

FTPS

Implicit FTPS

The code below contains examples to upload, download and delete files on an implicit FTPS server and to get a directory listing:

```
<?php
//upload test
$server="servername";
$user="username";
$pass="password";
$remotefilename="/InBound/test.txt";
$localfilename="test.txt";

/*
$fp = fopen($localfilename, 'r');
$ftp_server = 'ftps://'.$server.'/'.$remotefilename;
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $ftp_server);
curl_setopt($ch, CURLOPT_USERPWD,$user.':'.$pass);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
curl_setopt($ch, CURLOPT_FTP_SSL, CURLFTPSSL_TRY);
curl_setopt($ch, CURLOPT_FTPSSLAUTH, CURLFTPAUTH_TLS);
curl_setopt($ch, CURLOPT_UPLOAD, 1);
curl_setopt($ch, CURLOPT_INFILE, $fp);

$output = curl_exec($ch);
$error_no = curl_errno($ch);
var_dump(curl_error($ch));
curl_close($ch);
*/

/*
//download test
$remotefilename="/OutBound/test.txt";
$localfilename="testdown.txt";
$fp = fopen($localfilename, 'w');
$ftp_server = 'ftps://'.$server.'/'.$remotefilename;
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $ftp_server);
curl_setopt($ch, CURLOPT_USERPWD,$user.':'.$pass);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
curl_setopt($ch, CURLOPT_FTP_SSL, CURLFTPSSL_TRY);
curl_setopt($ch, CURLOPT_FTPSSLAUTH, CURLFTPAUTH_TLS);
curl_setopt($ch, CURLOPT_UPLOAD, 0);
```

```
curl_setopt($ch, CURLOPT_FILE, $fp);

$output = curl_exec($ch);
$error_no = curl_errno($ch);
var_dump(curl_error($ch));
curl_close($ch);
*/

//Diretory Listing test
$remotepath="/OutBound/";
$ftp_server = 'ftps://'.$server.'/'.$remotepath;
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $ftp_server);
curl_setopt($ch, CURLOPT_USERPWD,$user.':'.$pass);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
curl_setopt($ch, CURLOPT_FTP_SSL, CURLFTPSSL_TRY);
curl_setopt($ch, CURLOPT_FTPSSLAUTH, CURLFTPAUTH_TLS);
curl_setopt($ch, CURLOPT_FTPLISTONLY, true);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

$output = curl_exec($ch);
//create array from directory listing
$output = preg_split("/\r\n|\n|\r/", $output, null, PREG_SPLIT_NO_EMPTY);
$error_no = curl_errno($ch);
#var_dump(curl_error($ch));
var_dump($output);
curl_close($ch);

/*
//Delete file test
$remotepath="/InBound/";
$remotefile="test2.txt";

$ftp_server = 'ftps://'.$server.'/'.$remotepath;
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $ftp_server);
curl_setopt($ch, CURLOPT_USERPWD,$user.':'.$pass);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, FALSE);
curl_setopt($ch, CURLOPT_FTP_SSL, CURLFTPSSL_TRY);
curl_setopt($ch, CURLOPT_FTPSSLAUTH, CURLFTPAUTH_TLS);
curl_setopt($ch, CURLOPT_QUOTE, array("DELE $remotepath$remotefile"));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

$output = curl_exec($ch);
$error_no = curl_errno($ch);
#var_dump(curl_error($ch));
var_dump($output);
```

```
curl_close($ch);  
*/
```

Explicit FTPS

The code below contains example of directory listing of an explicit FTPS server - the normal PHP FTP functions can be used:

```
<?php  
  
$ftp_server="server";  
$ftp_port="990";  
$ftp_user_name="username";  
$ftp_user_pass="password";  
  
// set up basic ssl connection  
$conn_id = ftp_ssl_connect($ftp_server,$ftp_port);  
  
if(!$conn_id) {  
    //could not connect  
    die("can't connect to $ftp_server on port $ftp_port\n");  
}  
  
// login with username and password  
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);  
  
if (!$login_result) {  
    // PHP will already have raised an E_WARNING level message in this case  
    die("can't login\n");  
}  
  
echo ftp_pwd($conn_id); // /  
  
// get contents of the current directory  
$contents = ftp_nlist($conn_id, ".");  
  
// output $contents  
var_dump($contents);  
  
// close the ssl connection  
ftp_close($conn_id);
```

From:
<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:
http://wuff.dyndns.org/doku.php?id=php:implicit_ftps

Last update: **2023/05/29 11:55**



