

SpamTitan API Users Guide **version 1.0**

October 10th 2008

COPYRIGHT

Copyright © 2008 Copperfasten Technologies. All rights reserved.

The product described in this document is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Copperfasten Technologies gives no condition, warranty, expressed or implied about the fitness or quality of this manual or the accompanying product. Copperfasten reserves the right to make changes to this manual or the accompanying product, without notice to any person or company. Copperfasten shall not be liable for any indirect, incidental, special, or consequential damages, loss of profits, loss of goodwill, loss of reputation or economic loss resulting from the use of this manual or the accompanying product whether caused through Copperfasten negligence or otherwise and based on contract, tort, strict liability or otherwise, even if Copperfasten or any of its suppliers has been advised of the possibility of damages.

SpamTitan is a trademark of Copperfasten Technologies Limited.

Printed in Ireland.

CONTACTING SPAMTITAN CUSTOMER SUPPORT

You can request support by phone or email 24 hours a day, 7 days a week. During our office hours (9am to 5pm, Monday to Friday excluding holidays), one of our engineers will contact you in response to your request.

Telephone: +1 201 984-3271
Email: helpdesk@spamtitan.com
Web: www.spamtitan.com

SPAMTITAN WELCOMES YOUR COMMENTS

We want to know about any corrections or clarifications that you would find useful in our documentation, which will help us improve future versions. Include the following information:

- Version of the manual that you are using
- Section and page number
- Your suggestions about the manual

Send your comments and suggestions to us at the following email address: info@SpamTitan.com

Introduction

SpamTitan exposes some of its functionality via an Application Programming Interface (API). This document is a reference for that functionality, and aims to serve as a reference for developers building tools that integrate with SpamTitan. The API currently supports a RESTful interface, that essentially means you can send an HTTP GET or POST to call exposed methods, and you'll get back an XML document in return.

When you send a request, you'll get a response in XML that looks like this:

```
<Response stat="ok" code="200">
  [Data in XML format - see individual method docs]
</Response>
```

If there's an error, SpamTitan will respond with an error message about the problem.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="404" stat="fail">
  <error>Error Message</error>
</Response>
```

Authentication

Access to the SpamTitan API is limited to IPs on a trusted IP list configured on the Settings->Access/Authentication page, under the "API Allowed Hosts" section. Any attempt to utilize the APIs from any IP address that is not on that list will be denied.

Parameters

Some API methods take optional or requisite parameters. Where applicable, we've documented those parameters. Unknown parameters will be silently ignored.

1. Domain Methods

exists

Tests if the specified domain is been relayed by SpamTitan

URL: `http://10.0.0.82/domain/exists?name=example.com`

Parameters:

- **name:** Required. The name of the Domain to add

Response: If the domain exists:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Domain exists</success>
</Response>
```

If the domain does not exist or the request could not be satisfied then you will get one of the following error codes:

- **501:** Insufficient rights
- **502:** Required parameter not provided
- **405:** Domain not found

For example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="API_DOMAIN_NOTFOUND" stat="fail">
  <error>Domain not found</error>
</Response>
```

add

Adds the specified domain to the current list of domains relayed by SpamTitan to the specified destination server.

URL: `http://10.0.0.82/domain/add?name=example.com&server=1.2.3.4`

Parameters:

- **name:** Required. The name of the Domain to add
- **server:** Required. The destination mail server for this domain. To specify a non-default port (25) append " :8025"

Response: If the domain is successfully added then you will receive a 200 success code:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Domain example.com added</success>
</Response>
```

If the domain already exists then the call will fail.

