



# Defensio API

RESTful and Asynchronous

v2.0

©1996–2009, Websense Inc.

All rights reserved.

10240 Sorrento Valley Rd., San Diego, CA 92121, USA

Published 2009

Printed in the United States and Ireland

The products and/or methods of use described in this document are covered by U.S. Patent Numbers 5,983,270; 6,606,659; 6,947,985; 7,185,015; 7,194,464 and RE40,187 and other patents pending.

This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent in writing from Websense Inc.

Every effort has been made to ensure the accuracy of this manual. However, Websense Inc., makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. Websense Inc., shall not be liable for any error or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. The information in this documentation is subject to change without notice.

### **Trademarks**

Websense is a registered trademark of Websense, Inc., in the United States and certain international markets. Websense has numerous other unregistered trademarks in the United States and internationally. All other trademarks are the property of their respective owners.

Other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are the sole property of their respective manufacturers.

# Contents

<b>Chapter 1</b>	<b>Introduction</b> .....	<b>1</b>
	Accessing the Defensio API v2.0 .....	1
	REST conventions .....	1
	HTTP methods .....	2
	Status Codes .....	2
	Supported Formats .....	2
<b>Chapter 2</b>	<b>Resources: Documents</b> .....	<b>5</b>
	POST .....	5
	INPUT .....	5
	client (required) .....	5
	content (required) .....	5
	platform (required) .....	6
	type (required) .....	6
	async .....	6
	async-callback .....	7
	author-email .....	7
	author-ip .....	7
	author-logged-in .....	8
	author-name .....	8
	author-openid .....	8
	author-trusted .....	8
	author-url .....	9
	browser-cookies .....	9
	browser-javascript .....	9
	document-permalink .....	9
	http-headers .....	10
	parent-document-date .....	10
	parent-document-permalink .....	10
	referrer .....	11
	title .....	11
	OUTPUT .....	11
	allow .....	11
	api-version .....	11
	classification .....	11
	profanity-match .....	11
	message .....	11
	signature .....	11
	spaminess .....	12
	status .....	12
	EXAMPLE OUTPUT (yaml) .....	12

---

	GET .....	12
	PUT.....	13
<b>Chapter 3</b>	<b>Resources: Users.....</b>	<b>15</b>
	GET .....	15
	OUTPUT.....	15
	api-version .....	15
	message .....	15
	owner-url .....	15
	status .....	15
	EXAMPLE OUTPUT (yaml) .....	16
<b>Chapter 4</b>	<b>Resources: Statistics.....</b>	<b>17</b>
	GET (basic statistics) .....	17
	OUTPUT.....	17
	api-version .....	17
	false-negatives .....	17
	false-positives .....	17
	learning .....	17
	learning-status .....	17
	legitimate .....	18
	message .....	18
	recent-accuracy .....	18
	status .....	18
	unwanted .....	18
	EXAMPLE OUTPUT (yaml) .....	18
	GET (extended statistics).....	19
	INPUT.....	19
	from (required).....	19
	to (required).....	19
	OUTPUT.....	19
	api-version .....	19
	chart-urls .....	20
	recent-accuracy .....	20
	total-legitimate.....	20
	total-unwanted .....	20
	data (array).....	20
	message .....	20
	status .....	21
	EXAMPLE OUTPUT (yaml) .....	21
<b>Chapter 5</b>	<b>Resources: Profanity-Filter.....</b>	<b>23</b>
	POST .....	23
	INPUT.....	23
	OUTPUT.....	23
	api-version .....	23
	filtered (hash).....	23

---

message .....	23
status .....	23
EXAMPLE OUTPUT (yaml) .....	24



# 1

## Introduction

Defensio API version 2.0 introduces many improvements to the original API.

