

Ok. First things first, you need a config.xml file that looks like this:

```
<config-file platform="android" parent="/manifest/application" mode="merge">
  <plugin name="cordova-plugin-whitelist" source="npm"/>
  <!-- Don't block any requests -->
  <access origin="*" />
  <!-- A wildcard can be used to whitelist the entire network,
    over HTTP and HTTPS.
    *NOT RECOMMENDED* -->
  <allow-navigation href="*" />
</config-file>
```

(edited)

You then need an index.html file that redirects to the proper start html file like this:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <script src="https://code.jquery.com/jquery-3.3.1.min.js" integrity="sha256-
FgpCb/KJQlLnFou91ta32o/NMZxltwRo8QtmkMRdAu8=" crossorigin="anonymous"></script>
  <meta charset='UTF-8'>
  <link
href='https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/css/bootstrap.min.css'
rel='stylesheet'>
  <script
src='https://stackpath.bootstrapcdn.com/bootstrap/4.1.2/js/bootstrap.min.js'></script>
  <script src = "globalFunctions.js"></script>
  <style>label {width: 5em;}</style>
</head>
<body onload="window.location.href = 'customer.html' ">
</body>
</html>
```

the line `<body onload="window.location.href = 'customer.html' "> </body>` is the important part

except you might replace 'customer.html' with whatever file you want to redirect too

Now that those are put in place, are you making sure you are loading all html code from the .php files with XMLHttpRequests or ajax? php files will not render in PhoneGap

Corser goes over how to accomplish that in his echo360 videos

After that, you have to make sure you are distinguishing between GET and POST when making a request. Example php code:

```
switch ($_SERVER['REQUEST_METHOD']) {
    case 'GET':
        break;
    case 'POST':
        break;
}
```

If you have your html code at the bottom of the html file and / or are always echoing, that will be the first response. It has to be wrapped in a switch case statement like that.

Correct code example:

```
<?php
session_start();
require "database.php";
```


In the second example, I put my html code at the bottom of the html file and is perfectly valid, however because I didn't apply any php logic to it, it will always be the response I get when making a request on this php file.

Also, XMLHttpRequest doesn't seem to handle redirects very well so instead I encode my responses to a JSON format like this when I want to redirect:

```
header('Content-Type: application/json');
echo json_encode(['location'=>'login.html?errorMessage=Invalid Username or Password!']);
```

I then handle the response like this to see if it was a redirect:

```
function loadDoc(url, method, form) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var redirect = false;
            try {
                var response = JSON.parse(this.responseText);
                if (response.location) {
                    window.location.href = response.location;
                    redirect = true;
                }
            } catch {

            }

            if (!redirect)
                document.getElementById("htmlDiv").innerHTML = this.responseText;
        }
    };

    xhttp.open(method, "http://csis.svsu.edu/~tclange/cis355/Prog05/" + url, true);

    if (form != null && form != undefined){
        xhttp.send(new FormData(form));
    } else {
        xhttp.send();
    }
    return false;
}
```

As you can see, the variable `location` gets mapped to a JSON object. I then simply check to see if that variable exists in the response with

```
if (response.location) {
    window.location.href = response.location;
    redirect = true;
}
```

If `response.location` doesn't exist, then I know I'm not trying to redirect to a different html / php file and am instead populating my html page with the html code passed in during the request.

You also have to make sure to point your hosting location of the php files. If you simply try to call `xhttp.open(method, "customer.php", true);` it will try to open a local php file which isn't valid in PhoneGap. Instead you have to point it to the server location. Example:

```
xhttp.open(method, "http://csis.svsu.edu/~tclange/cis355/Prog05/" + url, true);
```

You then also have to make sure to allow cross origin requests in your php files on your server as well by putting this at the top of each of your php scripts you are calling:

```
header('Access-Control-Allow-Origin: *');
```

if you don't do that, your requests will be blocked by CSIS or 000Webhost or wherever else you might be hosting.

Also don't forget to use PhoneGap cloud build here: <https://build.phonegap.com/apps>

You will have to sign in with Google first but after that it will allow you to build apps.