WORK

Ads via Carbon

Chat, made for business. Video

chat, file sharing & IM. \$0/unlimited users.

SELECTED POSTS

Need for Speed 2: Improving Front-End Performance Faster Font Loading with Font Events Git Command-Line Shortcuts

Responsible Social Share Links

POST TOPICS Development Tutorial Design Journal Review

SAY HEY!

Drop me an email

Start a conversation @jonsuh

Stay up-to-date with all the blog posts via

You can also subscribe to my newsletter to receive occasional emails from me.

RSS or the Archives.

SUBSCRIBE

Your email address Subscribe

Convert and Loop through JSON with PHP and JavaScript Arrays/Objects September 30, 2014 by Jonathan Suh

If you're working with JSON (JavaScript Object Notation) and either need

to convert a JSON string to array or object and loop through it or viceversa, take an array or object and convert it to a JSON string to return, both can be done in PHP or JavaScript. I broke up this post into three sections:

Working with PHP

- Working with JavaScript Working with both PHP and JavaScript
- **Convert JSON String to PHP Array or Object** PHP >= 5.2.0 features a function, json_decode, that decodes a JSON

string into a PHP variable. By default it returns an object. The second parameter accepts a boolean that when set as true, tells it to return the

\$someArray = json_decode(\$someJSON, true);

objects as associative arrays. You can learn more about the json decode function from PHP's documentation. <?php // JSON string \$someJSON = '[{"name":"Jonathan Suh", "gender": "male"}, { "name" // Convert JSON string to Array

```
echo $someArray[0]["name"]; // Access Array data
  // Convert JSON string to Object
  $someObject = json decode($someJSON);
  echo $someObject[0]->name; // Access Object data
Loop through PHP Array or Object
Loop through a PHP array or object with a foreach loop.
```

// Loop through Object \$someObject = ...; // Replace ... with your PHP Object

// Loop through Array

<?php

<?php

// Array

foreach (\$someArray as \$key => \$value) {

\$someArray = ...; // Replace ... with your PHP Array

echo \$value["name"] . ", " . \$value["gender"] . "
";

```
foreach($someObject as $key => $value) {
     echo $value->name . ", " . $value->gender . "<br>";
Note the differences in accessing the values of an array vs an object.
Convert PHP Array or Object to JSON String
PHP also features a json_encode function to convert an array or object
into a string. Read more about the json encode function from PHP's
documentation.
```

\$someArray = ["name" => "Jonathan Suh",

"gender" => "male"

"name" => "William Philbin",

"gender" => "male"

"name" => "Allison McKinnery", "gender" => "female"]; // Convert Array to JSON String \$someJSON = json encode(\$someArray); echo \$someJSON;

```
Note that I'm using the short array syntax that's featured in PHP 5.4+.
  <?php
    $array = array(
      "foo" => "bar",
      "bar" => "foo"
    );
    // as of PHP 5.4
    array = [
      "foo" => "bar",
      "bar" => "foo"
    ];
```

not support it (i.e. <= IE 7. More information at caniuse.com). jQuery 1.x has a \$.parseJSON() method that should fill in the gaps for

Convert JSON String to JavaScript Object

// Convert JSON String to JavaScript Object

var JSONObject = JSON.parse(JSONString);

string and returns an object.

JSON-js library as a polyfill.

console.log(JSONObject);

<script>

</script>

</script>

JSON string.

<script>

},

},

];

JavaScript has a built-in JSON.parse() method that parses a JSON

alert(JSONObject[0]["name"]); // Access Object data

JSON.parse() is very well-supported, but there are browsers that do

var JSONString = '[{"name":"Jonathan Suh", "gender": "male"}, {"

// Dump all data of the Object

<script> // Convert JSON String to JavaScript Object with jQuery var JSONString = "..."; // Replace ... with your JSON String var JSONObject = \$.parseJSON(JSONString);

alert(JSONObject[0]["name"]); // Access Object data

those browsers if you're needing to support them. You can also use the

```
You can then loop through a JavaScript object using a for in loop.
 <script>
   // Loop through Object
   var JSONObject = ...; // Replace ... with your JavaScript Obje
   for (var key in JSONObject) {
     if (JSONObject.hasOwnProperty(key)) {
       console.log(JSONObject[key]["name"] + ", " + JSONObject[ke
 </script>
```

var JSONObject = ["name": "Jonathan Suh", "gender": "male"

