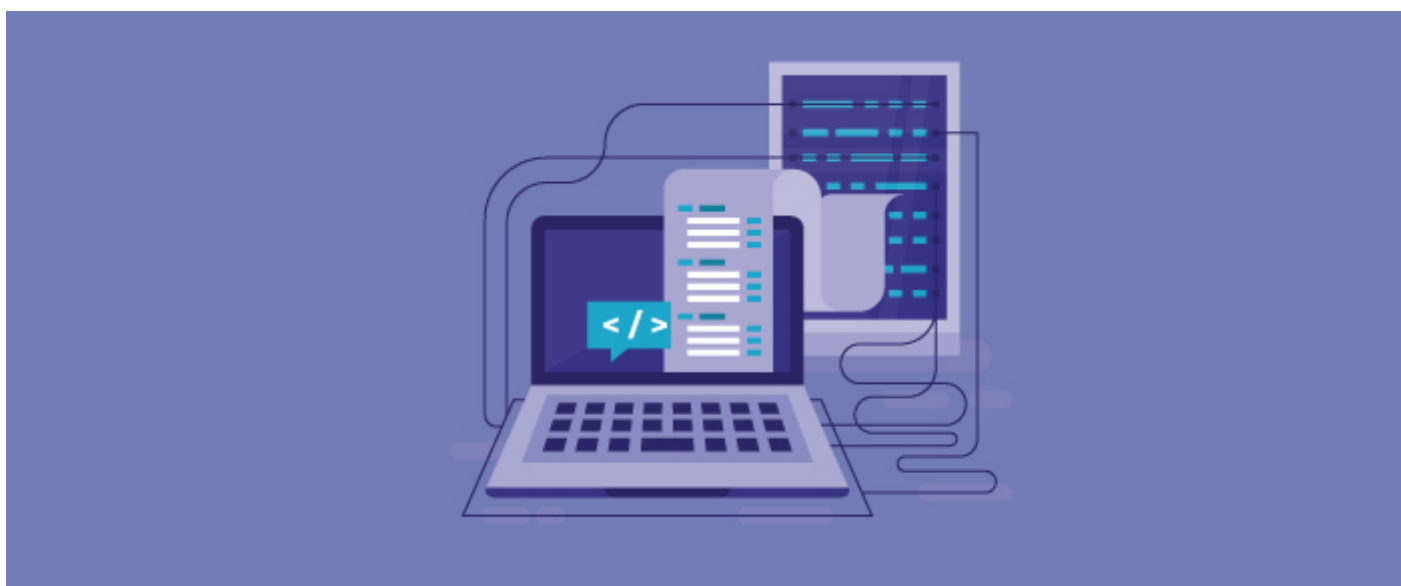


## Lavorare su FTP remoto con PHP

*lunedì, 18 Marzo 2019*

Sapevi che è possibile aprire una connessione FTP utilizzando il linguaggio PHP? Certo che è possibile ed è una pratica davvero molto utile per lavorare file o cartelle presenti su server remoti. Grazie alle funzioni presenti di default nella libreria PHP, puoi gestire in pieno controllo files e cartelle presenti su un server remoto di cui hai un account FTP.

Ti interessa sapere come fare?



### Sommario

I parametri necessari

Connessione al server

Login al server

Download di un file dal server remoto FTP

Upload di un file sul server remoto FTP

Rinominare un file dal server remoto FTP

Altre funzioni utili

## I parametri necessari

Ovviamente, la connessione FTP ad un server remoto è possibile solamente se si dispone delle credenziali di accesso allo stesso.

Ti occorrono:

- **HOST**: l'indirizzo IP oppure il nome host a cui connettersi  
*Esempio: ftp://nomedelsito.com*
- **USER**: il nome utente dell'account FTP con cui si desidera connettersi
- **PASSWORD**: la password di accesso dell'account con cui si desidera connettersi

Iniziamo quindi proprio con il dichiarare le variabili necessarie da utilizzare per la connessione:

```
<?php  
  
// Parametri di accesso al server remoto tramite FTP  
$ftp_host = "ftp://nomedelsito.com";  
$ftp_username = "nome_utente";  
$ftp_password = "password_di_accesso";
```

Il primo passo è compiuto, abbiamo preparato il terreno per effettuare la tua prima connessione FTP tramite PHP, ora hai la possibilità di connetterti al server FTP utilizzando due funzioni già incluse nella libreria PHP: **ftp\_connect()** e **ftp\_login()**.

