HTTP Headers-ה

פרוטוקול HTTP
הפונקציה header
שליחת מידע דחוס
Browser Cashing-ה
עבודה עם cookies

© 2009 Haim Michael. All Rights Reserved.

פרוטוקול HTTP

כאשר הדפדפן מקבל קובץ משרת זה מתרחש בהמשך לבקשה לקבלת הקובץ אשר נשלחת מהדפדפן לשרת. בקשה שנשלחת מדפדפן לשרת כוללת HTTP Headers שמספקים אינפורמציה נוספת.

דוגמא:

```
GET / index.html HTTP/1.1
Host: www.xperato.com
Accepted-Encoding: UTF-8
User-Agent: Fireworks/1.2
```

בדוגמא זו רואים בקשה שיוצאת מדפדפן לשרת בבקשה לקבל ב-GET קובץ בשם index.html. כמו כן. הבקשה כוללת headers שמספקים פרטים נוספים. מ-headers אלה ניתן ללמוד שהבקשה נעשית על פי הכללים של HTTP 1.1, הדפדפן מסוגל לקבל קבצים ב-encoding מסוג UTF-8 וסוג הדפדפן ששולח את הבקשה הוא Fireworks מגירסה 1.2.

כאשר השרת מחזיר את תשובתו בהמשך לקבלת הבקשה הוא שולח אותה בצירוף headers שמספקים אינפורמציה נוספת.

דוגמא:

```
HTTP/1.1 200 OK
Server: Apache
Content-Type: text/html
Content-Encoding: UTF-8
Content-Length: 232332
```

בדוגמא זו ניתן לראות בתשובה שמגיעה מהשרת פרטים נוספים (headers) אשר מספקים את הפרטים הבאים: התשובה בפרוטוקול HTTP 1.1, הקובץ הנדרש נמצא בהצלחה (סטטוס קוד 200), השרת ששולח את התשובה הוא מסוג Apache, סוג הקובץ שנשלח חזרה הוא טקסט (וליתר דיוק: HTML), הטקסט ב-encoding מסוג UTF-8 (פרט חשוב עבור הדפדפן כדי שידע כיצד להציג את הטקסט שהגיע) ואורך התשובה כולה הוא 232332 בתים.

כל HTTP Header הוא צירוף פשוט של key עם value בפורמט טקסטואלי פשוט. כל HTTP Header נשלח בשורה נפרדת. סימן של שורה חדשה מפריד בין כל אחד מה-headers לאחרים וסימן של שורה חדשה מפריד גם בין ה-headers והתוכן שמגיע בחזרה.

שרת ה-web ומנוע ה-PHP מטפלים באופן אוטומאטי בשליחת ה-headers.

© 2009 Haim Michael. All Rights Reserved.

הפונקציה header

פונקציה זו מאפשרת שליחה בחזרה לדפדן שורת HTTP Header נוספת שאנחנו יוצרים. בהפעלתה ניתן לשלוח ארגומנט אחד אשר כולל גם את שם ה-header וגם את ערכו.

דוגמא:

```
header("Content-type: text/html");
```

בשורה זו אנו גורמים להתווספות של שורת header נוסף אשר מעיד על כך שה-content type של תשובת השרת היא text/html.

דוגמא:

```
header("Location: http://www.jacado.com");
```

בשורה זו אנו גורמים להתווספות של שורת header נוסף אשר מיידע את הדפדפן בכך שהמסמך המבוקש נמצא בכתובת URL אחרת ובכך גורם לדפדפן לפנות אל אותה כתובת URL.

דוגמא:

```
header("HTTP/1.0 404 Not Found");
```

בשורה זו אנו גורמים להתווספות של שורת header נוסף אשר מיידע את הדפדפן בסטטוס קוד של תשובת השרת לפניה שקיבל.

כאשר קוראים להפעלת הפונקציה header יש לעשות זאת לפני שמבצעים פעולות output כלשהן... לרבות ריווחים לבנים מחוץ לגבולות תגיות ה-PHP. הימנעות מכלל זה עלולה לגרום לשגיאה בצד השרת במהלך הניסיון לסיים בהצלחה את הקריאה להפעלת הפונקציה headers.

הדוגמא הבאה מציגה שימוש בסיסי בפונקציה header.

```
<?php
header("Content-Type: text/xhtml");
header("Expires: Mon, 22 Jul 2047 05:00:00 GMT");
echo "If you want to see the headers try to browse this page via one of the
following web based sniffers:";
echo "<BR>http://webtools.mozilla.org/web-sniffer/";
echo "<BR>http://www.delorie.com/web/headers.html";
echo "<BR>http://www.rexswain.com/httpview.html";
?>
```

© 2009 Haim Michael. All Rights Reserved.

