**Insertar un archivo JSON en una base de datos utilizando PHP**

**Cómo insertar un archivo JSON en una base de datos utilizando PHP**

En este artículo en primer lugar voy a hacer una breve introducción sobre JSON y luego explicar cómo insertar un archivo JSON en una base de datos utilizando PHP y luego hacer el proceso a la inversa; es decir, recuperar nuevamente los datos de la base de datos y almacenarlos en un archivo JSON usando PHP. Vamos a empezar con la introducción a JSON:

**JSON**

JSON básicamente significa JavaScript Object Notation. JSON es bastante simple, fácil y ligero y sirve para el intercambio y almacenamiento de datos desde y hacia la base de datos. Es una alternativa al conocido XML (Extensible Markup Language). Los datos en formato JSON se almacenan en el archivo JSON con la extensión  “**.json**”. Los datos presentes en el archivo JSON son compatibles con todos los navegadores modernos actuales.

**Sintaxis JSON**

A continuación mostramos la sintaxis JSON:

**Listado 1** . studJson.json

{

"StdID": "0106",

"StdData": {

"StdName": "Mirwaise",

"StdAge": "23",

"StdGender": "Male",

"StdNo": "12345",

"StdAddress": {

"StdStreet": "786 Street",

"StdCity": "Riad",

"StdCountry": "Arabia Saudita",

"StdPostal": "98765"

}

},

"STDEDU": {

"StdDept": "Computer Science",

"StdSemester": "8",

"StdMajor": "Programación Web"

}

}

El código JSON anterior se utilizará como base para el resto de este artículo. El siguiente paso es mostrar cómo insertar un archivo JSON en una base de datos utilizando PHP. Para este ejemplo voy a usar una base de datos MySQL.

**Insertar un archivo JSON en una base de datos utilizando PHP**

Para insertar un archivo JSON en una base de datos utilizando PHP se necesitan algunos pasos con el fin de hacer más fácil esta tarea.

**Paso 1: Creación de base de datos y conexión de base de datos (MySQL) con PHP**

En primer lugar crearemos una base de datos (“**jsondb**” en mi caso) y en la misma crearemos una tabla llamada (“**stdtable**”). El número de columnas no debe ser inferior a los campos requeridos en el archivo JSON. En mi caso, he creado una tabla con los campos necesarios, como se muestra a continuación:

**Figura 1** . Tabla stdtable junto con las columnas.

Una vez que hayas creado la tabla base con los campos necesarios, el siguiente paso es conectarse a la base de datos desde el archivo PHP. Para ello, vamos a crear un archivo PHP con los siguientes datos:

**Listado 2** .jsonCon.php

<?php

//connect and select the database

$connect = mysql\_connect("localhost","root","") or die('Database Not Connected. Please Fix the Issue! ' . mysql\_error());

mysql\_select\_db("jsondb", $connect);

// get the contents of the JSON file

$jsonCont = file\_get\_contents('studJson.json');

//decode JSON data to PHP array

$content = json\_decode($jsonCont, true);

//Fetch the details of Student

$std\_id = $content['stdID'];

$std\_name = $content['stdData']['stdName'];

$std\_age = $content['stdData']['stdAge'];

$std\_gender = $content['stdData']['stdGender'];

$std\_no = $content['stdData']['stdNo'];

$std\_street = $content['stdData']['stdAddress']['stdStreet'];

$std\_city = $content['stdData']['stdAddress']['stdCity'];

$std\_country = $content['stdData']['stdAddress']['stdCountry'];

$std\_postal = $content['stdData']['stdAddress']['stdPostal'];

$std\_dept = $content['stdEdu']['stdDept'];

$std\_sem = $content['stdEdu']['stdSemester'];

$std\_major = $content['stdEdu']['stdMajor'];

//Insert the fetched Data into Database

$query = "INSERT INTO stdtable(std\_id, std\_name, std\_age, std\_gender, std\_num, std\_street, std\_city, std\_country, std\_postal, std\_dept, std\_semstr, std\_major)

VALUES('$std\_id', '$std\_name', '$std\_age', '$std\_gender', '$std\_no', '$std\_street', '$std\_city', '$std\_country', '$std\_postal', '$std\_dept', '$std\_sem', '$std\_major')";

if(!mysql\_query($query,$connect)) {

die('Error : Query Not Executed. Please Fix the Issue! ' . mysql\_error());

} else{

echo "Data Inserted Successully!!!";

}

?>

En el código anterior, lo primero que hacemos es establecer una conexión de base de datos mediante la función PHP “ **mysql\_connect ()** ”. A continuación, después de la conexión, seleccionamos la base de datos usando la función de PHP “ **mysql\_select\_db ()** ”.