- ◆ It has been redesigned from scratch to be faster, more flexible, and more generic.
- ◆ The wording is no blog-specific, to help developers better understand how to use the API on sites that do not resemble a blog.
- ◆ Both functionality and accuracy have been improved.
- ◆ Significantly, the API is now fully RESTful (<http://bit.ly/hq9Da>), which makes it more consistent with today's Web standards and conventions.
- ◆ It also supports asynchronous requests when analyzing documents (previously called comments).

This speeds up your Web site considerably, since the site no longer needs to wait for our servers to process a document. Instead, we send you the result once analysis has been completed.

Version 2.0 is fully integrated with the Websense Threat-Seeker Network to prevent malicious or obscene content from crawling onto your website, effectively making Defensio the premier spam and malicious content detection API available today.

## Accessing the Defensio API v2.0

---

Defensio API version 2.0 can be accessed via [api.defensio.com](http://api.defensio.com). The URL will look similar to this:

`http://api.defensio.com/2.0/users/abcdef`

## REST conventions

---

Defensio API version 2.0 is fully REST compliant. For more information about REST, see <http://bit.ly/hq9Da>.

## HTTP methods

Like other REST-compliant APIs, Defensio API v2.0 uses the following HTTP methods to perform different operations on a given resource:

- ◆ GET
- ◆ POST
- ◆ PUT
- ◆ DELETE.

Not all resources support all methods. Please read this document carefully to understand which resources support which methods.

If, for any reason, you are not able to use the PUT or DELETE methods, you can instead use POST, adding a **\_method** field containing the method to use. For example:

```
POST http://.../2.0/users/abcdef/documents/
123456.xml?_method=PUT
```

## Status Codes

Following REST conventions, Defensio API v2.0 uses the following standard HTTP status codes:

- ◆ **200 OK:** Request successfully completed.
- ◆ **400 Bad Request:** Your request is malformed and cannot be understood.
- ◆ **401 Unauthorized:** The API key is invalid or not active.
- ◆ **403 Forbidden:** The API key is not allowed to use this resource.
- ◆ **404 Not Found:** The resource you are trying to access cannot be found.
- ◆ **405 Method Not Allowed:** The requested HTTP method is not supported for this resource. Not all methods (GET, POST, PUT, DELETE) are implemented for all resources.
- ◆ **500 Internal Server Error:** There was a problem at our end. Under normal circumstances, you should not encounter this error.

## Supported Formats

---

Defensio API v2.0 supports the following formats:

- ◆ XML
- ◆ JSON
- ◆ YAML

Append the format that you prefer to the URL you are trying to access. For example:

- ◆ GET http://.../2.0/users/abcdef.xml

- ◆ GET `http://.../2.0/users/abcdef.json`
- ◆ GET `http://.../2.0/users/abcdef.yaml`



# 2

## Resources: Documents

A document contains content to be analyzed by Defensio, or that has been analyzed.

Most of the Defensio API revolves around documents, including the detection of unwanted content.

### POST

---

Uses the syntax:

```
POST /2.0/users/{api_key}/documents.{xml|yaml|json}
```

Creates a new document to be analyzed for spam or malicious content.

### INPUT

#### client (required)

##### Format

Lists the library or plugin name, version number, author name, and author email, with the items separated by a pipe symbol with spaces on either side ( | ). For example:

```
MyPluginName | 1.0 | Joe Author | joe@author-domain.com
```

##### Description

Identifies the plugin or library used to access Defensio.

#### content (required)

##### Format

The string containing the body of the document.

## platform (required)

### Format

One word, lower case. Spaces should be converted to underscores.

### Examples

wordpress, pixelpost, drupal, phpbb, movable\_type

## type (required)

### Accepted values

comment, trackback, pingback, article, wiki, forum, other, test

### Description

Identifies the type of content to be analyzed.

Use **test** only for testing purposes.

When **type** is set to **test**, Defensio parses content for classification and spaminess. For example, if you want our API to return **malicious** as the classification and a spaminess of **0.99**, insert the following in content:

```
[malicious,0.99]
```

There are 3 possible classifications:

- ◆ innocent
- ◆ spam
- ◆ malicious

Spaminess should be a decimal value between 0 and 1.



