



Fax via HTTP (POST)

Index:

Introduction:3

Usage:3

Introduction:

TraiTel Telecommunications offers several delivery methods for its faxing service. This document will describe the HTTP/POST delivery method. The HTTP/POST delivery method is a simple fax delivery method supporting only one recipient per fax, and only one document per fax. This API is provided to clients who wish to programmatically send faxes using a method other than email (SMTP).

The HTTP/POST delivery method is provided both under standard HTTP, and secured HTTP over SSL (HTTPS).

Usage:

HTTP Request:

The HTTP request must be a POST request using the content-type: "multipart/form-data" in order to function properly. In HTML, this is done by including the parameters *METHOD=POST* and *ENCTYPE="multipart/form-data"* in the *<FORM>* tag. The form must call TraiTel Telecommunication's fax gateway script, located at:

<http://api.traitel.com/faxgateway.pl> (unsecure), or

<https://api.traitel.com/faxgateway.pl> (secure)

There are redundant gateways for both the secure and non-secure gateway. These gateways may be reached if a connection error occurs with a primary gateway. TraiTel **strongly recommends** configuring your application to access the redundant gateways when, and only when, a connection attempt to the primary gateway fails. These gateways are located at:

Non-secure: <http://api2.traitel.com/faxgateway.pl>

Secure: <https://api2.traitel.com/faxgateway.pl>

Fields:

The following table lists the fields available in the POST method. The fields marked 'required' must be present in order for the fax to be delivered.

Required	Name	Description
Yes	"mode"	A string specifying the type of operation. Options are: "verify" - verifies the content of the request, but does not send the fax. "send" - verifies and sends the fax request.
No	"subject"	A subject line for the fax job. This subject line will not appear on any faxed document. It only appears in the emailed fax reports.
Yes	"user"	The TraiTel account username
Yes	"pass"	The TraiTel account password. This field is case sensitive.
Yes	"recipient"	A string containing the destination fax number, in international format, without any spaces or symbols. A correct value would be "612901234567".
Yes	"file"	A file object. This must be an attached file.

File types:

The TraiTel Telecommunications fax service supports several file types. The PDF format is the most recommended as it preserves formatting, and embeds related images and fonts. However, TraiTel also supports Microsoft Office documents (Word, Excel, PowerPoint...), HTML, plain text and RTF documents. Support for other document formats may be added in the future. If a fax is sent in a format that is not supported, an e-mail will be sent in reply stating that the format is not supported.

Note: You must include the content of each attached file in the HTTP POST, encoded within a MIME multipart body (multipart/form-data). Specifying only the full path of the file is not sufficient.

Specifying file type:

In some cases it may be necessary to specify what the attached files contain. (for example, if your file attachment does not have a file extension matching its contents). In these cases, you may explicitly specify which file extension should be used in its place. For this, use the follow key/value pair:

```
ext = xxx
```

For example, if the file attached is a PDF but TraiTel is reporting that the file format is unrecognized, you may include `ext=pdf` within your HTTP POST to assist TraiTel in identifying the attached file's type.

Multiple attachments:

The fax gateway allows up to 10 separate attachments to be included with each fax. As specified earlier, the first attachment must be attached using the key "file", and, if needed, the file type may be specified using "ext" as specified above.

Additional documents may be attached using the keys file2, file3, file4... up to file10; and the associated, optional file type identifiers, ext2, ext3, ext4.... up to ext10.

Documents beyond the 10th attachment will not be sent.

Fax reports via HTTP:

When creating automated processes, it is often necessary to track the status of the fax delivery, as well as whether it was successfully sent or not. These details are included in the email fax reports sent after each fax; however this is not a suitable solution for various implementations.

In addition to email fax reports, it is possible to receive fax reports via a script or software located on your HTTP server. For each success and failure, TraiTel will send an HTTP GET query to your URL, indicating the status of the fax, the error code, and whether or not the fax will be retried.

To enable this feature, log-in to your web-messaging account from our website <http://www.traitel.com.au>. From there, select 'Developers' then 'Fax via HTTP Settings' then 'Fax Reports via HTTP'. Simply enter your script or application's URL, then click 'Save'.

