

PHP: Send PDF or other binary file with SendLetter()

```
<?php

/***** Settings Begin *****/

$filename = 'test.doc'; // file to be posted; contents need to conform to
requirements (address in proper address area)
$username = '*****';
$password = '*****';
$description = 'Sending a letter using PHP';
$mode = 'Default'; // Default, Production, or Development.

/***** Settings End *****/

// Open File
if ( !($fp = fopen($filename, "r"))){
    // Error opening file
    // Handle error however appropriate for your script
    echo "Error opening file";
    exit;
}

// Read data from the file into $data
$data = "";
while (!feof($fp)) $data .= fread($fp,1024);
fclose($fp);

$soapclient = new SoapClient('https://api.postalmethods.com/2009-02-
26/PostalWS.asmx?WSDL');
$result = $soapclient->SendLetter(array('Username'    => $username,
                                     'Password'     => $password,
                                     'MyDescription'  => $description, // free-form
description for your records
                                     'FileExtension' => end(explode(".", $filename)),
// make sure the extension reflects the file type
                                     'FileBinaryData' => $data, // PHP5 does
base64_encoding implicitly
                                     'WorkMode'     => $mode));

//print_r($result);

$status_code = $result->SendLetterResult; // $status_code is the (positive)
transaction ID if successful, or a (negative) error number if unsuccessful
if ($status_code > 0){
    print "Message submitted successfully with transaction ID
<b>$status_code</b>";
} else {
    print "Message submission failed on error <a
href=\"http://www.postalmethods.com/resources/reference/status-
codes\">$status_code</a>";
}
?>
```

PHP: Retrieve The PDF Document Of The Letter Or Postcard With GetPDF()

```

<?php

/***** Settings Begin *****/

$username = '*****';
$password = '*****';
$id = 99999999; // ID of transaction being queried
$output_filename = 'PostalMethods_' . $id . '.pdf'; // Comment out to view on screen

/***** Settings End *****/

$soapclient = new SoapClient('https://api.postalmethods.com/2009-02-26/PostalWS.asmx?WSDL');
$result = $soapclient->GetPDF(array('Username' => $username,
                                   'Password' => $password,
                                   'ID' => $id));

//print_r($result);
//exit;

// Extract returned fields
$result_code = $result->GetPDFResult->ResultCode;
$file_data = $result->GetPDFResult->FileData;

if ( $result_code == -3000 ) {
    // an letter has been successfully retrieved
    if ( $output_filename != '' ) {
        // output to file
        print "PDF successfully retrieved<br>";
        if ( !$handle = fopen($output_filename, 'w') ) {
            echo "Cannot open file ($output_filename)";
            exit;
        }
        if ( fwrite($handle, $file_data) === FALSE ) {
            echo "Cannot write to file $output_filename";
            exit;
        }
        print "PDF successfully written<br>";
        fclose($handle);
    } else {
        // output to screen
        header ("Content-type: application/pdf");
        echo $file_data;
    }
} else {
    // no letter retrieved
    echo "Error: <a href=\"http://www.postalmethods.com/resources/reference/status-codes\">$result_code</a><br>";
}

?>

```

PHP: Accept status callback by HTTP POST from PostalMethods

```

<?php

/**** Settings *****/

```

```

$myAuthValue = ''; // The verification value you entered in PostalMethod's control
panel (Control Panel -> User Settings -> Feedback)

/*****/

if(! ctype_digit($_REQUEST['ID'])){
    $pmId = 'Suspect value; not a positive integer';
} else {
    $pmId = $_REQUEST['ID'];
}

if(! ctype_digit($_REQUEST['ItemID'])){
    $pmItemId = 'Suspect value; not a positive integer';
} else {
    $pmItemId = $_REQUEST['ItemID'];
}

if($_REQUEST['Auth'] != $myAuthValue){
    $pmAuth = 'Suspected forgery; authorization value doesn\'t match';
} else {
    $pmAuth = $_REQUEST['Auth'];
}

if($_REQUEST['Status'] != strval(floatval($_REQUEST['Status']))) {
    $pmStatus = 'Suspect value; not a positive or negative integer';
} else {
    $pmStatus = $_REQUEST['Status'];
}

$arrStatusTime = date_parse($_REQUEST['StatusTime']);

if($arrStatusTime['error_count'] > 0 || $arrStatusTime['warning_count'] > 0){
    $pmStatusTime = 'Suspect value; not a valid date-time';
} else {
    $pmStatusTime = $_REQUEST['StatusTime'];
}

//
// Put your business logic here
// In this case, we're just writing the returned values to a file,
// but you will probably want to do something more intelligent
//

$fh = fopen('pm_feedback.txt', 'a+') or die("can't open pm_feedback.txt:
$php_errormsg");

fwrite($fh, date('r', time()) . "\r\n");
fwrite($fh, "=====\r\n");

fwrite($fh, "ID: $pmId \r\n");
fwrite($fh, "ItemID: $pmItemId \r\n");
fwrite($fh, "Auth: $pmAuth \r\n");
fwrite($fh, "Status: $pmStatus \r\n");
fwrite($fh, "StatusTime: $pmStatusTime \r\n");

fwrite($fh, "\r\n");
fclose($fh);

//
// End business logic here
//

```

```
echo "0"; // Tells PostalMethods that the callback has succeeded  
?>
```