"name": "William Philbin",

"name": "Allison McKinnery",

"gender": "male"

"gender": "female"

JSON.stringify as well.

on gender, like this:

--Select--

</select>

</thead>

\$.ajax({

},

data: {

}

type: "POST",

url: "response.php",

var peopleHTML = "";

dataType: "json",

<script>

Name Gender

Convert JavaScript Object to JSON String

JavaScript has a JSON.stringify method to convert a value into a

Loop through JavaScript Object

var JSONString = JSON.stringify(JSONObject); alert(JSONString); </script>

Like JSON.parse, JSON.stringify is not supported in dinosaur

browsers like <= IE 7. You can use the JSON-js library to polyfill

You can combine the methods above to create powerful, dynamic

implementations on your website or application.

```
Let's say you want to get information from a database, safely return the
data as JSON, and loop through it dynamically, you can do so with a bit of
PHP and JavaScript with Ajax.
Dynamically Get JSON via Ajax and Loop Through
JSON
Let's assume your database structure looks like the following:
                               gender
   id
         Jonathan Suh
                               male
         William Philbin
                               male
                               female
                               female
                               female
```

And you want to dynamically get a list of people from the database based

<thead> Name Gender

Let's start with the front-end file index.html that'll have a select

the script to handle the Ajax. The JavaScript is written in jQuery.

<select id="gender" name="gender">

<option value="male">Male</option>

\$("#gender").on("change", function() {

"gender": \$("#gender").val()

success: function(JSONObject) {

for (var key in JSONObject) {

peopleHTML += "";

peopleHTML += "";

<option value="female">Female</option>

dropdown with genders to select from, a table to display the results, and

<script src="http://code.jquery.com/jquery-1.11.1.min.js"></scri</pre>

// Loop through Object and create peopleHTML

peopleHTML += "" + JSONObject[key]["name"] + "<,</pre> peopleHTML += "" + JSONObject[key]["gender"] +

if (JSONObject.hasOwnProperty(key)) {

```
// Replace table's tbody html with peopleHTML
       $("#people tbody").html(peopleHTML);
   });
  });
 </script>
Now let's create a response.php file to handle the back-end logic of
getting the information from the database and returning the results as a
JSON string.
   // File: response.php
   // Get POST gender value
   $gender = $ POST["gender"];
    // Connect to the database
   // replace the parameters with your proper credentials
   $connection = mysqli_connect("localhost", "username", "passwor
    // Query to run
   $query = mysqli_query($connection,
             "SELECT * FROM people WHERE gender = '" . $gender .
   // Create empty array to hold query results
   $someArray = [];
   // Loop through query and push results into $someArray;
   while ($row = mysqli_fetch_assoc($query)) {
      array_push($someArray, [
       'name' => $row['name'],
        'gender' => $row['gender']
     ]);
   // Convert the Array to a JSON String and echo it
   $someJSON = json_encode($someArray);
   echo $someJSON;
To get a more in-depth and better example of PHP-JSON-
JavaScript/jQuery-Ajax interaction, read my jQuery Ajax Call to PHP Script
```

← Previous Post Next Post →

jQuery Ajax Call to PHP Script with JSON Return

Jonathan Suh

Detect the End of CSS Animations and Transitions with JavaScript Retina Image Replacement Better Scroll-To Anchor Links

Responsible Social Share Links

RELATED POSTS

with JSON Return post.

SHARE 💆

I'm a multidisciplinary developer & designer. I'm currently with the folks at Juice Interactive, and I've worked with well-known brands and agencies, which include Allstate, Brian Hoff Design, Leo Burnett, and OLSON. More about me.

Blog 3

Contact

Follow @jonsuh

Jonathan Suh

About

Work