Once enabled, when a fax is sent to the gateway, in addition to the Success: 200 message, the gateway will include, on a second line, the phrase:

TrackID: #####

This ID is a unique number assigned to your fax. Each fax report delivered by HTTP will include this ID #, thus allowing you to associate faxes sent with each fax report.

Each HTTP GET query sent to your server will include the following key/value pairs:

- number:* The number to which the fax was sent.
- account:* Your TraiTel account in the format (username@traitel.com.au)
- subidapi:* The API ID of the associated sub-account, if any
- trackid:* The Tracking ID # of the fax.
- seconds:* The duration of the fax delivery, in seconds
- tries:* The number of times TraiTel tried to deliver the fax

errorcode: The error number. Any non-zero value indicates a delivery failure

success: A non-zero value indicates the fax was successfully delivered

speed: The speed at which the fax was delivered

pages: The number of pages delivered

final: A non-zero value indicates that TraiTel will not attempt to send the fax again.

Note: If a fax report is received and both the 'final' and 'success' values are '0', your application **must** expect another report as the fax will be queued for another delivery attempt. Your application **must not** attempt to resend the document until a report is received with a non-zero 'final' value.

In the event of a failure to deliver the HTTP report, the HTTP report will be queued to be retried in 30 minutes from the time of failure. Each report will be queued up to 3 times, after which they will be discarded. The number of minutes in between each retry, as well as the number of retries are configurable in web-messaging.

Output:

After submitting a fax request, the result will be output to the client using the following format:

[HTTP Headers]

["Error" or "Success"] : [Error Code] : [Message]

Error codes:

200: No error – Fax accepted for delivery.

202: There is no recipient.

204: Your balance is too low. (Your current balance will appear in the message)

205: Your username or password is invalid; or the service is temporarily unavailable.

207: There was no attached file.

209: Request not understood; one or more of the required values are invalid or not present.

210: This service is currently disabled in your account. Please contact customer service.

211: You are not currently subscribed to the faxing service.

If you are unsure of your form data, submit it using “mode”=“verify” and check the error code. Correct your data until the error code is “200” (Ok), then submit your form again using “mode”=“send”.

Examples:

Success: 200

Error: 205: Not Authenticated

Error: 210: Service Disabled

Example with HTTP fax reports:

Success: 200

TrackID: 28948720

List of Fax Failure codes sent in HTTP Fax Reports

100,General Error	412,Unspecified Error
101,Line Error	413,Unallocated Number
103,Invalid Handle	414,Another App
104,Invalid Parameter	415,Data Channel Failed
109,Destination Busy	416,No Fax Device
110,No Answer	417,Fax Training Failed
111,No Channel	418,Remote Abort
200,General Error	419,Local Abort
201,Disconnected	420,Modem Negotiation Failed
202,Destination Busy	421,Modem: No Answer
203,Rejected	422,Modem: Carrier Lost
204,No Answer	423,Incompatible Destination
205,No Answer	424,File Access Error
206,Unknown Number	425,Low Memory
207,Invalid Number	426,Illegal Data
208,Number Changed	427,Time Out
209,No Channel	500,General Error
210,Line Error	503,Invalid Handle
211,Network Error	504,Invalid Parameter
212,Unspecified Error	505,Invalid State
213,Unallocated Number	512,File-Open Error

214,Another App	514,File-Read Error
215,Data Channel Failed	600,General Error
216,No Fax Device	601,Disconnected
217,Fax Training Failed	602,Destination Busy
218,Remote Abort	603,Rejected
219,Local Abort	604,No Answer
220,Modem Negotiation Failed	605,No Answer
221,Modem: No Answer	606,Unknown Number
222,Modem: Carrier Lost	607,Invalid Number
223,Incompatible Destination	608,Number Changed
224,File Access Error	609,No Channel
225,Low Memory	610,Line Error
226,Illegal Data	611,Network Error
227,Time Out	612,Unspecified Error
300,General Error	613,Unallocated Number
303,Invalid Handle	614,Another App
304,Invalid Parameter	615,Data Channel Failed
400,General Error	616,No Fax Device
401,Disconnected	617,Fax Training Failed
402,Destination Busy	618,Remote Abort
403,Rejected	619,Local Abort
404,No Answer	620,Modem Negotiation Failed

