

Tutorial - Håndtering af FTP i PHP del 1

Skrevet d. 19/06, 2008 kl 19:36 af Martin Visninger: 234 [Læs/skriv kommentar \(3\)](#)

Dette er en introduktion til [phps FTP funktioner](#), jeg vil indledningsvist gennemgå hvordan du forbinder til en FTP server, henter en liste over filer i en mappe, uploader en fil og selvfølgelig hvordan du henter en fil.

Først og fremmest skal vi forbinde til en FTP server, og herefter skal vi logge ind. Dette gøres ved at bruge funktionerne [ftp_connect\(\)](#) og [ftp_login\(\)](#) som det ses herunder:

```
<?php
$ftp= @ftp_connect('side') or die('Kunne ikke skabe forbindelse'); //skab
forbindelse
if (!@ftp_login($ftp, 'brugernavn','password')) { //hvis vi ikke kan logge ind
    die('Kunne ikke logge ind - forkert brugernavn eller password');
}
?>
```

Ovenstående kode skaber først en forbindelse til FTP serveren, eller stopper udførelsen af scriptet hvis forbindelsen ikke kunne åbnes. Herefter forsøger vi at logge ind med [ftp_login\(\)](#), det ikke lykkedes stopper vi igen udførelsen af scriptet. Hvis koden ikke outputter noget er der skabt forbindelse til ftp serveren, og herefter kan vi udfører vores kommandoer.

Hent fil liste fra mappe

Når du ønsker at hente en liste over indholdet i en mappe, er der to funktioner der kan bruges, enten [ftp_nlist\(\)](#) eller [ftp_rawlist\(\)](#). Førstnævnte returnerer et array med filnavnene, mens sidstnævnte returnerer et array med fuld ftp info omkring filen. Jeg foretrækker næsten altid nlist. For at hente listen over filer i rodmappen bruger vi ftp_nlist på følgende måde:

```
<?php
$indhold= ftp_nlist($ftp, "."); //hent liste over indhold i den nuværende mappe
foreach ($indhold as $fil) { //gennemløb arrayet
    echo $fil.'<br />'; //udskriv indholdet
}
?>
```

Upload af fil fra HTTP server til FTP server

Upload er lige så let som at hente en fil oversigt, her bruger vi bare funktionen [ftp_put\(\)](#) funktionen skal vidde hvilken fil den skal uploade og hvor til og om det skal være i Binary eller Ascii:

```
<?php
ftp_put($ftp, '/lokal/sti/server/fil.ext', '/ftp/sti/fil.ext', FTP_BINARY);
?>
```

Om du skal bruge FTP_BINARY eller FTP_ASCII kommer an på hvilken type fil du uploader. Som hovedregel, hvis du uploader andet end tekst skal du bruge FTP_BINARY, men hvis du fx uploader en HTML fil med JavaScript i, kan det også være nødvendigt at bruge FTP_BINARY.

BEMÆRK: Du kan ikke angive en fil på din lokale computer der skal uploades.

Hentning af fil fra FTP server til HTTP server

Hentning af filer sker ved at bruge [ftp_get\(\)](#) - som har næsten samme opbygning som [ftp_put\(\)](#):

```
<?php
$fil= ftp_get($ftp, '/lokal/sti/server/fil.ext', '/ftp/sti/fil.ext', FTP_BINARY
);
?>
```

Når du er færdig med at bruge ftp forbindelsen bør du kalde ftp_close(\$ftp) for at lukke denne.

Afsluttende bemærkninger

Dette var en kort gennemgang af de mest basale FTP funktioner, og jeg håber det gav mening.

Hvis du sidder og tænker "hvad kan jeg bruge dette til", så er der egentlig massere. Det kunne være man kørte en FTP server og ønskede at gøre nogle af disse filer tilgængelige via http også. Eller hvis du laver administrationssystemer som skal kunne uploade til flere forskellige servere.. osv osv. Jeg vil i den nærmeste fremtid skrive del 2 af denne tutorial som vil gennemgå lidt flere muligheder