### Important

Do NOT leave type set to **test** in production. This could represent a significant security breach.

---

## async

### Format

true or false

### Description

Specifies whether document analysis should be done asynchronously.

The default value is **false**, but **true** is strongly recommended:

- ◆ You will obtain better accuracy with asynchronous calls.

- ◆ Use the GET HTTP method to obtain the analysis result at a later time.

Do not poll our servers more than once every 30 seconds for each document. To avoid polling our servers for a result, you can use **async-callback**.

## async-callback

### Format

`http://yourdomain.com/path`

### Description

If you provide a URL, Defensio uses POST to return the analysis result as it is ready. The POST response data uses the format specified in the original request (json, xml, or yaml). The data has the same format as a GET or POST request.



### Important

Defensio will NOT retry unsuccessful callbacks to your server. If you do not see a POST originating from Defensio after 5 minutes, perform a GET request to obtain the analysis result.

---

Occasionally, Defensio may perform more than one POST to your server for the same document. For example, if new evidence indicates that a document is unwanted, even though it was originally identified as legitimate, Defensio might notify you that the classification has changed.

If you do not provide a URL, you must poll our servers with a GET request to obtain the result.

You can debug callbacks using <http://postbin.defensio.com>.

## author-email

### Format

`author@example.com`

### Description

The email address of the author of the document.

## author-ip

### Format

`0.0.0.0` (an IPV4 address)

### **Description**

The IP address of the author of the document.

For example, this could be the IP address of the person posting a comment on a blog.

## **author-logged-in**

### **Format**

true or false

### **Description**

Whether or not the user posting the document is logged onto your Web site, either through your own authentication mechanism or through OpenID.

The default is **false**.

## **author-name**

### **Format**

Firstname Lastname

### **Note**

The name of the author of the document.

## **author-openid**

### **Format**

<http://myopenid.com/authorname>

### **Description**

The OpenID URL of the logged-on user. Must be used in conjunction with **user-logged-in=true**.

OpenID authentication must be taken care of by your application. Only send this parameter if you have successfully authenticated the user with OpenID.

## **author-trusted**

### **Format**

true or false

**Description**

Whether or not the user is an administrator, moderator or editor of your Web site. Pass **true** only if you can guarantee that the user has been authenticated, has a role of responsibility, and can be trusted as a good Web citizen.

The default is **false**.

**author-url****Format**

http://authorname.com

**Description**

The URL of the person posting the document.

**browser-cookies****Format**

true or false (Leave empty if unknown.)

**Description**

Whether or not the Web browser used to post the document (i.e., the comment) has cookies enabled. If no such detection has been made, leave this value empty.

**browser-javascript****Format**

true or false (Leave empty if unknown.)

**Description**

Whether or not the Web browser used to post the document (i.e., the comment) has JavaScript enabled. If no such detection has been made, leave this value empty.

**document-permalink****Format**

http://yourdomain.com/article#comment-51

**Description**

The URL of the document being posted.

## Examples

For a comment on a blog, the permalink URL might be:

```
http://yourdomain.com/article#comment-51.
```

For an article, it might be:

```
http://yourdomain.com/article.
```

## http-headers

### Format

One key and value pair per line, starting with the key, a colon, and the value. For example:

```
HEADER_ONE: value
HEADER_TWO: value
```

### Description

Contains the HTTP headers sent with the request. You can send a few values or all values. Because this information helps Defensio determine if a document is innocent or not, the more headers you send, the better.

In PHP, HTTP headers can be obtained using `$_SERVER`.

## parent-document-date

### Format

```
yyyy-mm-dd
```

### Note

The date the parent document was posted. For example, on a blog, this would be the date the article related to the comment (document) was posted.

If you are using threaded comments, send the date the article was posted, NOT the date the parent comment was posted.

## parent-document-permalink

### Format

```
http://yourdomain.com/article
```

### Description

The URL of the parent document. For example, on a blog, this would be the URL of the article on which the comment (document) was posted.