## Connessione al server

La prima funzione, **ftp\_connect()**, si occupa di aprire una comunicazione con un server remoto ed accetta tre parametri, di cui il primo è obbligatorio:

- **\$host**: l'indirizzo del server verso cui aprire la connessione;
- **\$port**: la porta di comunicazione del server attraverso cui aprire la connessione. Se non specificato, la funzione utilizzerà la porta di default per FTP, ovvero 21;
- **\$timeout**: il tempo massimo di attesa senza operazioni entro cui la connessione viene mantenuta aperta. Se non specificato, il timeout verrà impostato a 90 secondi;

La funzione **ftp\_connect()** restituisce uno stream FTP in caso di successo, oppure il valore booleano `false` in caso di errore di connessione.

La teoria è sempre noiosa, lo so, ma è necessario avere le basi prima di provare! Passiamo ora alla pratica; riprendendo quanto già visto nel primo paragrafo circa le variabili necessarie, avrai:

```
<?php

// Parametri di accesso al server remoto tramite FTP
$ftp_host = "ftp://nomedelsito.com";
$ftp_username = "nome_utente";
$ftp_password = "password_di_accesso";

// Stabilisce la connessione al server remoto tramite FTP
$ftp = ftp_connect( $ftp_host );

// Controlla se la connessione Ã¨ andata a buon fine
if ( ! $ftp ) {
    echo "Connessione FTP fallita.";
    exit;
}

// Chiude la connessione FTP
ftp_close( $ftp_stream );
```

In questo modo, lo script tenta di aprire una connessione FTP verso il server host ed in caso di errore, termina la sua esecuzione visualizzando un avviso di connessione fallita.

Per approfondire il funzionamento della funzione **ftp\_connect()**, dai un'occhiata qui:

[La funzione ftp\\_connect\(\)](#)

## Login al server

Se la connessione va a buon fine, vuol dire che il server host ci ha risposto ed è disponibile a dialogare con lo script, ma per farlo è necessario "farsi riconoscere". La seconda funzione menzionata in precedenza, **ftp\_login()**, si occupa appunto di effettuare l'accesso al server, tramite la coppia username/password.

La funzione **ftp\_login()**, accetta tre parametri, tutti obbligatori:

- **\$ftp\_stream**: la risorsa FTP che identifica la connessione al server FTP, ottenuta in precedenza dalla funzione **ftp\_connect()**;
- **\$username**: il nome utente dell'account con cui si desidera accedere;
- **\$password**: la password di accesso relativa all'account con cui si desidera accedere;

La funzione **ftp\_login()** restituisce i valori booleani *true* oppure *false* a seconda se l'accesso va a buon fine o fallisce.

Aggiungiamo un altro mattoncino allo script precedente, ottenendo:

```
<?php

// Parametri di accesso al server remoto tramite FTP
$ftp_host = "ftp://nomedelsito.com";
$ftp_username = "nome_utente";
$ftp_password = "password_di_accesso";

// Stabilisce la connessione al server remoto tramite FTP
$ftp = ftp_connect( $ftp_host );

// Controlla se la connessione Ã" andata a buon fine
if ( ! $ftp ) {
    echo "Connessione FTP fallita.";
    exit;
}

// Effettua il login al server remoto tramite FTP
$ftp_login = @ftp_login( $ftp_stream, $ftp_username, $ftp_password );

// Controlla se il tentativo di login Ã" andato a buon fine
if ( ! $ftp_login ) {
    echo "Login FTP fallito.";
    exit;
}

// Chiude la connessione FTP
ftp_close( $ftp_stream );
```

Hai notato il simbolo "@" alla riga 18, prima della chiamata alla funzione **ftp\_login()**? A cosa serve? Il simbolo "@" prima della chiamata ad una funzione in PHP è utilizzato per "silenziare" gli eventuali errori minori (notices/warnings) generati dalla funzione. In questo caso, visto che la funzione **ftp\_login()** restituisce un warning in caso di insuccesso nel login, con il simbolo "@" previeni la visualizzazione del messaggio di warning, molto molto fastidiosa.

