

This article explains how to implement a realtime playlist on your website, so that visitors can view the song currently played on air, and also get access an archive of all played songs.

## Requirements

The solution presented here assumes that your website is running a web server with PHP enabled, and you also have access to an SQL server from there. Any kind of SQL server will do, we will use MySQL as an example as most people have that installed for their CMS/blog/forum anyway.

On the client (playout) side, you need a copy of Aircast with the HTTP GET logging interface available, i.e. at least Home Studio.

## Architecture

Our solution works as follows:

- A list of all played songs (start time, artist, title) is kept in a MySQL table.
- We place a “secret” PHP script on the webserver that receives new entries from Aircast and places them into the table.
- Aircast uses HTTP GET logging to transfer the song information whenever you start a new song.
- When the user visits your website, the PHP script receives the current item from the table.

## Setup

### Preparing the MySQL Server

We assume that you already have a MySQL server running, and that PHP is able to access it. For a tutorial, see e.g. here: <http://www.freewebmasterhelp.com/tutorials/phpmysql>

We have to create a new table, e.g. “playlistlog”, with three columns:

- The date/time at which the song was started
- The name of the artist
- The title of the song

You can add more fields if you like, and use Aircast Attributes to populate them.

It is also a good idea to create an [index](#) on the date/time column. This will speed up queries alot as the table grows.

To create the table and the index, open the MySQL console (or use a graphical frontend like phpMyAdmin) and issue the following commands:

```
CREATE TABLE playlistlog (  
  
    starttime DATETIME NOT NULL,  
  
    artist VARCHAR(255),  
  
    title VARCHAR(255)  
  
);  
  
CREATE INDEX playlistlog_starttime_index ON playlistlog(starttime);
```

# Preparing PHP Receive Script

Step #2 is to create a PHP script that receives new logging entries from Aircast via HTTP GET, and saves them into the MySQL table.

The script should be “secret” so that no unauthorized person can call it. Consider using an `.htaccess` file to protect it with a password, or only allow calls from specific IPs (your studio computers).

Let's assume the script is called `playlistlog.php` and resides in the top level folder of your webserver (i.e. <http://www.yourserver.com/playlistlog.php>). Here's the content of the script:

```
<?php
```

```
$db = mysqli_connect('localhost', 'Aircast', 'secret', 'logdatabase');
```

```
$artist = mysqli_escape_string($db, $_GET['artist']);
```

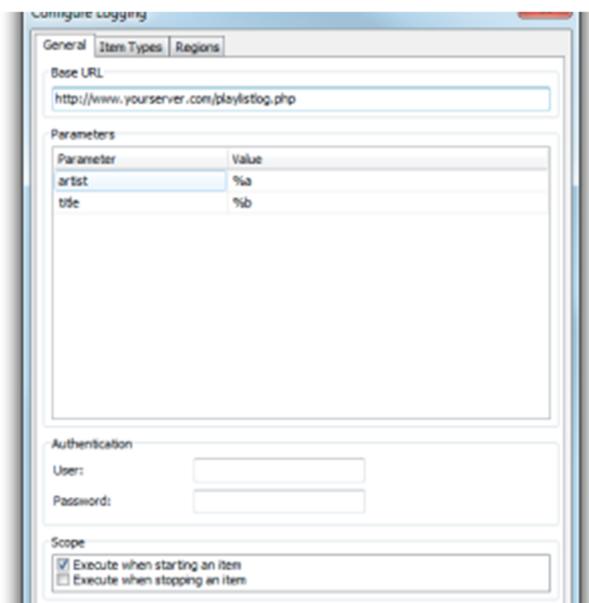
```
$title = mysqli_escape_string($db, $_GET['title']);
```

```
mysqli_query($db, "INSERT INTO playlistlog (starttime, artist, title) VALUES (NOW(), '$artist', '$title')");
```

Don't forget to adjust the MySQL login data in the first line (`mysqli_connect(...)`) - in this example, the server is running on localhost, the user is “Aircast” with password “secret”, and the database name is “logdatabase”.

## Setting up Aircast HTTP Logging

We use HTTP GET to transfer new entries to the webserver, so go to *Aircast Configuration* → *Logging*, click Add and add a new HTTP GET interface. The dialog will look like this:



As Base URL, enter the URL of the secret `playlistlog.php` script, e.g. <http://www.yourserver.com/playlistlog.php>.

In the Parameters table, add two parameters: *artist* with value *%a*, and *title* with value *%b*. **Use the cursor-down key to add more lines to the table.**

That's pretty much it. If you protected your script with an `.htaccess` file, enter the user/password. If you only want particular item types to be logged (e.g. only music but no jingles), go to the *Item Types* tab and uncheck the types that should be filtered.

Now run Aircast, start a couple of songs and watch how the `playlistlog` table is populated. If it doesn't work, it's a good idea to check the Aircast System Log for any error messages (double-click the status bar at the bottom of the Aircast main window).

## Retrieving the Data

Once the data is in the `playlistlog` table, you can use very simple PHP code to retrieve the last few songs:

```
<?php

$db = mysqli_connect('localhost', 'Aircast', 'secret', 'logdatabase');

$res = mysqli_query($db, 'SELECT starttime, artist, title FROM playlistlog ORDER BY
starttime DESC LIMIT 10');

print "<table>\n";

while (list ($starttime, $artist, $title) = mysqli_fetch_row($res)) {

    print "<tr><td>$starttime</td><td>$artist</td><td>$title</td></tr>\n";

}

mysqli_free_result($res);

print "</table>\n";
```

As all entries are stored along with the date/time the song was started, it is also possible to implement a search function that looks up the songs played in a given time frame.