


Обучение: Таблица PHP/SQL часть 2

Июн 17 2010Настраиваем PHP/SQL backend

Я предполагаю, что Вы уже знаете немного о PHP / SQL. Я буду придерживаться в этой обучающей программе определенных базовых элементов. Итак, давайте устанавливать требуемые SQL таблицы! Мы собираемся сделать небольшую американскую Хронологию, поскольку мы будем иметь дело с различными президентами США. Я составил две таблицы, ПРЕЗИДЕНТОВ и ПАРТИЙ, и добавил всех кроме одного президента. (мы добавим последнего позже в этой обучающей программе).

 [Загрузите SQL файл](#) Убедитесь, что Вы загрузили скаченный SQL файл перед продолжением работы, нам понадобится подготовленная база.

- Создание двух таблиц в "учебной" базе используя pdts_usal.sql файл :

```
[me@mysql-server ~]$ mysqlshow
+-----+
| Databases |
+-----+
| information_schema |
| tutorial          |
+-----+
[me@mysql-server ~]$ mysql tutorial < pdts_usa.sql
[me@mysql-server ~]$
```

- Верификация.

```
[tjyang@ibm ~]$ mysql -u root -p myrootpassword
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 328
Server version: 5.0.67 Source distribution
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test       |
| tutorial    |
+-----+
4 rows in set (0.01 sec)
```

```
mysql> connect tutorial;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Connection id: 329
Current database: tutorial
```

```
mysql> show tables;
+-----+
| Tables_in_tutorial |
+-----+
| parties            |
| presidents         |
+-----+
2 rows in set (0.00 sec)
```

mysql>

Теперь убедимся что mysql сервер запущен, работает с нашей "учебной" базой данных и имеет две таблицы, которыми мы можем управлять (запрос, добавьте, удалите, измените) на.

JSON Формат

Прежде, чем переходить к нашему файлу PHP, Вы должны быть в курсе того, что такое формат JSON. Я думаю, что могу просто направить Вас в [Wikipedia](#). JSON - в общих чертах это способ форматировать данные, точно так же, как в XML или CSV... За исключением того, что это обычно намного меньше чем XML.

Пример Меню JSON :

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        { "value": "New", "onclick": "CreateNewDoc()" },
        { "value": "Open", "onclick": "OpenDoc()" },
        { "value": "Close", "onclick": "CloseDoc()" }
      ]
    }
  }
}
```

И тоже самое в XML :

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

Это экономит нам много места, правда? Ну честно говоря, это фактически сказывается при работе с большими массивами... Тем не менее, Ext принимает JSON так же как XML. Теперь, мы сконцентрируемся на JSON, так как большинство примеров, которые Вы будете вероятно видеть, будут использовать этот формат. Давайте иметь в виду, что мы должны будем отправить отформатированный JSON массив обратно к нашему Ext скрипту. Мы позволим нашему файлу PHP делать это через функцию под названием JEncode ().

Файл PHP

Хорошо, дальше идет наш скрипт сервера. Идем дальше...

Во-первых, сервер должен соединиться с базой данных. Мы сделаем очень простую связь:

```
<?php
//////////////////////////////////////////////////
// DATABASE.PHP
//////////////////////////////////////////////////

mysql_connect("localhost", "username", "pwd") or
die("Could not connect: " . mysql_error());

mysql_select_db("db_name");
```

Установите необходимые значения localhost / username / pwd / db_name.

Во-вторых, нам необходимо обработать запросы от Ext. У нас будут Ext запросы для различных задач посредством переменной \$_POST названной 'task'. Так, предполагая, что Ext отправит нам \$_POST ['task'], все что нам необходимо это ответить соответствующим образом.

```
$task = "";
if ( isset($_POST['task'])) {
    $task = $_POST['task']; // Get this from Ext
}

switch($task) {
    case "LISTING": // Give the entire list
        getList();
        break;
    default:
        echo "{failure:true}"; // Simple 1-dim JSON array to tell Ext the request failed.
        break;
}
```

Отлично... Теперь, когда это сделано, мы пойдем дальше и начнем формировать функция getList (), которая просто сделает запрос нашей базе данных. Единственная проблема состоит в том, что мы должны снабдить Ext форматом JSON, и что SQL только возвращает массив. Хорошо теперь Вы только должны выяснить для себя непосредственно скрипт, который будит делать преобразование между двумя.

Хорошо, в действительности это не надо делать... Фактически, на версиях PHP 5.2 и выше, эта функция уже выполняется. Однако, мы притворимся будто мы должны иметь возможность работать на любой версии PHP, таким образом мы должны будем добавить файл под названием JSON.php, который сделает преобразование для нас.

Allright, back to the code :

```
function getList()
{
    $query = "SELECT * FROM presidents pr, parties pa WHERE pr.IDparty = pa.IDparty";
    $result = mysql_query($query);
    $nbrows = mysql_num_rows($result);
    if($nbrows>0){
        while($srec = mysql_fetch_array($result)){
            // render the right date format
            $srec['tookoffice']=codeDate($srec['tookoffice']);
            $srec['leftoffice']=codeDate($srec['leftoffice']);
            $sarr[] = $srec;
        }
        $jsonresult = JEncode($sarr);
        echo '({ "total":' . $nbrows . ', "results":' . $jsonresult . '})';
    } else {
        echo '({ "total": "0", "results": ""})';
    }
}
```

Как Вы можете видеть, мы поместили все данные в большой \$arr массив и отправили это нашей функции JEncode (). Результатом, \$jsonresult, является массив JSON, который готов к отправке в Ext. Мы добавляем общее количество результатов и отправляем все назад через команду ECHO.


На заметку: команда ECHO - то, что облегчает отладку Ext/PHP... Если Вы не знаете то, о чем я говорю, пожалуйста загрузите FIREBUG и пользуйтесь им.

Между прочим, мы добавили небольшую функцию codeDate, которая легко приводит формат mm/dd/YYYY к формату YYYY-mm-dd. Мы делаем это именно так, чтобы мы могли послать это непосредственно в нашу базу данных SQL :

```
// Encodes a YYYY-MM-DD into a MM-DD-YYYY string
function codeDate ($date) {
    $tab = explode("-", $date);
    $r = $tab[1]."/". $tab[2]."/". $tab[0];
    return $r;
}
```

Хорошо, нам осталось только записать функцию JEncode, и все будет сделано!

```
function JEncode($arr){
    if (version_compare(PHP_VERSION,"5.2", "<"))
    {
        require_once("JSON.php"); //if php<5.2 need JSON class
        $json = new Services_JSON(); //instantiate new json object
        $data=$json->encode($arr); //encode the data in json format
    } else
    {
        $data = json_encode($arr); //encode the data in json format
    }
    return $data;
}
?>
```

Исходники для 2 части обучения вы можете скачать здесь :  [Скачать zip файл](#)