

11. JSON JSONP.

11.1. JSONP

JSONP is a method for sending JSON data without worrying about cross-domain issues.

JSONP does not use the XMLHttpRequest object.

JSONP uses the <script> tag instead.

11.2. JSONP Intro

JSONP stands for JSON with Padding.

Requesting a file from another domain can cause problems, due to cross-domain policy.

Requesting an external script from another domain does not have this problem.

JSONP uses this advantage, and request files using the script tag instead of the XMLHttpRequest object.

Example:

```
<script src="demo_jsonp.php">
```

11.3. The Server File

The file on the server wraps the result inside a function call:

Example:

```
<?php
$myJSON = '{ "name":"John", "age":30, "city":"New York" }';

echo "myFunc (". $myJSON . ")";
?>
```

The result returns a call to a function named "myFunc" with the JSON data as a parameter.

Make sure that the function exists on the client.

11.4. The JavaScript function

The function named "myFunc" is located on the client, and ready to handle JSON data:

Example:

```
function myFunc(myObj) {  
    document.getElementById("demo").innerHTML = myObj.name;  
}
```

11.5. Creating a Dynamic Script Tag

The example above will execute the "myFunc" function when the page is loading, based on where you put the script tag, which is not very satisfying.

The script tag should only be created when needed:

Example: Create and insert the <script> tag when a button is clicked:

```
function clickButton() {  
    var s = document.createElement("script");  
    s.src = "demo_jsonp.php";  
    document.body.appendChild(s);  
}
```

11.6. Dynamic JSONP Result

The examples above are still very static.

Make the example dynamic by sending JSON to the php file, and let the php file return a JSON object based on the information it gets.

Example: PHP file

```
<?php
header("Content-Type: application/json; charset=UTF-8");
$obj = json_decode($_GET["x"], false);

$conn = new mysqli("myServer", "myUser", "myPassword", "Northwind");
$result = $conn->query("SELECT name FROM ".$obj->$table." LIMIT ".$obj->$limit);
$outp = array();
$outp = $result->fetch_all(MYSQLI_ASSOC);

echo "myFunc(".json_encode($outp).")";
?>
```

PHP File explained:

- Convert the request into an object, using the PHP function `json_decode()`.
- Access the database, and fill an array with the requested data.
- Add the array to an object.
- Convert the array into JSON using the `json_encode()` function.
- Wrap "myFunc()" around the return object.

Example: JavaScript Example. The "myFunc" function will be called from the php file:

```
function clickButton() {
    var obj, s
    obj = { "table": "products", "limit": 10 };
    s = document.createElement("script");
    s.src = "jsonp_demo_db.php?x=" + JSON.stringify(obj);
    document.body.appendChild(s);
}

function myFunc(myObj) {
    var x, txt = "";
    for (x in myObj) {
        txt += myObj[x].name + "<br>";
    }
    document.getElementById("demo").innerHTML = txt;
}
```

11.7. Callback Function

When you have no control over the server file, how do you get the server file to call the correct function?

Sometimes the server file offers a callback function as a parameter:

Example: The php file will call the function you pass as a callback parameter:

```
function clickButton() {  
    var s = document.createElement("script");  
    s.src = "jsonp_demo_db.php?callback=myDisplayFunction";  
    document.body.appendChild(s);  
}
```