PHPit
Totally PHP

No more spam; Akismet and PHP to the rescue!

By Dennis Pallett

Abstract

In this article you'll learn how to use the Akismet anti-spam service to prevent any spam from being added to your PHP website. Live examples and demo's are included.

Introduction

Spam is one of the biggest problems on the internet, and I'm sure you've received a fair share of spam in your inbox as well. It's always been a cat-and-mouse game between developers and spammers, and it seems the spammers are always one step ahead. Simple measures like black lists or URL filtering simply don't work anymore, and spam always finds its way through. That's why we need a different solution, one that actually works. Introducing Akismet, the anti-spam service by the brilliant guys from Automattic who are also behind WordPress and WordPress.com.

I've started using Akismet myself recently, and it has completely cut down all the spam I was receiving. Normally I'd get about 700 spam comments a day on my various blogs and websites (including PHPit), but since installing Akismet I've only received four. That means that approximately 99,5% is being caught by Akismet, which is an extremely good rate, and what's even better is that Akismet hasn't caught one good spam comment (no positives). Sounds pretty good, right?

That's why in this article I'll show you how to integrate Akismet in your own PHP scripts. But first let's have a closer look at how Akismet works.

How Akismet works

Akismet is basically a web service which analyzes your comments. Your script sends the contents of the comment to the Akismet service, using standard HTTP commands, and then Akismet analyzes the comment, and returns a thumbs up or down on the comment. Since Akismet is used by some many people (34,216,155 spams caught so far) its ability to analyze a comment is excellent, and it can detect almost any type of spam, and it's also smart enough to let normal comments go through.

Of course, nothing is perfect, so Akismet is bound to make a few mistakes here and there. That's why it's also possible to send suggestions to the Akismet service. If you discover that Akismet has missed a spam message, you can tell Akismet it's spam after all, which improves the analyzing ability of Akismet. Similarly, it's also possible to tell Akismet a message is actually a "good" comment and not spam after all.

A common problem with these types of central anti-spam services is that it's very easy for spammers to

poison the spam database, by marking spam comments as good comments and good comments as spam. According to the Akismet FAQ, they have developed an algorithm against this type of behavior, but you also need an API key to use Akismet, so it's very easy for the people behind Akismet to ban any user.
Let's get started with some code, so quickly get your API key at WordPress.com, since you need to have an API key to use Akismet. When you have your API key, continue with this article.

Using the Akismet service

Akismet is very easy to use, and it's all basic REST calls, and all you really need to use is the fsockopen() function to send requests to Akismet. The full API documentation can be found by clicking here.

But since we're a lazy bunch we're going to use a pre-made PHP class which handles all the nitty gritty for us, leaving only the easy things. Head on over to the Akismet Development Page and download the PHP 4 class by Bret Kuhns.

Let's get started with an example, which demonstrates the basic usage of the Akismet service, using the PHP class by Bret:

```
<?php
 include ('Akismet.class.php');

// Create object instance, and pass website URL and API key
 $akismet = new Akismet('http://www.phpit.net', '[YOUR API KEY HERE]');

$comment = "Hi!\n\nI really like your website\n\n<a href=\"http://www.viagra-today.com\">Buy Viagra</a>";

$akismet->setCommentAuthor('Buy my viagra');
 $akismet->setCommentAuthorEmail('viagra@casino-now.com');
 $akismet->setCommentAuthorURL('http://www.buy-viagra.com');
 $akismet->setCommentContent($comment);

if($akismet->isCommentSpam() == true) {
        echo 'We don\'t want your spam, punk!';
 } else {
        echo 'Great comment, thanks!';
 }

?>
```

As you can see, it's really simple to use Akismet. Let's go through this example. First, we create a new instance of the Akismet class, and immediately pass the URL of our website and our API key. These two values are required by Akismet, and especially the API key must be correct. If this is incorrect, you will get a fatal error and the script execution will be halted, so make sure it's correct.

In the next few lines we set various properties of the comment which have been submitted by someone. In the example we use fixed values, which are very spammy (buy Viagra, casino's, etc).

Finally, we use the isCommentSpam() method to check whether the comment is spam. I'm sure you can guess the outcome in this example. And that's all really; I told you it was easy!

If you want to play with Akismet yourself and test different messages, click here for a live demo.

Helping out Akismet

Since we're using the Akismet service in our script it's almost our honor-bound duty to help out as well, so let's do that by helping Akismet with its mistakes.

The Akismet class we're using comes with two methods to help Akismet. The first one, called submitSpam(), is used to mark a comment as spam, and it's used almost the exact same way as the isCommentSpam() method we've been using previously. See the example below:

```
<?php
 include ('Akismet.class.php');

// Create object instance, and pass website URL and API key
 $akismet = new Akismet('http://www.phpit.net', '[YOUR API KEY HERE]');

$comment = "Hi!\n\nI really like your website\n\n<a href=\"http://www.viagra-today.com\">Buy Viagra</a>";

$akismet->setCommentAuthor('Buy my viagra');
 $akismet->setCommentAuthorEmail('viagra@casino-now.com');
 $akismet->setCommentAuthorURL('http://www.buy-viagra.com');
 $akismet->setCommentContent($comment);

// Mark it as spam
 $akismet->submitSpam();

?>
```

The second method is called submitHam(), which is used to mark a comment as not-spam (Akismet calls these good comments 'Ham'). It's used in the exact same way as the submitSpam() method:

```
<?php
 include ('Akismet.class.php');

// Create object instance, and pass website URL and API key
 $akismet = new Akismet('http://www.phpit.net', '[YOUR API KEY HERE]');

$comment = "Hi!\n\nI really like your website\n\n<a href=\"http://www.viagra-today.com\">Buy Viagra</a>";

$akismet->setCommentAuthor('Buy my viagra');
 $akismet->setCommentAuthorEmail('viagra@casino-now.com');
 $akismet->setCommentAuthorURL('http://www.buy-viagra.com');
 $akismet->setCommentContent($comment);

// Mark it as spam
 $akismet->submitHam();

?>
```

And there you have it! You can now help the Akismet service by marking comments as spam and marking badly tagged comments as ham. Make sure you properly mark things though, and don't try to mess around with the Akismet service, because your API key could be banned.

Conclusion

In this article I've shown you how the Akismet anti-spam service works, and how to integrate it into your own PHP scripts. I've had much success with it, and so have many other people. It's probably one of the best anti-spam services around at the moment.

If you want to have a look at a few more examples of how Akismet is used, have a look at their development page and download the WordPress plugin to see a full-blown implementation of Akismet.

If you have any questions or comments about this article, please leave them below and I will respond as fast as possible. If you want to discuss this article, or something else, join us at PHPit Forums for more PHP talk.

About this PDF

For more information, contact us at support@pallettgroup.com.