Possible failure codes include:

- **501:** Insufficient rights
- **502:** Required parameter not provided
- **404:** Domain exists

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="404" stat="fail">
  <error>Cannot add domain: Already exists</error>
</Response>
```

show

Returns extended information of a given domain, specified by domain name as per the required name parameter below.

URL: <http://10.0.0.82/domain/show?name=example.com>

Parameters:

- **name:** Required. The name of the Domain to add

Response: If the domain exists then you will receive a 200 success code along with the extended information regarding the specified domain:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <Data>
    <Domain>abc.com</Domain>
    <Destination_Server>1.2.3.4:25</Destination_Server>
    <Recipient_Verification>None</Recipient_Verification>
  </Data>
</Response>
```

If the domain does not exist then the call will fail. Possible failure codes include:

- **501:** Insufficient rights
- **502:** Required parameter not provided
- **405:** Domain does not exist

edit

Modify the settings for a particular domain

URL: <http://10.0.0.82/domain/edit?name=example.com&server=10.0.0.101>

Parameters:

- **name:** Required. The name of the Domain to add
- **server:** Optional. The new destination server IP address or FQDN for this domain
- **rv:** Optional. The recipient verification method to use for this server. Possible values are: "none", "dynamic", "ldap", "list"
- **dyn_server:** Optional. The recipient verification server to use if the recipient verification is "dynamic". This setting will have no effect if the recipient verification (rv) setting is not set to "dynamic"
- **ldap_server:** Optional.
- **ldap_port:** Optional.
- **ldap_search_dn:** Optional.
- **ldap_password:** Optional.
- **ldap_filter:** Optional.
- **ldap_searchbase:** Optional.
- **email:** Optional. If list based recipient verification is been used, then use this setting to add email addresses to the list. The API call needs to be called once for each email address added.

Response: If the domain exists and was successfully updated, then you will receive a 200 success code along with the extended information regarding the specified domain:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <data>
    <Domain>foobar.com</Domain>
    <Destination_Server>1.3.4.5:25</Destination_Server>
  </data>
</Response>
```

If the domain does not exist or the modification could not be performed then the call will fail.

Possible failure codes include:

- **501:** Insufficient rights
- **502:** Required parameter not provided
- **405:** Domain does not exist

list

List all the domains relayed by SpamTitan

URL: <http://10.0.0.82/domain/list>

Response: Sample output:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <Domains>
    <Domain>abc.com</Domain>
    <Domain>foobar.com</Domain>
    <Domain>xyz.com</Domain>
    <Domain>xyz2.com</Domain>
  </Domains>
</Response>
```

2. Policy Methods

exists

Returns if the specified email address or domain policy exists

URL: <http://10.0.0.82/policy/exists?user=sean@example.com>

Parameters:

- **user:** Required. The email address or domain of the policy to check for

Example: Check if there exists a policy for jack@abc.com

```
$ wget -q -O - "http://localhost/policy/exists?user=jack@abc.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Policy found for jack@abc.com</success>
</Response>
```

Response Codes:

- **200:** Operation successful; Policy exists
- **301:** License not found
- **302:** License Invalid
- **405:** Missing sender parameter
- **601:** Policy not found

show

Return detailed information about the specified policy.

URL: <http://10.0.0.82/policy/exists?user=sean@example.com>

Parameters:

- **user:** Required. The email address or domain of the policy to display. **NOTE:** If no policy exists for the specified user then a 601 code will be returned. However, even if no specific policy exists for a particular email address it will automatically inherit the policy of its parent domain.

Example: Show policy for jack@abc.com

```
$ wget -q -O - - "http://localhost/policy/show?user=jack@abc.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <Policy>
    <id>7</id>
    <policy_name>jack@abc.com</policy_name>
    <virus_lover>N</virus_lover>
    <spam_lover>N</spam_lover>
    <banned_files_lover>N</banned_files_lover>
    <bad_header_lover>N</bad_header_lover>
    <bypass_virus_checks>N</bypass_virus_checks>
    <bypass_spam_checks>N</bypass_spam_checks>
    <bypass_banned_checks>Y</bypass_banned_checks>
    <bypass_header_checks>N</bypass_header_checks>
    <spam_modifies_subj>N</spam_modifies_subj>
    <spam_subject_tag2 />
    <spam_quarantine_to>spam-quarantine</spam_quarantine_to>
    <virus_quarantine_to>virus-quarantine</virus_quarantine_to>
    <banned_quarantine_to>banned-quarantine</banned_quarantine_to>
    <bad_header_quarantine_to />
    <spam_tag_level>-999</spam_tag_level>
    <spam_tag2_level>5</spam_tag2_level>
    <spam_kill_level>5</spam_kill_level>
    <locked>N</locked>
    <digest>N</digest>
    <report_type>N</report_type>
    <report_kill_level>999</report_kill_level>
    <spam_dsn_cutoff_level>0</spam_dsn_cutoff_level>
    <spam_quarantine_cutoff_level>999</spam_quarantine_cutoff_level>
    <digest_language>en_US</digest_language>
  </Policy>
</Response>
```

