COMPUTERWORLD

Komunikation på tværs af servere med PHP

Hvordan kan man kommunikerer internt med andre servere via PHP?
Jeg har opstillet denne funktion der gør det nemt i et php script at komunikerer med andre scripts på andre servere. Ud over at kunne sende data til andre scripts kan man også modtage og rea

Skrevet den **12. Feb 2009** af **hkb-x** I kategorien **Programmering / PHP** | ⭐⭐⭐☆☆

Hvordan kan man kommunikerer internt med andre servere via PHP?
Jeg har opstillet denne funktion der gør det nemt i et php script at komunikerer med andre scripts på andre servere. Ud over at kunne sende data til andre scripts kan man også modtage og reagere på data.

For at gøre det så simpelt som muligt at bruge i praksis har jeg lavet det som en funktion ved navn http_post(). Først får du hele funktionen og så vil jeg gennemgå den i mindre bider.

```
function http_post($host, $query){

    $path = explode("/", $host);
    $host = $path[0];
    unset($path[0]);
    $path = "/".(implode("/", $path));

    $post  = "POST $path HTTP/1.0\r\n";
    $post .= "Host: $host\r\n";
    $post .= "Content-type: application/x-www-form-urlencoded\r\n";
    $post .= "User-Agent: PHP Script by www.HKB.it\r\n";
    $post .= "Content-length: ".strlen($query)."\r\n";
    $post .= "Connection: close\r\n\r\n";
    $post .= "$query";

    $sock = fsockopen($host, 80);

    fwrite($sock, $post);


    for($i=0;!$i;) {

       $q = fread($sock, 8192);
       $r .= $q;
       $i = ($q=='') ? 1 : 0;

     }

     fclose($sock);

    $c = explode("\r\n\r\n", $r);

    $content[head] = $c[0];
    $content[body] = $c[1];
```

```
    return $content;
}
```

I Brug
Funktionen fungerer ved at man skriver den totale sti til det script man gerne vil poste til samt de værdier men vil poste til scriptet. De værdier skrives lige som hvis man vil skrive værdier til $_GET i en URL (dvs. hver værdi adskilles med & og værdiens indhold sættes med =).

Så den kommer til at se således ud i praksis:

```
http_post("www.minside.dk/remote_post.php", "string=tekst&return=true");
```

Dette vil poste værdierne string og return til scriptet remote_post.php på www.minside.dk/.

Bemærk at vi ikke angiver hvilken protokol vi bruger (http) da dette er indbygget i scriptet at vi bruger http 1.0. Grunden til at der bruges http 1.0 frem for http 1.1 er at http 1.1 efterlaver noget "slamkode" omkring vores body og så har det vist sig at http 1.0 i mange tilfælde er væsenligt hurtigere (op til 40%) når man bruger php (Dette er dog ikke noget jeg personligt har testet endnu).

Gennemgang af funktionen
Men lad os gennemgå funktionen i små bider af gangen.

Send Data
For at gøre det simpelt har jeg samlet hostnavnet og stien i et. Men for at kunne poste skal de være delt i sidens head. Dette gøres således:

```
function http_post($host, $query){

    $path = explode("/", $host);
    $host = $path[0];
    unset($path[0]);
    $path = "/".(implode("/", $path));
```

Først opdeler vi vore indkommende værdi $host ved hvert / tegn via explode(). Den første værdi vil så være vores host. De efterfølgende vil være stien til vores script vi skal kontakte. For at genskabe stien bruger vi implode().

Bemærk at vi bruger unset til at fjerne vores host inden vi genskaber stien.

```
    $post  = "POST $path HTTP/1.0\r\n";
    $post .= "Host: $host\r\n";
    $post .= "Content-type: application/x-www-form-urlencoded\r\n";
    $post .= "User-Agent: PHP Script by www.HKB.it\r\n";
    $post .= "Content-length: ".strlen($query)."\r\n";
```

Så skaber vi den header vi vil sende. Den indeholder først informationer om at vi vil poste via http 1.0 og hvilket script vi poster til.
Derefter angives vores host samt indholdstypen af det vi sender. application/x-www-form-urlencoded viser at vi poster som om det kom fra en <form>.

Så angiver vi hvem vi er (User-Agent). Her fortæller vi at vi er et PHP script samt hvem der har lavet det (reklame er godt ;)).
Til sidst angiver vi længden af de data vi poster.

```
$post .= "Connection: close\r\n\r\n";
$post .= "$query";
```

Så angiver vi at vi (når dataen er sendt) lukker forbindelsen igen. Og så kommer det vigtige, efter at have lavet alle vores headere viser vi at vi nu sender vores inhold. Det gør vi ved at lave 2 linieskift \r\n\r\n (det tog mig en krig at finde ud af at dette skiller headere fra indhold da jeg arbejdede på mit webmail system.).
Og så til sidst sender vi de værdier der skal postes til scriptet.

```
$sock = fsockopen($host, 80);

fwrite($sock, $post);
```

Med fsockopen() opretter vi forbindelse til den anden server (www.minside.dk). Bemærk at dette gøres via port 80 der er standartporten når man ser en hjemmeside.

Med fwrite() sender vi så vores data ($post) via den forbindelse vi har oprettet ($sock).