שליחת מידע דחוס

פרוטוקול HTTP תומך בשליחת מידע דחוס (באמצעות ה- gzip אלגוריתם). דרגת הדחיסות האפשרית נעה בין 1 ו-9. ברירת המחדל היא 6. כאשר שרת מחזיר לדפדפן תשובה בפורמט gzip ה-Content-Encoding מוחזר עם הערך "gzip".

דוגמא:

תשובה של שרת בפורמט דחוס (gzip) עשוייה להיראות כדלקמן.

```
HTTP/1.1 200 OK
Server: Apache
Content-Type: text/html
Content-Encoding: gzip
Content-Length: 26434
```

ומייד לאחר `headers` אלה יופיע המידע הדחוס בפורמט `gzip`.

באמצעות הפעלת הפונקציה ob_start נוכל לגרום להפעלת ה-output buffering. אפשר לקרוא להפעלתה מבלי לשלוח ארגומנטים. אחד הארגומנטים שניתן לשלוח לפונקציה זו בעת הפעלתה הוא הגודל של הבאפר שיהיה בפועל. ארגומנט אחר שניתן לשלוח הוא שמה של פונקציה שתופעל בכל עת שהבאפר מתמלא. זו תהיה פונקציה שתקבל כארגומנט את הנתונים שעומדים להישלח חזרה ללקוח ובתגובה תחזיר את התשובה שתוחזר בפועל.

הפונקציה ob_start מחזירה true אם הצליחה בפעולתה ו-false אם נכשלה.

כדי לגרום לשליחת התשובה חזרה בפורמט דחוס (gzip) יש לקרוא להפעלת הפונקציה ob_start ולשלוח אליה את שם הפונקציה ob_gzhandler.

דוגמא:

```
ob_start(“ob_gzhandler”);
```

שורה זו גורמת לקריאה להפעלת הפונקציה ob_start באופן שבכל עת שהבאפר מתמלא תוכנו יישלח אל הפונקציה ob_gzhandler והערך שהיא מחזירה הוא זה שיוחזר חזרה לדפדפן.

© 2009 Haim Michael. All Rights Reserved.

דוגמא:

```
<?php
ob_start("ob_gzhandler");
?>
<html><body><p>
If your browser supports receiving compressed data then this page was received in a
compressed.</p></body></html>
```

ניתן לשלוט באופן שבו ה-PHP engine מאפשר שליחת נתונים בצורה דחוסה באמצעות עידכון הפרמטרים המתאימים בקובץ php.ini.

דוגמא:

```
zlib.output.compression = on
zlib.output.compression.level = 9
```

© 2009 Haim Michael. All Rights Reserved.

ה-Browser Caching

רוב הדפדפנים משתמשים ב-cache memory כדי לשפר את ביצועיהם. בכל עת שהדפדפן מצפה לקבל קובץ מסויים הוא בודק תחילה אם הקובץ זמין ב-cache memory ואם הקובץ בשרת איננו עדכני יותר. במידה שהקובץ ששמור ב-cach memory עדכני הדפדפן משתמש בו.

באמצעות ה-Cache Control וה-Expires, שני headers שניתן לשלוט בערכם בתשובה שמוחזרת לדפדפן ניתן להורות לדפדפן לשמור (או לא לשמור) את התשובה שנשלחת בזיכרון ה-cache memory ובמידה שמורים לדפדפן לשמור את התשובה אז ניתן לקבוע עבורו את תאריך ה-expires של התשובה המוחזרת.

דוגמא:

```
header("Cache-Control: no-cache, must-revalidate");
```

שורה זו תיגרום לכך שהתשובה שמוחזרת לדפדפן לא תישמר בזיכרון ה-cache memory.

דוגמא:

```
header("Expires: Mon, 31 Jan 2008 02:55:22 GMT");
```

שורה זו תגרום לכך שהתשובה שמוחזרת לדפדפן תישמר ב cache memory בתוספת תאריך תפוגה. בכך יובטח שהקובץ שיישמר ב cache memory לא ישמש את הדפדפן במידה שתאריך התפוגה כבר פג.

© 2009 Haim Michael. All Rights Reserved.

עבודה עם cookies

ה-cookie היא פיסת מידע טקסטואלי מצומצם שהדפדפן קיבל משרת מסויים. לכל cookie יש, בין היתר, שם וערך. כדי לגרום לדפדפן לשמור cookie במחשב שעליו הוא רץ יש לשלוח אליו בתשובה שמגיעה מהשרת את ה-header המתאים שיגרום להיווצרות ה- cookie.