Response Codes:

- **200:** Operation successful
- **201:** Database Error
- **301:** License not found
- **302:** License Invalid
- **601:** Policy not found

add

Add user policy. The policy which is added will be the same policy as the domain policy. It can subsequently modified using the edit API.

URL: <http://10.0.0.82/policy/add?user=joe@example.com>

Parameters:

- user: Required. The email address of the policy to add

Response:

If the policy is successfully added, then you will receive a 200 success code along with the extended information regarding the specified policy:

```
$ wget -q -O - - "http://10.42.0.170/policy/add?user=sean@abc.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <Policy>
    <id>4</id>
    <policy_name>sean@abc.com</policy_name>
    <virus_lover>N</virus_lover>
    <spam_lover>N</spam_lover>
    <banned_files_lover>N</banned_files_lover>
    <bad_header_lover>N</bad_header_lover>
    <bypass_virus_checks>N</bypass_virus_checks>
    <bypass_spam_checks>N</bypass_spam_checks>
    <bypass_banned_checks>Y</bypass_banned_checks>
    <bypass_header_checks>N</bypass_header_checks>
    <spam_modifies_subj>N</spam_modifies_subj>
    <spam_subject_tag2 />
    <spam_quarantine_to>spam-quarantine</spam_quarantine_to>
    <virus_quarantine_to>virus-quarantine</virus_quarantine_to>
    <banned_quarantine_to>banned-quarantine</banned_quarantine_to>
    <bad_header_quarantine_to />
    <spam_tag_level>-999</spam_tag_level>
    <spam_tag2_level>5</spam_tag2_level>
    <spam_kill_level>5</spam_kill_level>
    <locked>N</locked>
    <digest>N</digest>
    <report_type>N</report_type>
    <report_kill_level>999</report_kill_level>
    <spam_dsn_cutoff_level>0</spam_dsn_cutoff_level>
    <spam_quarantine_cutoff_level>999</spam_quarantine_cutoff_level>
    <digest_language>en_US</digest_language>
  </Policy>
</Response>
```

If the policy cannot be added, for instance, if the specified domain is a non-local domain then you will receive a 602 error code:

```
wget -q -O - "http://10.42.0.170/policy/add?user=sean@abcs.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="602" stat="fail">
  <error>Invalid Domain. Must be local.</error>
</Response>
```

delete

Deletes the specified user policy if it exists.

URL: <http://10.0.0.82/policy/delete?user=sean@abc.com>

Parameters:

- **user:** Required. The email address of the policy to delete.

Example 1: Delete policy sean@abc.com which does exist:

```
$ wget -q -O - "http://10.42.0.170/policy/delete?user=sean@abc.com"
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Policy Deleted</success>
</Response>
```

Example 2: Delete policy simon@abc.com which does not exist:

```
$ wget -q -O - "http://10.42.0.170/policy/delete?user=simon@abc.com"
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="601" stat="fail">
  <error>No policy exists for simon@abc.com</error>
</Response>
```

edit

Modify the settings for an existing policy.

