LINKBACKS

ILDIKÓ TÓTH | IST 479 | ADVANCED TOPIC PAPER

INTRODUCTION

The general purpose of a linkback is to keep track of who may be linking to your page. They also provide the site owner other data from the referring site such as the title of the page that contains the link, the language and URL of the page, as well as an excerpt from around the link. There are three kinds of linkbacks: Trackbacks, Pingbacks and Refbacks, all of which accomplish this task in different ways.

REFBACKS

The refback is the only method of linkback that uses *HTTP referrers*. The HTTP Referrer is the URL that is passed in the notification that is sent when someone accesses a site through a link on another page. i.e. my website refers to twitter.com; if anyone clicks on the link to twitter on my page, it could send a notification to Twitter.com, which can use the URL of my site as an HTTP referrer to identify where the visitor to their site came from (my site). In PHP for example, this variable can be accessed through the variable HTTP_REFERER. The HTTP referrer value is usually in the file header, and can be extracted.

HTTP referrers have become something of a controversy (and less used since the rise of trackbacks and pingbacks) as they can pose a security issue with servers logging all refers. Also, the rise of referrer spam, which is more of an annoyance, has caused added problems. Referrer spam is a term used for refers that come from spam sites. Spammers are adding refers to their sites in the hope that bloggers will refer back to them as has been the recent trend; in order to create more conversations, bloggers are referring back to basically anyone who refers to them.

Although it is good that there need not be any code on the server to make refbacks work; the notification of reference is sent when someone clicks on a link, they have several disadvantages. First, the refback can clearly be spoofed, thus the information referred back cannot be trusted 100 percent of the time. Also, the only time a referring site is recognized is when someone actually clicks on a link on the referring site. Otherwise, an <a> tag linking to another page is not noticed by that other site.

TRACKBACKS

Trackbacks and Pingbacks are a lot smarter than a Refback. The biggest difference is that the server is involved. Specifically, if a trackback or pingback is enabled and correct code is written, the linking server is able to analyze the HTTP code, run through the page recognizing any links to outside pages, and send notifications to the linked server.

A trackback uses the HTTP POST mechanism, and provides an explicit link between two sites versus an implicit link (requires outside action such as clicking a link). In order for a Trackback to exist between two sites, site A doesn't necessarily have to have an implicit link to site B. When a trackback is enabled, site A can send a Trackback to site B whenever they want to; i.e. if they decide to post something interesting and want to share it.

HOW TRACKBACKS WORK

The client sends a POST request with a content-type header where the type is *application/x-www-form-urlencoded*. The client sends the HTTP POST to a trackback Ping URL. This URL is to the referred site. An example request might look like this:

```
POST http://www.example.com/trackback/5
Content-Type: application/x-www-form-urlencoded; charset=utf-8
title=Foo+Bar&url=http://www.bar.com/&excerpt=My+Excerpt&blog_name=Foo
```

This request is processed by the server and sends the associated title, url, exerpt, and blog name to the referred server. The url is the only one that is required. When the ping is successful, the referred server must send an acknowledgment in this format:

```
<?xml version="1.0" encoding="utf-8"?>
    <response>
    <error>0</error>
    </response>
```

The encoding must be the same as the originating request (in this case utf-8). If the request is unsuccessful, the error tag contains a number, and then the error message inside the tags <message></message>.

When a trackback is sent, all the necessary information is included in the notification to the referred server. However, it is more susceptible to spamming as compared to a pingback. It also requires a longer response from the referred server, which in turn causes more network traffic and more processing time. In addition, if there's an auto-discovery notification in place, the code may prevent XHTML validation. Fortunately, this last problem can be avoided by using HTML comments; i.e. <--! Auto-discovery code in here -->

PINGBACKS

Pingbacks are similar to trackbacks, but there are significant technical differences between the two. A pingback can use an HTML or XTML header mechanism or a <link> element, but instead of POST, pingbacks use *XML-RPC* to notify the referred site.

The header mechanism should be in the header of an HTML/XHTML document. The <link> tag referring to the pingback URL must be before the </head> and <body> tags. XML-RPC interface suggests that there be either the header mechanism or the link mechanism, but not both unless they both have the same information. If they contain different information, the data in the header overrides the <link> tag.

HOW PINGBACKS WORK

For example, on my website which is hosted by Wordpress, the following header can be extracted:

HTTP Response Header

Name	Value
Status: HTTP/1.1 301	
Date:	Sat, 01 May 2010 21:48:54 GMT
Server:	Apache
X-Pingback:	http://www.ildikototh.com/wordpress/xmlrpc.php
X-Powered-By:	PHP/4.4.9
Location:	http://www.ildikototh.com/
Connection:	close
Transfer-Encoding:	chunked
Content-Type:	text/html; charset=UTF-8

The highlighted URL is the link to which the ping request would be sent to. This request includes the source and target URL; the source being the post that is referencing the target and the target being the href part of the link on the source site.

The format of such a request to http://www.ildikototh.com/wordpress/xmlrpc.php (from alice.example.org) would be as follows:

```
pingback.ping ('http://alice.example.org/#p123', 'http://www.ildikototh.com/#foo')
```

Once my server receives such a ping, it will document the source URL (http://alice.example.org/#p123), check that it actually does have a reference back to ildikototh.com, and that the target url (http://www.ildikototh.com/#foo) is actually a post on my site (to avoid spamming). In addition, my server scans that page for other information such as blog title, author, language, etc. Finally recording all of the information in my database, which then could (if enabled), add to the list of references wherever I have it implemented.

In case there is nothing about X-Pingback in the header, a link tag like <link rel="pingback" href="http://www.ildikototh.com/wordpress/xmlrpc.php"> would also do that trick to trigger the sending of a pingback ping.

The pingback mechanism is less susceptible to spamming because it does check back and because the information passed in the pingback are simply the URLs of the source and target. This is unlike the trackback which sends all metadata like title, author, etc. through the notification in a trackback. With a pingback however, because of the lack of data inside the notification, the target server must scan the source's page for this extra information and parse it in order to get the data it needs.

BIBLIOGRAPHY

Neumann, D. (2009). *Coding your own blog Pingback in php*. Retrieved May 1, 2010, from Quiet Earth: http://www.quietearth.us/articles/2006/10/30/Coding-your-own-blog-Pingback-in-php

HTTP_referer. (2010). Retrieved May 1, 2010, from Wikipedia: http://en.wikipedia.org/wiki/HTTP_referer

Linkback. (2010). Retrieved May 1, 2010, from Wikipedia: http://en.wikipedia.org/wiki/Linkback

Trackback. (2010). Retrieved May 1, 2010, from Wikipedia: http://en.wikipedia.org/wiki/Trackback

Hickson, I. (2010). *Pingback 1.0*. Retrieved May 1, 2010, from Specs: http://www.hixie.ch/specs/pingback/pingback

sixapart.com. (2010). *TrackBack Technical Specification*. Retrieved May 1, 2010, from Six Apart: http://www.sixapart.com/pronet/docs/trackback_spec