Per approfondire il funzionamento della funzione **ftp\_login()**, puoi leggere:

[La funzione ftp\\_login\(\)](#)

Bene, ora sei ufficialmente connesso al server FTP remoto! Ma cosa è possibile fare adesso?

## Download di un file dal server remoto FTP

Una delle operazioni più utilizzate è sicuramente il download di un file tramite FTP. La funzione da utilizzare è **ftp\_get()**, che accetta cinque parametri, di cui i primi tre obbligatori:

- **\$ftp\_stream**: la risorsa FTP che identifica la connessione al server FTP, ottenuta in precedenza dalla funzione **ftp\_connect()**;
- **\$local\_file**: il percorso locale del file da prelevare;
- **\$remote\_file**: il percorso sul server remoto in cui è presente il file, completo del nome del file;
- **\$mode**: la modalità di trasferimento del file. Può essere **FTP\_ASCII** o **FTP\_BINARY**;
- **\$resume\_pos**: la posizione in cui iniziare il download del file;

La funzione **ftp\_get()** restituisce i valori booleani *true* oppure *false* a seconda se il download va a buon fine o fallisce.

Nel caso tu voglia effettuare il download di un file da server remoto FTP, lo script diventa:

```
<?php

// Parametri di accesso al server remoto tramite FTP
$ftp_host = "ftp://nomedelsito.com";
$ftp_username = "nome_utente";
$ftp_password = "password_di_accesso";

// Stabilisce la connessione al server remoto tramite FTP
$ftp = ftp_connect( $ftp_host );

// Controlla se la connessione Ã" andata a buon fine
if ( ! $ftp ) {
    echo "Connessione FTP fallita.";
    exit;
}

// Effettua il login al server remoto tramite FTP
$ftp_login = @ftp_login( $ftp_stream, $ftp_username, $ftp_password );

// Controlla se il tentativo di login Ã" andato a buon fine
if ( ! $ftp_login ) {
    echo "Login FTP fallito.";
    exit;
}

// Imposta le variabili relative al file da gestire
$remote_file = "remote_test.txt";
```

```
$local_file = "downloads/local_test.txt";

// Tenta di prelevare il file da server remoto FTP
$ftp_get_response = ftp_get( $ftp_stream, $local_file, $remote_file );

// Controlla se il tentativo di download Ã¨ andato a buon fine
if ( $ftp_get_response ) {
    echo "Download completato con successo";
} else {
    echo "Si Ã¨ verificato un errore nel download del file";
}

// Chiude la connessione FTP
ftp_close( $ftp_stream );
```

In questo esempio, abbiamo effettuato il download del file *remote\_test.txt* presente nella root del server remoto nella cartella *downloads/* del server locale, rinominandolo in *local\_test.txt*.

Semplice no?

Per approfondire il funzionamento della funzione **ftp\_get()**, puoi leggere:

[La funzione ftp\\_get\(\)](#)

## Upload di un file sul server remoto FTP

Per trasmettere un file da un server ad un altro tramite FTP, la funzione da impiegare è **ftp\_put()**, che, analogamente ad **ftp\_get()**, utilizza cinque parametri, di cui i primi tre obbligatori:

- **\$ftp\_stream**: la risorsa FTP che identifica la connessione al server FTP, ottenuta in precedenza dalla funzione **ftp\_connect()**;
- **\$remote\_file**: il percorso sul server remoto in cui posizionare il file, completo del nome del file;
- **\$local\_file**: il percorso locale del file da uploadare;
- **\$mode**: la modalità di trasferimento del file. Può essere **FTP\_ASCII** o **FTP\_BINARY**;
- **\$startpos**: la posizione in cui iniziare l'upload del file;

La funzione **ftp\_put()** restituisce i valori booleani *true* oppure *false* a seconda se l'upload va a buon fine o fallisce.

