

Building an advertising system with PHP, Part 1

By Dennis Pallett

Introduction

Most of us who run websites will most likely want to add advertising on our websites some time in the future. The easiest way to do this is to simply take the advertising code (be it HTML or JavaScript), and paste it directly into the code of our HTML. But what if we eventually get multiple advertisers for the same spot? In the beginning this is unlikely, but as your website grows and becomes more popular, it's to be expected that you will have accommodate for multiple advertisers.

In that case you must use some sort of advertising management script. The best one around, phpAdsNew, is completely free and can be downloaded at <http://www.phpadsnew.com>, but this might be slightly overkill for most websites, since phpAdsNew is a complete ad management application, packed with features you probably won't need anyway. Why don't we create our own simple advertising system?

In this article, the first part of a new three part series, I will show you how to build your own advertising system. In this first part I will take you through the basics of dynamic JavaScript, and how to serve multiple advertisements in the same spot. Let's get started.

Dynamic JavaScript using PHP

The easiest way to serve different advertisements is to use an external JavaScript file, e.g.

```
<html>
  <head>
    <title>Demo 1</title>
  </head>

  <body>

    <script src="ads.js" type="text/javascript"></script>

  </body>
</html>
```

The ads.js file can insert any content using document.write (or by adding new elements to the DOM). This means we can use JavaScript to serve different ad's in the same spot, but why do all the backend work in JS, when we've got PHP to work with?

First of all, point the script tag to a PHP file, like this:

```
<html>
  <head>
    <title>Demo 1</title>
  </head>

  <body>

    <script src="js1.php" type="text/javascript"></script>

  </body>
</html>
```

Then paste the following in js1.php:

```
<?php

$time = date("g:i:s A", mktime());
header ("Content-Type: text/javascript");
echo "document.write ('The time is now $time');";

?>
```

[\(View Live demo\)](#)

This simple example demonstrates that it's possible to use JS and PHP together to add dynamic content. The js1.php uses JS to output extra content, with can contain anything.

Let's have a look at creating a really bare-bones advertising system now.

Creating the ad system

Our ad system will pull it's advertisements from an array that looks like this:

```
// ### Advertisements:
$sads = array(
    // Array (link to website, image url)
    array('http://www.phpit.net', './phpitbanner.gif'),
    array('http://www.aspit.net', './aspitbanner.gif'),
    array('http://www.google.com', './468x60banner.gif'),
);
```

As you can see it's very easy to understand, and all we need is two things: a link and an image url.

What our ad system must do now is pull a random ad from the above array, and display it using JavaScript. The code to do this looks something like this:

```
<?php

// ### Advertisements:
$sads = array(
    // Array (link to website, image url)
    array('http://www.phpit.net', './phpitbanner.gif'),
```

```
array('http://www.aspit.net', './aspitbanner.gif'),
array('http://www.google.com', './468x60banner.gif'),
);
```

```
// Get a random ad
```

```
$adkey = array_rand($ads, 1);
```

```
// Get ad
```

```
$ad = $ads[$adkey];
```

```
$url = $ad['0'];
```

```
$image = $ad['1'];
```

```
// Display ad:
```

```
header ("Content-Type: text/javascript");
```

```
echo "document.write ('<a href=\"\$url\"><img src=\"\$image\" alt=\"Ad Banner\" style=\"border:0;\" width=\"468  
\" height=\"60\" /></a>');";
```

```
?>
```

If you now run the above code in conjunction with the ad page, which looks like this:

```
<html>
  <head>
    <title>Ad System (Part 1)</title>
  </head>

  <body>

    <script src="ad1.php" type="text/javascript"></script>

  </body>
</html>
```

[\(View Live Demo\)](#)

You will notice that random ads are displayed. We've now got a very basic, but functional advertising system, using nothing more than some simple PHP and JavaScript.

Creating a frequency cap

Often you might want to limit a certain advertisement to a certain number of impressions, i.e. an ad only gets 1000 impressions and not more because the advertiser paid for only 1000. Our ad system can't do that yet, but we can certainly build it in.

To be able to limit an ad, the system must log each view of an advertisement, which means we must write to a log file. To keep everything as simple as possible, our ad system will simply write to a log file every time an ad is viewed like.

The new code for the ad system looks like this:

```
<?php
```

```
// ### Log file
```

```

$logfile = 'log.txt';

// ### Advertisements:
$sads = array(
    // Array (id, link to website, image url, impressions)
    array('phpit', 'http://www.phpit.net', '../phpitbanner.gif', 5),
    array('aspit', 'http://www.aspit.net', '../aspitbanner.gif', 5),
    array('gen', 'http://www.google.com', '../468x60banner.gif', 1000),
);

// ##### Don't edit below #####

$logpath = dirname(__FILE__) . '/';
$logfile = $logpath . $logfile;

// Open log file
$log = array();
if (file_exists($logfile)) {
    $log = file_get_contents($logfile);
    $log = unserialize($log);

    // Update imps
    for ($i = 0; $i < count($sads); $i++) {
        $sad =& $sads[$i];

        if (!isset($log[$sad['0']])) continue;
        $sad['3'] = $log[$sad['0']];
    }
}

// Which ads have no impressions left?
$temp = array();
foreach ($sads as $a) {
    if (intval($a['3']) > 0) {
        array_push($temp, $a);
    }
}
$sads = $temp;

// Get a random ad
$sadkey = array_rand($sads, 1);

// Get ad
$sad = $sads[$sadkey];
$url = $sad['1'];
$image = $sad['2'];

// Display ad:
header ("Content-Type: text/javascript");
echo "document.write ('<a href=\"\$url\"><img src=\"\$image\" alt=\"Ad Banner\" style=\"border:0;\" width=\"468
\" height=\"60\" /></a>');";

```

```
// Decrease impressions
$ad['3'] = $ad['3']-1;
$log[$ad['0']] = $ad['3'];

// Serialize log
$log = serialize($log);

// Write log
$f = fopen($logfile, 'w');
fwrite($f, $log);
fclose($f);
?>
```

What this code does is first get the log file, and get the updated number of impressions for each ad (with each having a unique id). After that it sorts through the ads array, and removes all the advertisements that are at zero impressions, because they shouldn't be shown anymore. Finally it displays the advertisement, decreases the impressions count by one, and writes the new log file.

This code will count down till zero, and will then stop displaying the ad, and thus we have built in a frequency cap. Simple and messy, but it does the job!

Conclusion

In this first part of the 'Building an advertisement system in PHP' series I have shown you a really simple way of creating an ad system for your website. The above code is still very messy, and not ready for production use at all, but it was more of an introduction, and in the next part we'll start creating some beautiful code.

In the next part I will move away from the simple approach, and show you how to run our ad system from a MySQL database. We will also add two new features to our ad system (priority ads, and different types), and we'll be creating a simple admin page for the ad system.

If you would like to discuss this article, hop on over to the [PHPit Forums](#) or use the comments below.

[The demos from this article are downloadable by clicking here.](#)

About this PDF

You may distribute this PDF in any way you like, as long as you don't modify it in any way. You can ONLY distribute the unchanged original PDF.

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