתגית .Select

## עוז הנעה טכנית קאה

**בשביל יותר שליטה על המתרחש בצד לקוח,** אני מציע בשביל לקבל יותר שליטה על העדכון והאפשרויות הרבות בצד הלקוח, שדף התגובה יחזיר פקודות JavaScript מופרדות בנקודה פסיק (";"), ובצד הלקוח, הטיפול בקבלת מידע יהיה כזה:

```
eval (response) ;
```

### כלומר,

צד הלקוח פשוט יריץ את פקודות ה JavaScript שנשלחו ע"י השרת. – כך יש הרבה יותר שליטה על המתרחש אצל המשתמש.

### בנוסף,

אני ממליץ להגדיר פונקציות JavaScript שידוע מראש השימוש בהן, ושהן נפוצות, כדי לחסוך בכמות המידע שנשלח מהשרת, ובכלל בשביל סדר וניקיון.

---

## סיכום

---

### במה השתמשנו?

XMLHttpRequest – אובייקט לתקשורת דו כיוונית עם השרת,  
(HTML, DHTML (JavaScript  
ו-ASP בשביל דפי שרת דינאמיים.

### מה למדנו?

איך לתקשר עם השרת דו כיוונית במהירות, בלי תחמונים או משחקים במהירות ובפשטות.

### אבל עדיין לא הבנתי!

אפשר לפנות אלי באימייל או בפורום (מפורט בתחילת המסמך)

### האם יבואו מסמכי המשך?

כחחח, כאילו שמישהו באמת ישאל את זה!  
קצת הגזמתי.

### אפשר עוד דוגמא שכתבת?

אהממ, כן בטח,  
להלן מוד שנכתב למערכת הפורומים webwiz,  
אפשר להבין ממנו די הרבה גם בלי להכיר את המערכת עצמה  
(אבל כמובן, מומלץ להוריד אותה ואולי אפילו לנסות אותה).  
[מוד השלמה אוטומטית](#) – במוד משולבות כמה מהשיטות שהוסברו במסמך זה יחד עם עבודה עם  
מסדי נתונים ודפי ASP דינאמיים שאכן מבצעים לוגיקה בצד השרת.