Nel caso tu voglia effettuare l'upload di un file su server remoto FTP, lo script diventa:

```
<?php

// Parametri di accesso al server remoto tramite FTP
$ftp_host = "ftp://nomedelsito.com";
$ftp_username = "nome_utente";
$ftp_password = "password_di_accesso";

// Stabilisce la connessione al server remoto tramite FTP
$ftp = ftp_connect( $ftp_host );

// Controlla se la connessione Ã" andata a buon fine
if ( ! $ftp ) {
    echo "Connessione FTP fallita.";
    exit;
}

// Effettua il login al server remoto tramite FTP
$ftp_login = @ftp_login( $ftp_stream, $ftp_username, $ftp_password );

// Controlla se il tentativo di login Ã" andato a buon fine
if ( ! $ftp_login ) {
    echo "Login FTP fallito.";
    exit;
}

// Imposta le variabili relative al file da gestire
$remote_file = "uploads/remote_test.txt";
```



```
$local_file = "local_test.txt";

// Tenta di caricare il file su server remoto FTP
$ftp_put_response = ftp_put( $ftp_stream, $remote_file, $local_file );

// Controlla se il tentativo di upload Ã andato a buon fine
if ( $ftp_put_response ) {
    echo "Upload completato con successo";
} else {
    echo "Si Ã verificato un errore nell'upload del file";
}

// Chiude la connessione FTP
ftp_close( $ftp_stream );
```

In questo esempio, abbiamo effettuato l'upload del file *local\_test.txt* presente nella stessa cartella dello script nella cartella del server remoto *uploads/*, rinominandolo in *remote\_test.txt*.

Per approfondire il funzionamento della funzione **ftp\_put()**, puoi leggere:

[La funzione ftp\\_put\(\)](#)

## Rinominare un file dal server remoto FTP

Un'altra funzione utile per lavorare con FTP e PHP è sicuramente **ftp\_rename()** che permette di modificare il nome di un file presente su un server remoto. Accetta tre parametri, tutti obbligatori:

- **\$ftp\_stream**: la risorsa FTP che identifica la connessione al server FTP, ottenuta in precedenza dalla funzione **ftp\_connect()**;
- **\$oldname**: il nome del file da rinominare, completo di percorso;
- **\$newname**: il nuovo nome che si desidera dare al file;

La funzione **ftp\_rename()** restituisce i valori booleani *true* oppure *false* a seconda se la modifica del nome va a buon fine o fallisce.

Uno script di esempio per rinominare un file remoto tramite FTP potrebbe essere:

```
<?php

// Parametri di accesso al server remoto tramite FTP
$ftp_host = "ftp://nomedelsito.com";
$ftp_username = "nome_utente";
$ftp_password = "password_di_accesso";

// Stabilisce la connessione al server remoto tramite FTP
$ftp = ftp_connect( $ftp_host );

// Controlla se la connessione Ã" andata a buon fine
if ( ! $ftp ) {
    echo "Connessione FTP fallita.";
    exit;
}

// Effettua il login al server remoto tramite FTP
$ftp_login = @ftp_login( $ftp_stream, $ftp_username, $ftp_password );

// Controlla se il tentativo di login Ã" andato a buon fine
if ( ! $ftp_login ) {
    echo "Login FTP fallito.";
    exit;
}

// Imposta le variabili relative al file da gestire
$old_filename = "old_test.txt";
$new_filename = "new_test.txt";
```

```
// Tenta di rinominare il file su server remoto FTP
$ftp_rename_response = ftp_rename( $ftp_stream, $old_filename, $new_filename );

// Controlla se il tentativo di rinomina Ã" andato a buon fine
if ( $ftp_rename_response ) {
    echo "File rinominato con successo";
} else {
    echo "Si Ã" verificato un errore nel rinominare il file";
}

// Chiude la connessione FTP
ftp_close( $ftp_stream );
```

Come sempre, per approfondire l'utilizzo della funzione **ftp\_rename()**, troverai utile:

[La funzione ftp\\_rename\(\)](#)

## Altre funzioni utili

Analogamente alle funzioni più utilizzate, già descritte in questa guida, ne esistono altre altrettanto utili per poter lavorare tramite connessione FTP con PHP, eccone alcune:

- **ftp\_delete( \$ftp\_stream, \$path )**: elimina un file da server remoto;
- **ftp\_chmod( \$ftp\_stream, \$mode, \$filename )**: permette di modificare i permessi di lettura/scrittura di un file presente su server remoto;
- **ftp\_mkdir( \$ftp\_stream, \$directory )**: consente di creare una nuova cartella su server remoto tramite connessione FTP;

Per una panoramica di tutte le funzioni che puoi utilizzare per gestire risorse tramite connessione FTP in PHP, ti consiglio di leggere:

[Le funzioni FTP](#)

Cosa aspetti? Non ti resta che iniziare a buttare giù qualche riga di codice!