## referrer

Provide the value of HTTP\_REFERER (note spelling) in this field.

## title

Provide the title of the document being sent. For example, this might be the title of a blog article.

Do not send this information if no title has been provided.

## OUTPUT

### allow

Whether the document should be published on your Web site or not. For example, spam and malicious content are not allowed.

### api-version

The version of the Defensio API being used for this request.

### classification

An the type of content in the document.

The possible values are **innocent**, **spam**, and **malicious**.

### profanity-match

Whether the document matches profanity or other words defined by the user. For example, this is useful to detect obscene comments posted to your Web site. When true, you can obtain a filtered version of the document by POSTing to **profanity-filter**.

The possible values are **true** and **false**.

### message

An information or error message related to the status of the request. This message should be consumed by humans only.

### signature

A unique identifier for the document. You need this value to perform GET and PUT requests on the same document. Signatures should be kept private and never be shared with your users.

## spaminess

A numeric value indicating how strongly the document resembles spam. For example, a document containing many links to pharmaceutical sites is likely to have a very high spaminess value. This value should **only** be used for sorting, and should never be used to determine if a document should be allowed or not.



### Tip

Spaminess should be kept private and never be shared with your users.

---

This returns a floating value between 0 and 1, with 1 being extremely spammy. For example, 0.89.

## status

Whether or not an error was encountered during the request.

A value of **pending** means that the document processing has not yet completed.

The possible values are **success**, **fail**, and **pending**.

## EXAMPLE OUTPUT (yaml)

```
defensio-result:
  api-version: 2.0
  status: success
  message:
  signature: 950b3b9846qrgktjejk6ge
  allow: false
  classification: malicious
  spaminess 0.95
  profanity-match: true
```

---



### Note

When you perform asynchronous requests, the **allow**, **classification**, **spaminess**, and **profanity-match** values will be empty, and the **status** value will be **pending** until the analysis is completed.

---

## GET

---

Uses the syntax:

```
GET /2.0/users/{api_key}/documents/
    {signature}.{xml|yaml|json}
```

Retrieves the status of a comment (innocent or not).

Use this option within 30 days of posting a document. The output is the same as for POST.

## PUT

---

Uses the syntax:

```
PUT /2.0/users/{api_key}/documents/  
    {signature}.{xml|yaml|json}
```

Submits classification errors (false positives or false negatives) to Defensio. With this action, the **allow** value can be only **true** or **false**.

Use this option within 30 days of posting a document. The output is the same as for POST.



# 3

## Resources: Users

### GET

---

Uses the syntax:

```
GET /2.0/users/{api_key}.{xml|yaml|json}
```

Determines if the given API key is valid or not. This should only be used when configuring the client and prior every content analysis (Document POST) .

### OUTPUT

#### api-version

The version of the Defensio API being used for this request.

#### message

An information or error message related to the status of the request. This message should be consumed by humans only.

#### owner-url

The URL where the API key is used

#### status

Whether or not an error was encountered during the request.

The possible values are **success** and **fail**.

## EXAMPLE OUTPUT (yaml)

```
defensio-result:  
  api-version: 2.0  
  status: success  
  message:  
  owner-url: http://yourwebsite.com
```

# 4

## Resources: Statistics

### GET (basic statistics)

---

Uses the syntax:

```
GET 2.0/users/{api_key}/basic-stats.{xml|yaml|json}
```

Returns basic statistics.

### OUTPUT

#### api-version

The version of the Defensio API being used for this request.

#### false-negatives

The number of documents that have been allowed but that should have been blocked.

#### false-positives

The number of documents that have been blocked but that should have been allowed.

#### learning

Whether Defensio is learning from the documents you post.

The possible values are **true** and **false**.

#### learning-status

A message explaining why Defensio is in learning mode.

## legitimate

### total

The total number of legitimate documents analyzed.

## message

An information or error message related to the status of the request. This message should be consumed by humans only.