בכל עת שהדפדפן פונה שוב לשרת שבעבר גרם להיווצרות של עוגיה אצלו הדפדפן כולל בפנייתו לשרת את אותה עוגיה (בהנחה שהעוגיה עדיין בחיים (טרם פגה).

כדי לגרום לכך שהתשובה שחוזרת לדפדפן ממסמך ה-PHP תיכלול את ה-header המתאים (כדי לגרום להיווצרות עוגיה) יש לקרוא להפעלת הפונקציה setcookie.

בדומה לאופן העבודה עם פונקציות אחרות שמשפיעות על ה-headers גם בהפעלת פונקציה זו יש לוודא שאנו מפעילים אותה לפני שנכתב מידע כלשהו חזרה לדפדפן.

בעת הקריאה להפעלת setcookie קיימים ארגומנטים רבים שניתן לשלוח. ארגומנט אחד חייבים תמיד לשלוח וזהו השם של ה-cookie.

דוגמא:

```
setcookie("id",1232323);
```

שורה זו גורמת להיווצרות עוגיה חדשה ששמה `id` והערך שהיא מחזיקה הוא 1232323.

כיוון שלא נשלח לפונקציה ארגומנט שמתייחס למשך חייה היא תהיה בחיים כל עוד ה-`session` בחיים. כאשר ה-`session` מסיים את חייו גם העוגיה מסיימת את חייה. כדי לגרום לכך שהעוגיה שנוצרת תמשיך בחייה גם לאחר מות ה-`session` יש לשלוח אל הפונקציה `setcookie` גם את ה-`expiration date`, ערך מספרי אשר מתאר באלפיות השניה את המועד שבו העוגיה תסיים את חייה.

דוגמא:

```
setcookie("id","1000232",time()+86400*3);
```

שורה זו תגרום לדפדפן להחזיק את העוגיה בחיים לפרק זמן של 3 ימים. הדפדפן יעשה זאת באמצעות שמירת העוגיה כקובץ טקסטואלי קטן על המחשב.

© 2009 Haim Michael. All Rights Reserved.

בכל עת שמתרחשת פנייה של הדפדפן לאתר שבעבר יצר אצל הדפדפן cookie ובמידה שה-cookie עדיין בחיים אז ה-cookie יישלח יחד עם הפנייה לשרת.

ה-PHP Engine מזהה באופן אוטומטי את כל ה-cookies שמגיעים יחד עם ה-request שמגיע מהדפדפן, מפריד אותם מה-request ושם אותם במערך גלובלי בשם $_COOKIE. ה-keys הם שמות העוגיות וה-values הם הערכים שלהם. במידה שאנו מצפים לעוגיה מסויימת ואנו יודעים את שמה ניתן לקבל את ערכה באופן מיידי באמצעות פניה למערך הגלובלי.

דוגמא:

```
$id_val = $_COOKIE['id'];
```

שורת קוד זו מחלצת למעשה את ערכה של העוגיה id ומאחסנת אותו בתוך המשתנה id_val.

ניתן לקבוע את ערכה של עוגיה חדשה שיוצרים תוך שימוש בתחביר המשמש לעבודה עם מערכים ובדרך זו לגרום להיווצרותה של עוגיה שבמקום value אחד יש לה אוסף של values שהגישה אליהם נעשית בדומה לאופן שבו ניגשים לאיברי מערך.

דוגמא:

```
setcookie("names[0]","moshe");
setcookie("names[1]","david");
setcookie("names[2]","john");
```

שורות קוד אלה גורמות להיווצרות cookie ששמו names ושבמקום ערך אחד יש לו אוסף של ערכים. חילוץ הערכים הללו בכל ביקור חוזר של הגולש ייעשה באופן דומה.

```
$vec = $_COOKIE['names'];
echo $vec[0]; // printing "moshe"
echo $vec[1]; // printing "david"
echo $vec[2]; // printing "john"
```

אין כל דרך להורות לדפדפן למחוק עוגיה מסויימת. עם זאת, ניתן באופן עקיף להשיג תוצאה דומה. קיימות שתי דרכים. דרך אחת באמצעות הפעלת הפונקציה setcookie וקביעתה של עוגיה בשם זהה עם ערך שלילי כמועד ה-expiration שלה. פעולה זו תגרום לכך שהעוגיה תסיים את חייה מייד עם תום ה-session. דרך שניה כרוכה גם היא בהפעלת הפונקציה setcookie והיא כוללת את השמת הערך false בתור ה-value של ה-cookie.

© 2009 Haim Michael. All Rights Reserved.