Modtag Data
Nu har vi så sendt alle vores data og vi skal nu have svar fra www.minside.dk.

```
for($i=0;!$i;) {

   $q = fread($sock, 8192);
   $r .= $q;
   $i = ($q=='') ? 1 : 0;

 }

 fclose($sock);
```

Via fread() henter vi det retunerede data fra vores forbindelse ($sock) i bider af 8MB eller rettere 8192 byte.

For at kunne konstatere om alt indhold er hentet bruger jeg en lidt kryptisk metode.

Så længe $i er lig 0 (kan også skrives som !$i) skal den hente data fra vores forbindelse.

De data der hentes gemmes som $q, $q gemmes her efter som $r. Så undersøges der om $q er tom. Er $q det så sættes $i lig 1 og det hentes ikke mere data ellers er $i lig 0 og vi fortsætter med at hente data.

Når vi så er færdige med at hente data ligger alt vi har fået retuneret i $r. Så lukker vi vore forbindelse med fclose().

Håndter retuneret data.
Så mangler vi bare at få sorteret det vi har fået sendt tilbage. Det gør vi ved at vi ved at \r\n\r\n adskiller headers og body.

```
    $c = explode("\r\n\r\n", $r);

    $content[head] = $c[0];
    $content[body] = $c[1];

    return $content;
```

Så gemmer vi headere og data i et array som vi så retunerer fra funktionen. Og volá, det var det.

Eksempel på praktisk brug.
Ndedenfor har jeg listet 2 scripts; kontakt.php og svar.php. I dette eksempel kontaker kontakt.php svar.php og registrerer det svar man har fået.

kontakt.php

```
$svar = post("www.minside.dk/remote_conection/svar.php",
"valider=".md5("noget_hemmeligt")."reply=true");

$værdier = explode("&", $svar['body']);

for($i=0;$i<count($værdier);$i++) {

    $d = explode("=", $værdier[$i]);

    $return[$d[0]] = $d[1];

}

if($return['kontrol'] == md5("kontrol_værdi")) {

    echo $return['svar'];

}
```

svar.php

```
if($_POST['valider'] == md5("noget_hemmeligt")) {

    //PHP udfører en eller anden komando og laver et $svar

    if($_POST['reply'] == "true") {

        echo "kontrol=".md5("kontrol_værdi")."svar=".$svar;

    }
}
```

Det er sker når man aktiverer kontakt.php er:

kontakt.php kontakter svar.php for at få den til at udføre noget. For at svar.php er sikker på at det er kontakt.php der aktiverer det så sendes en md5 værdi med. Der ud over sendes en besked om at kontakt.php gerne vil have et svar tilbage.

svar.php undersøger først og md5 værdien er korrekt, er den det vil den udfører en handling. Hvis der har

fået af vide at den skal sende et svar tilbage så gør den det. svar.php laver dog også en værdi kontakt.php skal validerer.

kontakt.php behandler svaret hvor den deler det svar den har fået fra svar.php op i værdier der gemmes i $return. Hvis md5 værdien kan godkendes skriver den det svar svar.php har sendt til brugeren.

Yderligere Kommentarer
Det var helt basalt lidt om hvordan man poster på tværs af servere. Det er desværrer ikke noget man får brugt særligt tit men meget spændende at lege med.

Hvis man vil sende noget over en ssl krypteret forbindelse skal man skrive ssl:// før hostnavnet (ssl://www.minside.dk).

Svar
Da det desværrer ikke er muligt for mig at svare på folks kommentarer til denne artikkel vil jeg placere dem her.

heine112 Jeg ved godt man burde bruge http 1.1 men da http 1.0 er renere og hurtigere vælger jeg at bruge den. Jeg gør folk opmærksomme på det så de kan ændre det til http 1.1 hvis de ønsker. Jeg vælger dog personligt http 1.0.

Kommentar af htx98i17 d. 22. Jun 2006 | 1

God artiklen. Eksempel på brug: remote-capture af dankorttransaktioner hos dandomain, over SSL...

Kommentar af ceec d. 08. Jun 2006 | 2

God artikel, glæder mig til at prøve det af.

Kommentar af per-d d. 24. Jul 2006 | 3

En god artikel der går ind i et område hvor mange hjemmeudviklere somregel står af, men hvor du alligevel for opsamlet de forskellige ting fint. Måske du skulle nævne andre måder at gøre det på for eksempel via SOAP.

Kommentar af mathiash d. 18. Jun 2006 | 4

God artikel, men jeg kunne godt tænkte mig at vide mere om hvor det for eksempel kunne være nyttigt.

Kommentar af heine112 d. 29. Jun 2006 | 5

I http-protokollen kan der være mange headers og bodies. Ikke blot 1 header og 1 body, som antages.

I øvrigt burde du anvende http 1.1 - der var noget snavs ved 1.1 skriver du, men det må du jo blot gøre noget ved.

Kommentar af kamiga d. 08. Jun 2006 | 6

God artikel, klart de 5 point værd :-D

Kommentar af webudvikleren d. 08. Jun 2006 | 7

Rigtig fed artikel! Bestemt 5 points værd - efter min mening!

Kommentar af mcmini d. 08. Jun 2006 | 8

Man får mere end man betaler for !

Kommentar af reinhardt d. 21. Aug 2006 | 9

Kommentar af kristianiversen d. 02. Jun 2007 | 10

God artikel! Jeg fik rigtigt meget ud af den.