URL: <http://10.0.0.82/policy/edit?user=jokes@foobar.com&attribute=digest&value=Y>

Parameters:

- **user:** Required. The email address or domain of the policy to modify
- **attribute:** Required. The policy attribute which you wish to modify
- **value:** Required. The value to set for the specified attribute

Attributes which you can edit are as follows:

- **virus_lover:** Specifies if virus infected files should be passed for this user (Y/N). Default N.
- **spam_lover:** Specifies if messages exceeding the spam threshold should be passed for this user (Y/N). Default N.
- **banned_files_lover:** Specifies if messages containing banned attachment should be passed for this user (Y/N). Default N.
- **bypass_virus_checks:** Specifies if virus checking should be disabled for this user (Y/N). Default N.
- **bypass_spam_checks:** Specifies if spam checking should be disabled for this user (Y/N). Default N.
- **bypass_banned_checks:** Specifies if banned attachment checking should be disabled for this user (Y/N). Default N.
- **spam_tag2_level;** Specifies the threshold over which messages will be considered spam. Default: 5. Type: float
- **locked:** Specifies if the policy is locked. If a policy is locked, changes to the domain policy will not be inherited by the locked user policy (Y/N). Default: N.
- **digest:** Specifies if this user should receive a quarantine report. N=Never, D=Daily, WD=Week Days, M=Monthly. Default N.
- **report_type:** Specifies the type of quarantine report to send the user. Possible values are N (New items since last report only) or A (all quarantine messages). Default N.
- **digest_language:** Specifies the language that the report should be generated in (cs_CZ/da_DK/de_DE/en_US/fr_FR/nl_NL/ja_JP/it_IT/pl_PL/es_ES). Default: en_US

Example 1: Modify the digest language for sean@abc.com to Japanese

```
$ wget -q -O -
"http://10.42.0.170/policy/edit?user=sean@abc.com&attribute=digest_language&value=ja_JP"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <Policy>
    <id>5</id>
    <policy_name>sean@abc.com</policy_name>
    <virus_lover>N</virus_lover>
    <spam_lover>N</spam_lover>
    <banned_files_lover>N</banned_files_lover>
    <bad_header_lover>N</bad_header_lover>
    <bypass_virus_checks>N</bypass_virus_checks>
    <bypass_spam_checks>N</bypass_spam_checks>
    <bypass_banned_checks>Y</bypass_banned_checks>
    <bypass_header_checks>N</bypass_header_checks>
    <spam_modifies_subj>N</spam_modifies_subj>
    <spam_subject_tag2 />
    <spam_quarantine_to>spam-quarantine</spam_quarantine_to>
    <virus_quarantine_to>virus-quarantine</virus_quarantine_to>
    <banned_quarantine_to>banned-quarantine</banned_quarantine_to>
    <bad_header_quarantine_to />
    <spam_tag_level>-999</spam_tag_level>
    <spam_tag2_level>4</spam_tag2_level>
    <spam_kill_level>5.5</spam_kill_level>
    <locked>N</locked>
    <digest>N</digest>
    <report_type>N</report_type>
    <report_kill_level>999</report_kill_level>
    <spam_dsn_cutoff_level>0</spam_dsn_cutoff_level>
    <spam_quarantine_cutoff_level>999</spam_quarantine_cutoff_level>
    <digest_language>ja_JP</digest_language>
  </Policy>
</Response>
```

Example 2: If you enter an incorrect value you will get the following type response

```
$ wget -q -O -
"http://10.42.0.170/policy/edit?user=sean@abc.com&attribute=bypass_banned_checks&value=F"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="602" stat="fail">
  <error>Invalid value for attribute bypass_banned_checks. Must be Y/N</error>
</Response>
```

Example 3: If you specify an invalid parameter then you will get the following response:

```
$ wget -q -O -  
"http://10.42.0.170/policy/edit?user=sean@abc.com&attribute=badvar&value=123"  
  
<?xml version="1.0" encoding="ISO-8859-1"?>  
<Response code="405" stat="fail">  
  <error>Invalid or Missing parameters</error>  
</Response>
```

3. Blacklist Methods

list

List all Blacklisted email addresses and/or domains for the specified user/domain. If no user or domain is specified then the global blacklist will be returned.