## recent-accuracy

How accurate Defensio has recently been for this user.

This returns a floating point value between 0 and 1. For example, 0.9525 means 95.25% accurate.

## status

Whether or not an error was encountered during the request. A value of **pending** means that document processing is not complete.

The possible values are **success**, **fail**, and **pending**

## unwanted

### malicious

The number of documents containing malicious content.

### spam

The number of spam documents analyzed.

### total

The total number of unwanted documents.

## EXAMPLE OUTPUT (yaml)

```
defensio-result:
  api-version: 2.0
  status: success
  message:
  recent-accuracy: 0.9975
  legitimate:
    total: 100
  unwanted:
```

```
spam: 100
malicious: 50
total: 150
false-positives: 1
false-negatives: 2
learning: true
learning-status: In learning mode
```

## GET (extended statistics)

---

Uses the syntax:

```
GET /2.0/users/{api_key}/extended-stats.{xml|yaml|json}
```

Returns extended statistics from a given date to another one.

### INPUT

#### from (required)

**Format**

yyyy-mm-dd

**Description**

The starting date

#### to (required)

**Format**

yyyy-mm-dd

**Description**

The ending date

### OUTPUT

#### api-version

The version of the Defensio API being used for this request.

## **chart-urls**

Provides a set of URLs that chart the data provided in the data array.

## **recent-accuracy**

A chart of representing the recent accuracy per day for the specified period.

## **total-legitimate**

A chart of representing the total number of legitimate documents per day for the specified period.

## **total-unwanted**

A chart of representing the total number of unwanted documents per day for the specified period.

## **data (array)**

### **date**

The date of the current set of data in the format **yyyy-mm-dd**.

### **false-negatives**

The number of false negatives for the specified date.

### **false-positives**

The number of false positives for the specified date.

### **legitimate**

The number of legitimate documents processed on the specified date.

### **recent-accuracy**

How accurate Defensio has recently been for the current user on the specified date.

This is returned as a floating point value between 0 and 1. For example, 0.9525 means 95.25% accurate.

### **unwanted**

The number of unwanted documents processed on the specified date.

## **message**

An information or error message related to the status of the request. This message should be consumed by humans only

## status

Whether or not an error was encountered during the request.

The possible values are **success** and **fail**.

## EXAMPLE OUTPUT (yaml)

```
defensio-result:
  api-version: 2.0
  status: success
  message:
  data:
    - date: 2009-09-01
      recent-accuracy: 0.9975
      legitimate: 100
      unwanted: 500
      false-positives: 1
      false-negatives: 0
    - date: 2009-09-02
      recent-accuracy: 0.9985
      legitimate: 50
      unwanted: 475
      false-positives: 0
      false-negatives: 0
    - date: 2009-09-03
      recent-accuracy: 0.9992
      legitimate: 100
      unwanted: 500
      false-positives: 1
      false-negatives: 0
  chart-urls:
    recent-accuracy: http://domain.com/chart/123456
    total-unwanted: http://domain.com/chart/abcdef
    total-legitimate: http://domain.com/chart/xyzabc
```



# 5

## Resources: Profanity-Filter

### POST

---

Uses the syntax:

```
POST /2.0/users/{api_key}/profanity-filter.{xml|yaml|json}
```

Returns a filtered version of the provided fields. The filtering is based on a default dictionary and one previously configured by the user.

### INPUT

Any field.

### OUTPUT

#### **api-version**

The version of the Defensio API being used for this request.

#### **filtered (hash)**

The filtered version of the fields passed as input.

#### **message**

An information or error message related to the status of the request. This message should be consumed by humans only.

#### **status**

Whether or not an error was encountered during the request.

The possible values are **success** and **fail**.

## EXAMPLE OUTPUT (yaml)

```
defensio-result:  
  api-version: 2.0  
  status: success  
  message:  
    filtered:  
      content: This site ***** *****  
      author-name: Download free **** from our website
```