405, No Answer

406, Unknown Number

407, Invalid Number

408, Number Changed

409, No Channel

410, Line Error

411, Network Error

621, Modem: No Answer

622, Modem: Carrier Lost

623, Incompatible Destination

624, File Access Error

625, Low Memory

626, Illegal Data

627, Time Out

Example form:

The following example is a simple HTML form that will allow a user to submit a fax using a normal Internet browser.

```
<FORM METHOD="POST"
ACTION="http://api.traitel.com/faxgateway.pl"
    ENCTYPE="multipart/form-data">
<INPUT TYPE=HIDDEN NAME="mode" VALUE="send">
<TABLE BORDER=0>
<TR> <TD>Username:                                </TD>
    <TD><INPUT TYPE=TEXT NAME="user" VALUE="">
    </TD></TR>
<TR> <TD>Password:                                </TD>
    <TD><INPUT TYPE=PASSWORD NAME="pass" VALUE="">
    </TD></TR>
<TR> <TD>Subject:                                </TD>
    <TD><INPUT TYPE=TEXT NAME="subject" VALUE="" SIZE=40>
    </TD></TR>
<TR> <TD>To Fax Number:                            </TD>
    <TD><INPUT TYPE=TEXT NAME="recipient" VALUE="">
    </TD></TR>
<TR> <TD>File:                                    </TD>
    <TD><INPUT TYPE=FILE NAME="file" VALUE="">
    </TD></TR>
<TR> <TD>                                        </TD>
    <TD><INPUT TYPE=SUBMIT VALUE="Send">
    </TD></TR>
</TABLE>
</FORM>
```

Username:

Password:

Subject:

To Fax Number:

File:

Perl Example:

The following perl script automatically sends a fax; then verifies and outputs the result.

```
#####  
  
#!/perl  
  
# TraiTel Telecommunications  
  
use HTTP::Request::Common;  
  
use LWP::UserAgent;  
  
use HTTP::Response;  
  
print "Content-Type: text/html\n\n";  
  
# Send the fax  
  
my $errorcode =  
  
    TTSendFax("my_username", "my_password", "subject", "61291234  
567", "C:\\test.pdf");  
  
# Verify result  
  
print "Success!\n"           if ($errorcode == 200);  
print "Failed: No recipient.\n"   if ($errorcode == 202);  
print "Failed: Balance too low.\n" if ($errorcode == 204);  
print "Failed: Authentication error.\n" if ($errorcode ==  
205);  
  
print "Failed: No attachment.\n"   if ($errorcode == 207);  
print "Failed: General error.\n"   if ($errorcode == 209);  
print "Failed: Service is disabled.\n" if ($errorcode ==  
210);  
  
print "Failed: Not subscribed.\n"   if ($errorcode == 211);  
  
exit();
```

```

#####
# Function TTSendFax:
# Parameters:
#   username,password,subject,number,file
#####

sub TTSendFax()

{
    # Get parameters

    my $username = shift;
    my $password = shift;
    my $subject = shift;
    my $recipient = shift;
    my $file = shift;

    my $status; my $code; my $message; my $head; my $body;

    # Send POST request

    my $browser = new LWP::UserAgent;

    my $result = $browser->request(
        POST 'http://api.traitel.com /faxgateway.pl',
        Content_Type => 'multipart/form-data',
        Content => [
            'user'          => "$username",
            'pass'          => "$password",
            'subject' => "$subject",
            'recipient'    => "$recipient",
        ]
    );
}

```

```
        'mode'          => 'send',
        'file'          => [$file],
    ]);

# Extract error code
($head, $body)=split('\n\n', $result->as_string());
($status, $code, $message)=split(':', $body);

return $code;
}
```