Sessions 4 and 5: Web Services which Update Data and their Clients

POST-based web services

So far, we have written a web service which ***searches*** for music and connected to it from a "fan" website to list all songs by that artist. However, web services can also ***update*** information. For example, an airline might provide a web service to allow clients to book flights with them, and then a flight searching site such as Expedia could communicate with the booking web service to allow people to book flights through Expedia.

Or, a hotel could provide a web service to allow people to reserve rooms, and then a site such as booking.com could connect to this web service to allow people to book the hotel through them.

In this session, we will look at web services which allow clients to ***update*** data. Notice how I have modified the ***wadsongs*** database to include a ***downloads*** column. You are going to write a HitTastic! "download" web service to allow client websites to download music provided by HitTastic!.

Note that (obviously) this will not really download a piece of music. It will simply increase the number of ***downloads*** for that song in the database by one.

Standard Exercise 1

- Write the "download" web service. It should read in an ID, as a POST request (***why a POST request?***), connect to the database, and increase the number of downloads by one for that song.
- Test your web service with a form allowing the user to enter an ID, e.g.

```
<html>
<body>
<h1>POST Web Service Test</h1>
<form method="post" action="downloadwebservice.php">
ID:<input name="id" />
<input type="submit" value="Go!" />
</form>
</body>
</html>
```

Error checking in web services

- Try modifying your "download" web service to check for errors. An error should occur if the user does not have enough money in their account. Find this out as follows (choose the username in the ht_users table with the ID equal to the middle digits of your computer, as described in class, also see ***http://edward2/wad/users.php***):

```
$result = $conn->query("SELECT * FROM ht_users WHERE username='choose a username
$row = $result->fetch();
```

- Now try adding an ***if*** statement to check that the money or more, e.g:

```
if($row["balance"] >= 0.79)
{
    // .. increase the number of downloads ..
}
```

  Only increase the downloads if the balance is more than 0.79. Otherwise, indicate an error to the client using an ***echo*** statement. Here are two things you can echo to indicate this error:

```
Not enough funds in the account!
```

  or

```
INSUFFICIENT_FUNDS
```

  Given that this is a ***web service***, which would be better? Why?
- Also decrease the balance by 0.79 for that user.

Sending a POST request using cURL

So far we have seen how to connect to a web service using cURL to send a GET request, however what about a POST request? We can do this: it's a little more complex, but still relatively straightforward.

Rather than adding the data to the end of the URL using a query string, you set the ***CURLOPT_POSTFIELDS*** option with the data you want to POST as an associative array. So for example:

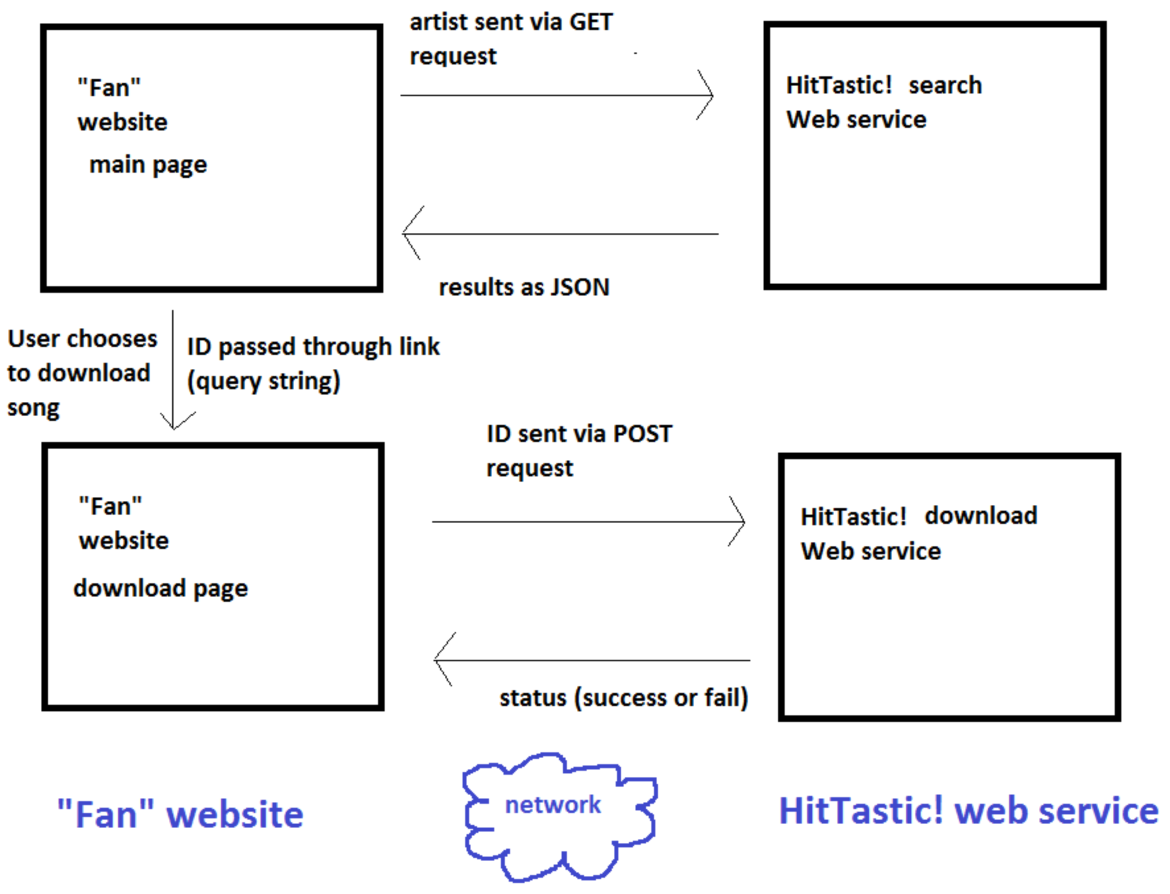
```
$connection = curl_init();
curl_setopt($connection, CURLOPT_URL, "http://www.solentairways.com/newflight.php");
$dataToPost = array
    ("origin" => "London" ,
     "destination" => "New York",
     "aircraft" => "Boeing 777",
     "flightnumber" => "SA123");
curl_setopt($connection,CURLOPT_RETURNTRANSFER,1);
curl_setopt($connection,CURLOPT_POSTFIELDS,$dataToPost);
$response = curl_exec($connection);
curl_close($connection);
```

would send the specified data describing a new flight to the ***newflight.php*** script. ***newflight.php*** could then read the various fields with ***$_POST["origin"]***, ***$_POST["destination"]*** and so on, and add them to the database.

Standard Exercise 2

Return to your "fan" website from last time. On the "fan" website, allow the user the ability to download each song. ***This should be done by connecting to your HitTastic! download web service, which you developed above***, so that even though the user is interacting with the "fan" site, the music is actually being downloaded from HitTastic!

The user should be able to click on a "Download" link on the "fan" site, and this should then run a new PHP script on the "fan" site which makes a cURL POST request to the download web service, above. This scenario is illustrated by the diagram below:


artist sent via GET request
"Fan" website main page
HitTastic! search Web service
results as JSON
User chooses to download song
ID passed through link (query string)
ID sent via POST request
"Fan" website download page
HitTastic! download Web service
status (success or fail)
"Fan" website
network
HitTastic! web service

Do this as follows:

- Add a "download" link beside each song. This should link to a new script, ***clientdownload.php***, and pass the ID across. Since the ID is a member of an array of associative arrays (a 2D array) you cannot embed it directly inside an echo statement. A simple way of dealing with this is to put the ID from the JSON in a variable and then put that in the link, e.g (not all code given, you have to complete the highlighted parts yourself!)

```
$songid = the song ID from the JSON;
echo "<a href='clientdownload.php?id=the variable containing the ID'>complete th
```

- Write ***clientdownload.php***. **This is part of the "fan" site, and should therefore have the same CSS and look and feel as the main page of the "fan" site.** Write some PHP code in this script to send a cURL POST request to your download web service, passing the ID across.
- Test the result of the download web service, and display something appropriate to the user.

Advanced exercise

Something to do if you finish everything else. Write a HitTastic! web service which outputs the top 40 downloads as JSON, with no.1 being the most downloaded track, using an appropriate SQL SELECT statement. Do this for all artists by default, or for a particular artist if a GET parameter of "artist" was specified, e.g:

```
if(isset($_GET["artist"]))
```

On your "fan" website use this web service to display the top downloads for your chosen artist.