URL: <http://10.0.0.82/blacklist/list>

Parameters:

- **user:** Optional. The email address or domain name of the user/domain for which to view the blacklist

Response: Show all blacklisted items for user jack@abc.com:

```
Request: wget -q -O - - "http://localhost/blacklist/list?user=jack@abc.com"
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <Blacklist>
    <item>@jacksbl.com</item>
    <item>jackbl@email.com</item>
  </Blacklist>
</Response>
```

exists

Tests if the specified email address/domain is blacklisted by the specified user/domain, or globally if no user/domain is specified.

URL: <http://10.0.0.82/blacklist/exists?sender=jokes@foobar.com>

Parameters:

- **sender:** Required. The email address or domain to check for the existence of in a particular blacklist
- **user:** Optional. The email address or domain name of the blacklist to check. If no user is specified, then the global blacklist will be checked

Example 1: Check if the domain *bldomain.com* is blacklisted globally

```
$ wget -q -O - - "http://localhost/blacklist/exists?sender=bldomain.com"
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Blacklist entry exists</success>
</Response>
```

Example 2: Check if the email address *joe@bloggs.com* is blacklisted under the domain *abc.com*. In this case it is not.

```
$ wget -q -O -  
"http://localhost/blacklist/exists?user=abc.com&sender=joe@bloggs.com"  
  
<?xml version="1.0" encoding="ISO-8859-1"?>  
<Response code="501" stat="fail">  
  <error>Blacklist entry not found</error>  
</Response>
```

Response Codes:

- **200:** Operation successful
- **201:** Database Error
- **301:** License not found
- **302:** License Invalid
- **405:** Missing sender parameter
- **501:** Blacklist entry not found
- **502:** User/domain not found

add

Add an email address or domain to a specified users or domains blacklist. If no user/domain is specified then the entry is added to the global blacklist.

URL: <http://10.0.0.82/blacklist/add?user=abc.com&sender=jokes@foobar.com>

Parameters:

- **sender:** Required. The email address or domain to add to the blacklist
- **user:** Optional. The email address or domain name of the blacklist entry to add. If no user is specified, then the entry will be added to the global blacklist.

Example 1: Add *joe@abcblacklist.com* to the blacklist for users in domain *abc.com*

```
$ wget -q -O -  
"http://localhost/blacklist/add?user=abc.com&sender=joe@abcblacklist.com"  
  
<?xml version="1.0" encoding="ISO-8859-1"?>  
<Response code="200" stat="ok">  
  <success>Blacklist entry added</success>  
</Response>
```

Example 2: Add domain *foobar.com* to the global blacklist

```
$ wget -q -O - "http://localhost/blacklist/add?sender=foobar.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Blacklist entry added</success>
</Response>
```

Response Codes:

- **200:** Operation successful
- **201:** Database Error
- **301:** License not found
- **302:** License Invalid
- **405:** Missing sender parameter
- **502:** User/domain not found

delete

Delete an entry from the specified users/domains blacklist. If no user is specified, then delete the entry from the global blacklist.

URL: <http://10.0.0.82/blacklist/delete?user=abc.com&sender=jokes@foobar.com>

Parameters:

- **sender:** Required. The email address or domain to deleted from the blacklist
- **user:** Optional. The email address or domain name of the owner of the blacklist entry to delete. If no user is specified, then the entry will be deleted from the global blacklist.

Example 1: Delete *joe@abcblacklist.com* from the blacklist for users in domain abc.com

```
$ wget -q -O -
"http://localhost/blacklist/delete?user=abc.com&sender=joe@abcblacklist.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Blacklist entry deleted</success>
</Response>
```

Example 2: Attempt to delete *doesnotexist@xyz.com* from the global blacklist

```
$ wget -q -O - "http://localhost/blacklist/delete?sender=doesnotexist@xyz.com"
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="502" stat="fail">
  <error>No such user or domain</error>
</Response>
```

Response Codes:

- **200:** Operation successful
- **201:** Database Error
- **301:** License not found
- **302:** License Invalid
- **405:** Missing sender parameter
- **502:** User/domain not found

4. Whitelist Methods

list

List all Whitelisted email addresses and/or domains for the specified user/domain. If no user or domain is specified then the global whitelist will be returned.

