

Tracking single page sites with Google analytics code

Posted At : 7 March 2010 18:45 | Posted By : Shaun McCran
Related Categories: Google, Best practices, Javascript

If you have a framework that controls the URL in some way then you may have an issue when it comes to Google Analytics tracking. In this blog entry I will examine how to alter your GA tracking so that you can specify custom URL values to track. I will then apply this to a FuseBox framework.

Traditionally Google Analytics code tracks each page impression by capturing the URL and all values of the Query string, storing it in a cookie and sending it back to the Google search engine through a JavaScript call. When all your site / applications pages are "index.cfm" it may prove difficult to generate useful Analytics information.

In this example I am using a FuseBox framework. If you are unfamiliar with this, the premise is that all the templates use "index.cfm" and then pass through two parameters. The first is the component to use as a controller (we will use public.cfc) and the second value is the function name to call within that controller. So our URL may look like this:

```
www.mysite.com/index.cfm?go=public.login
```

In this case our framework will go to the controller/public.cfc and find the function "login". Google analytics will log this hit as "/index.cfm?go=public.login". This is not at all user friendly, and would be very difficult to represent usefully to a client in a report.

Instead we alter the Google Analytics code so that it is logging a pseudo URL, rather than the actual URL.

The Google Analytics code resides just before the end body html tag, at the end of your page. In the FuseBox example I am using one layout template, "lay_template.cfm" so I only need add it there.

Firstly call the Google Analytics code.

```
gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http://www.");
</script type="text/javascript">
<script src=' + gaJsHost + "google-analytics.com/ga.js" type='text/javascript'>
</script>
```

Next we will call the Google Analytics JavaScript that builds and sends the request. The pageTracker variable is created, and your Google Analytics code is added to it. Next we call the initData function. There seems to be confliction information online about whether this is required or not. Essentially it tells the Google Logging mechanism to prepare for a request. Next we will add a line that remove the need for a Fully Qualified Domain Name (FQDN). This has a negative impact on the next line of code, and must be run before the trackPageview code.

The trackPageview code is usually left empty, but in this case we are inserting the fuseaction that has been requested. So if we are following our example above this will read as "/login". You can have as long a string here as you require. IE it could be "campaign5/landingpage2". In this way Google will log the hit as you specify as this overwrites the URL cookie that Google normally creates.

```
</script type="text/javascript">
var pageTracker = _gat._getTracker("UA-XXXXXX-XX");
pageTracker._initData();
pageTracker._setDomainName("none");
pageTracker._trackPageview('/#variables.myFuseBox.originalFuseaction#');
</script>
```

This principle can also be used to track AJAX URL requests, as they do not create a unique page impression.