### Paso 2: Obtener el contenido del archivo JSON

Ahora, para obtener el contenido del archivo JSON, utilizamos la función de PHP “ **file\_get\_contents** () ”. Esta función lee el contenido del archivo dado en una cadena. En el código anterior, el “ **studJson.json** ” representa el archivo JSON cuyos datos se van a leer.

### Paso 3: decodificar datos JSON hacia una matriz asociativa en PHP

Una vez que se lee y está disponible en forma de cadena los datos del archivo JSON, entonces se convierten los datos JSON en una matriz (o arreglo). Para ello, utilizamos la función PHP “ **json\_decode ()** ”. Esta función convierte los datos presentes en formato en una matriz asociativa en PHP. En el código anterior, guardamos en la variable “ **$jsonCont**” los contenidos de la cadena del archivo JSON.

### Paso 4: Obtener los valores decodificados

El siguiente paso es buscar los valores decodificados y almacenarlos en las variables de PHP. Esto se puede hacer buscando uno por uno los valores y almacenándolos en las diferentes variables.

### Paso 5: Insertar los datos obtenidos en la base de datos

Por último, los datos JSON se insertan en la base de datos mediante una instrucción de MySQL. Es importante tener en cuenta que los nombres de las columnas de la tabla en la base de datos deben ser escritos correctamente, de otra manera no se insertarán los datos en la base de datos.

Al ejecutar el código anterior obtendrás el siguiente resultado:

Insertar archivo JSON en base de datos

**La Figura 2**. Inserción del archivo JSON en una base de datos.

Actualiza tu base de datos y verás que los datos se insertaron correctamente. La salida de la base de datos será la siguiente:

**Figura 3** . Resultado de los datos insertados en la base de datos.

Se puede ver en la imagen anterior que los datos se insertan adecuadamente en las columnas correspondientes. Si deseas añadir más datos en las columnas de la tabla, cambia los datos en el archivo JSON o agrega otro archivo JSON con otro nombre y ejecuta el código de nuevo. Vamos a insertar otro archivo JSON. Simplemente añadimos otro archivo JSON ( **studJson2.json** ) en la función **file\_get\_contents**( ‘studJson2.json’). Los datos que utilizaremos en “ **studJson2.json”** son los siguientes:

**Listado 3** . studJson2.json

{

"StdID": "0110",

"StdData": {

"StdName": "Nisar",

"StdAge": "25",

"StdGender": "Male",

"StdNo": "78600",

"StdAddress": {

"StdStreet": "123 Street",

"StdCity": "Swat",

"StdCountry": "Canadá",

"StdPostal": "55511"

}

},

"STDEDU": {

"StdDept": "Ingeniería Informática",

"StdSemester": "7",

"StdMajor": "Ingeniería de Software"

}

}

Ejecuta de nuevo y ahora obtendrás una salida de la siguiente manera:El siguiente paso es a la inversa de lo anterior; es decir, recuperar los datos de la base de datos y almacenarlos en el archivo JSON a través de PHP.

## Recuperar los datos de la base de datos y almacenarlos en un archivo JSON a través de PHP

En esta parte, vamos a extraer los datos que se almacenaron en la base de datos y mostrarlos en un archivo con formato JSON. Este proceso es uno de los más utilizados por los desarrolladores web hoy en día. Para comenzar, debes tener una base de datos como explicamos anteriormente. Este proceso también necesita algunos pasos a tener en cuenta. Vamos a empezar con la explicación de cada paso:

### Paso 1: Conexión a la base de datos con PHP

Vamos a utilizar la misma base de datos y la misma conexión de base de datos como hicimos anteriormente:

**Listado 4** .jsonRev.php

<?php

// Paso No. 1: Conectar base de datos con PHP

$connect = mysql\_connect("localhost","root","") or die('Database Not Connected. Please Fix the Issue! ' . mysql\_error());

mysql\_select\_db("jsondb", $connect);

?>

### Paso 2: Conexión a la base de datos

Una vez establecida la conexión, entonces hay que extraer los datos presentes en la tabla de base de datos. Para esto, utilizaremos la función “mysql\_query()”:

**Listado 5.** jsonRev.php

<?php

//Step No. 2: Extracting data from database

$query = "SELECT \* FROM stdtable";

$res = mysql\_query($query,$connect) or die("Query Not Executed " . mysql\_error($connect));

?>

**Paso 3:** Extraer los datos de la base de datos

Ahora debemos guardar los datos en un arreglo en PHP. Para ello, creamos una variable y asignamos una matriz vacía a esa variable. Esta matriz vacía se utilizará para obtener los registros obtenidos como resultado de la consulta MySQL. En PHP utilizaremos la función “mysql\_fetch\_assoc()” para este propósito.

**Listado 6.** jsonRev.php

<?php

//Step No. 3: Putting the fetched data in Arrays

$data\_array = array();

while($rows =mysql\_fetch\_assoc($res)) {

$data\_array[] = $rows;

}

?>

**Paso 4:** Guardar datos en matriz asociativa

Una vez que tengas los datos en una matriz, entonces debemos codificarlo en formato JSON. Para ello, se utiliza la función de PHP “json\_encode()”. Esta función toma la matriz como un argumento y devuelve los datos de la matriz en cadenas de JSON.

### Paso 5: Escribir los datos en un archivo JSON

Finalmente, después de tener los datos en formato JSON, hay que escribir los mismos en un archivo JSON. Para ello, utilizaremos la función “fopen()” para abrir el archivo y a continuación, la función “fwrite()”, la cual toma dos argumentos; una con el archivo devuelto por “fopen” y el segundo argumento para la codificación de los datos presentes en la base de datos.

**Listado 7.** jsonRev.php

<?php

//Step No. 4 and 5: Encoding Array into JSON + Writing data to JSON file

$fp = fopen('studRecords.json', 'w');

//fwrite($fp, json\_encode($data\_array));

if(!fwrite($fp, json\_encode($data\_array))) {

die('Error : File Not Opened. ' . mysql\_error());

} else{

echo "Data Retrieved Successully!!!";

}

fclose($fp);

?>

Aquí está el código completo:

**Listado 8.** jsonRev.php

<?php

//Step No. 1: Connect database with PHP

$connect = mysql\_connect("localhost","root","") or die('Database Not Connected. Please Fix the Issue! ' . mysql\_error());

mysql\_select\_db("jsondb", $connect);

//Step No. 2: Extracting data from database

$query = "SELECT \* FROM stdtable";

$res = mysql\_query($query,$connect) or die("Query Not Executed " . mysql\_error($connect));

//Step No. 3: Putting the fetched data in Arrays

$data\_array = array();

while($rows =mysql\_fetch\_assoc($res)) {

$data\_array[] = $rows;

}

//Step No. 4 and 5: Encoding Array into JSON + Writing data to JSON file $fp = fopen('studRecords.json', 'w');

//fwrite($fp, json\_encode($data\_array));

if(!fwrite($fp, json\_encode($data\_array))) {

die('Error : File Not Opened. ' . mysql\_error());

} else{

echo "Data Retrieved Successully!!!";

}

fclose($fp);

mysql\_close($connect);

?>

La salida para el código anterior será como sigue:

Después de ejecutar el código anterior, se verá que el archivo “ **studRecords.json** ” se creará automáticamente si no había sido creado previamente, y contendrá los datos recuperados de la base de datos en formato JSON.

## Conclusión

El formato JSON juega un papel de vital importancia en el desarrollo web actual. En estos días JSON tomó el lugar que en algún momento ocupó XML. En este artículo vimos cómo insertar un archivo JSON en una base de datos utilizando PHP. Para ello, en primer lugar, vimos una breve descripción del formato JSON. A continuación, las diferentes etapas para insertar los datos JSON a una base de datos. Por último, hicimos el proceso a la inversa; es decir, recuperamos de datos de la base de datos y los guardamos en un archivo JSON.