URL: <http://10.0.0.82/whitelist/list>

Parameters:

- **user:** Optional. The email address or domain name of the user/domain for which to view the whitelist.

Response: Show all whitelisted items for user jack@abc.com:

```
Request: wget -q -O - - "http://localhost/whitelist/list?user=jack@abc.com"
-----
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <Whitelist>
    <item>@jacksbl.com</item>
    <item>jackbl@email.com</item>
  </Whitelist>
</Response>
```

exists

Tests if the specified email address/domain is whitelisted by the specified user/domain, or globally if no user/domain is specified.

URL: <http://10.0.0.82/whitelist/exists?sender=jokes@foobar.com>

Parameters:

- **sender:** Required. The email address or domain to check for the existence of in a particular whitelist
- **user:** Optional. The email address or domain name of the whitelist to check. If no user is specified, then the global whitelist will be checked

Example 1: Check if the domain *wldomain.com* is whitelisted globally

```
$ wget -q -O - - "http://localhost/whitelist/exists?sender=wldomain.com"
-----
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Whitelist entry exists</success>
</Response>
```

Example 2: Check if the email address *joe@bloggs.com* is whitelisted under the domain *abc.com*. In this case it is not.

```
$ wget -q -O -
"http://localhost/whitelist/exists?user=abc.com&sender=joe@bloggs.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="501" stat="fail">
  <error>Whitelist entry not found</error>
</Response>
```

Response Codes:

- **200:** Operation successful
- **201:** Database Error
- **301:** License not found
- **302:** License Invalid
- **405:** Missing sender parameter
- **501:** Whitelist entry not found
- **502:** User/domain not found

add

Add an email address or domain to a specified users or domains whitelist. If no user/domain is specified then the entry is added to the global whitelist.

URL: <http://10.0.0.82/whitelist/add?user=abc.com&sender=jokes@foobar.com>

Parameters:

- **sender:** Required. The email address or domain to add to the whitelist
- **user:** Optional. The email address or domain name of the whitelist entry to add. If no user is specified, then the entry will be added to the global whitelist.

Example 1: Add *joe@abcwhitelist.com* to the whitelist for users in domain *abc.com*

```
$ wget -q -O -
"http://localhost/whitelist/add?user=abc.com&sender=joe@abcwhitelist.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Whitelist entry added</success>
</Response>
```

Example 2: Add domain *foobar.com* to the global whitelist

```
$ wget -q -O - "http://localhost/whitelist/add?sender=foobar.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Whitelist entry added</success>
</Response>
```

Response Codes:

- **200:** Operation successful
- **201:** Database Error
- **301:** License not found
- **302:** License Invalid
- **405:** Missing sender parameter
- **502:** User/domain not found

delete

Delete an entry from the specified users/domains whitelist. If no user is specified, then delete the entry from the global whitelist.

URL: <http://10.0.0.82/whitelist/delete?user=abc.com&sender=jokes@foobar.com>

Parameters:

- **sender:** Required. The email address or domain to deleted from the whitelist
- **user:** Optional. The email address or domain name of the owner of the whitelist entry to delete. If no user is specified, then the entry will be deleted from the global whitelist.

Example 1: Delete *joe@abcwhitelist.com* from the whitelist for users in domain abc.com

```
$ wget -q -O -
"http://localhost/whitelist/delete?user=abc.com&sender=joe@abcwhitelist.com"

<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="200" stat="ok">
  <success>Whitelist entry deleted</success>
</Response>
```

Example 2: Attempt to delete *doesnotexist@xyz.com* from the global whitelist

```
$ wget -q -O - "http://localhost/whitelist/delete?sender=doesnotexist@xyz.com"
<?xml version="1.0" encoding="ISO-8859-1"?>
<Response code="502" stat="fail">
  <error>No such user or domain</error>
</Response>
```

Response Codes:

- **200:** Operation successful
- **201:** Database Error
- **301:** License not found
- **302:** License Invalid
- **405:** Missing sender parameter
- **502:** User/domain not found