

Adding Device Location to a Database

v1.0

By GoNorthWest

4 June 2012

If you think about it, there are any number of app you could make where it would be very advantageous to capture a device's location, submit it to a database (perhaps with some comments), and then extract those points to a multiple location map. Maybe you want to make an app that captures all the best pizza locations in the country. Or maybe donuts...or burgers...or...back in a second...I'm hungry all of a sudden!

In order to make this magic happen, you need to do a few things:

1. Capture the current location of the device.
2. Provide the user a form to validate their location, and to allow them to enter information like an email address, or comments.
3. Process the form information, inserting the data into a database.
4. Extract the information and present it in a useable format...like a multiple location map.

This seems much harder than it actually is. If you know anything about PHP, MySQL, and HTML, than what follows will be easy as pie! If you are a complete novice, you can still make this work by following the tutorial I'm about to present! And, as always, if you have any questions, just email me!

I think the best way to attack this problem is step-by-step. Here's what we're going to do:

1. Create the database table.
2. Create an HTML/PHP form that captures the user location and extra data.
3. Link the HTML/PHP form to your app.
4. Create a PHP script that extracts the data from the database, creates a JSON text file, and provides it to the app.

Easy, right?! OK...let's roll!

Create the Database and Table

The first thing you need to do is create a table in a database to hold your data. This example uses MySQL, and assumes that you already have the database created. What follows is the information you need to create the actual table. You can name the table whatever you want, but for purposes of this tutorial, I'm going to call it 'theguide.' Here is the MySQL syntax you need:

```
CREATE TABLE IF NOT EXISTS `theguide` (  
  `ID` int(11) NOT NULL auto_increment,  
  `SubmitterEmail` text NOT NULL,  
  `latitude` text NOT NULL,  
  `longitude` text NOT NULL,  
  `Description` text NOT NULL,  
  `Status` text NOT NULL,  
  PRIMARY KEY (`ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=29 ;
```

If you don't understand the above, try and do a bit of research in Google, as a huge discussion on databases is beyond the scope of this tutorial. In a nutshell, though, the commands above will create a table that holds an email address; the latitude and longitude of the device; a description of the location, and a status of the entry (hidden or visible...which you can use to control which ones you enable users to see..a good review tactic).

Create an HTML form that captures the device location and extra data.

If you understand HTML and forms, this will be easy. It does have a bit of PHP mixed in with it, but you don't really need to understand it in order to make it work. Just follow the example code provided, and you'll get the information above.

Begin by creating a file on your web server called location.php (or something similar, so long as you remember the name). Then grab the code from the following text file and paste it into that location.php file you just created:

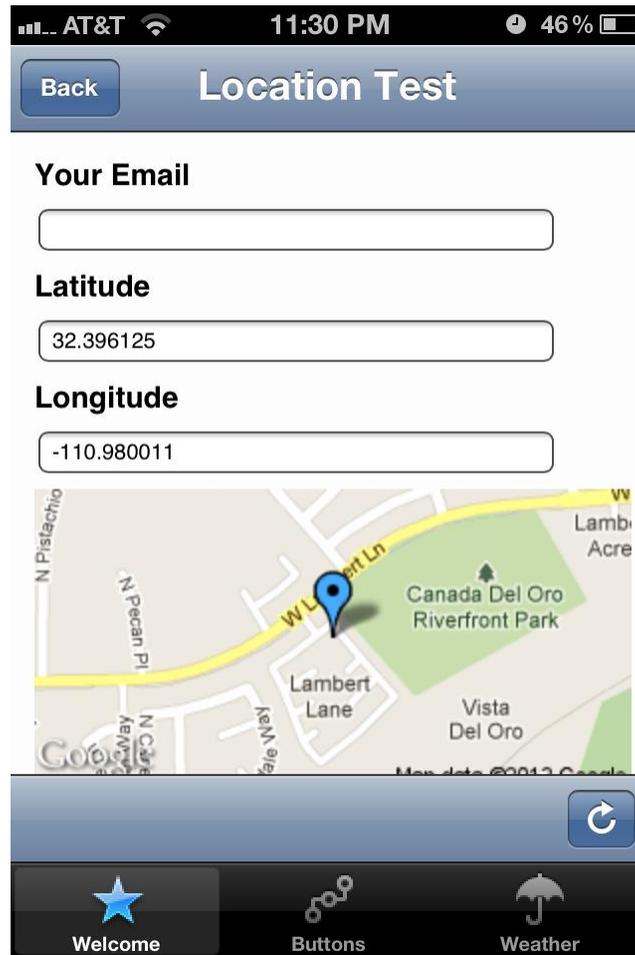
https://dl.dropbox.com/u/4993711/tutorial/location_php.txt

The only real interesting part of this code is the following. Everything else is just a form for collecting user input.

```
<?php
$latitude = $_GET['latitude'];
$longitude = $_GET['longitude'];
$db = mysql_connect("localhost", "yourusername", "yourpassword");
mysql_select_db("yourdatabasename",$db);
if (isset($_POST['submit'])) {
    $SubmitterEmail = $_POST["SubmitterEmail"];
    $latitude = $_POST["latitude"];
    $longitude = $_POST["longitude"];
    $Status = $_POST["Status"];
    $Description = $_POST["Description"];
    $sql = "INSERT INTO theguide (SubmitterEmail, latitude, longitude,
Description, Status) VALUES
('$SubmitterEmail','$latitude','$longitude','$Description','$Status')";
    $result = mysql_query($sql);
    header("location:http://www.google.com/");
}else{
    $error = $resp->error;
}
?>
```

This code is pretty simple in what it does, and is probably not optimally written. But, it gets the job done. Essentially, when the user loads the screen in your code will be the latitude and longitude of the device that is running the app. You'll be able to verify that because a map will be shown with your current location. It assigns those values to `$latitude` and `$longitude`, when then later get posted into your database when the user hits Submit. Also...the google.com URL you see there can be changed to anything you want...like a personal response page. Don't worry about if you understand this or

not...just go with the fact that it works! And here's proof that it works, with a screen shot from my test app!



Link the HTML/PHP form to your app

At this point we've created the table to hold our data, and we also created a form to capture the data from the user and submit it into the database. Now we need to create a screen in buzztouch that provides the user the appropriate screen.

Because this form is HTML/PHP, and because PHP is a server side scripting language, we need to create a **Custom URL** screen, and give it the following URL:

[http://www.yourdomain.com/location.php?latitude=\[deviceLatitude\]&longitude=\[deviceLongitude\]](http://www.yourdomain.com/location.php?latitude=[deviceLatitude]&longitude=[deviceLongitude])

This link tells the screen to head out to yourdomain.com, grab the location.php script, feed it the device latitude and longitude, and then return a form to your screen that the user can fill in with more information. Then the user can hit Submit and all their info will automatically be inserted into the database!

Create a PHP script that extracts the data from the database, creates a JSON text file, and provides it to the app

Now that we've put information into the database, and are hopefully collecting a steady stream of great data from our users, we need a way to get it out and into our app. The way we're going to do that is to call a script that will extract info from the database and put it into a JSON format that the multiple location map screen in buzztouch will understand. You can grab the script from here, in text format, and paste it into a file called, for instance, location-list.php.

<https://dl.dropbox.com/u/4993711/tutorial/location-list.txt>

And here's what the script itself looks like:

```
<?php
$db = mysql_connect("localhost", "yourusername", "yourpassword");
mysql_select_db("yourdatabasename",$db);
$result = mysql_query("SELECT ID, Latitude, Longitude, Description from theguide",$db);
$num = mysql_num_rows($result);
$i = 1;
print "{\"childItems\":[";
if ($myrow = mysql_fetch_array($result)) do {
    printf("{\"itemId\":\"%s\",", $myrow["ID"]);
    print "\"itemType\":\"BT_MapLocation\",";
    printf("\"latitude\":\"%s\",", $myrow["Latitude"]);
    printf("\"longitude\":\"%s\",", $myrow["Longitude"]);
    printf("\"title\":\"%s\"}", $myrow["Description"]);
    if ($num > $i) printf(",");
    $i++;
} while ($myrow = mysql_fetch_array($result));
else {
    echo "Sorry, nothing there to see!";
}
print "]}";
```

?>

If you were to run this script against your database, here is a snippet of what the output would look like:

```
{"childItems":[  
  {"itemId":"19",  
   "itemType":"BT_MapLocation",  
   "latitude":"45.567015",  
   "longitude":"-122.606872",  
   "title":"House in Portland"},  
  
  {"itemId":"20",  
   "itemType":"BT_MapLocation",  
   "latitude":"44.061006",  
   "longitude":"-123.096796",  
   "title":"A Sweet Location"}  
]}
```

Now you need to create a **Location Map** screen. Then go into **Advanced Properties, Screen Data URL**, and paste in a link to the script you created above. It should look something like this:

<http://www.yourdomain.com/location-list.php>

And that's really it! When the user opens up the Location Map screen, it will go out to that script, run it, return the correct JSON for the screen, and then the points in the database will be displayed.

This all seems a bit more complicated than it is, and I fear I haven't done a stellar job explaining it. So, please, give it a shot, and if you run into any problems, let me know!

Enjoy!

Comments? Post them in the forum or email me at MrkFleming@gmail.com.

Revision Log			
v1.0	06/04/2012	Initial release of document.	GoNorthWest