

Linux平台PHP服务端开发——

附加内容——Array转换成JSON

PHP数组转换成JSON字符串

- 不使用json_encode函数。
- 使用变量处理和字符串处理的基础函数实现。
- 换一种描述方式：使用PHP的基本功能实现json_encode函数。

实现之前以及遇到的问题

- 熟悉JSON格式，并使用json_encode输出结果查看。
- “ 要转义处理成\”，\也是如此，但是 ' 不处理。
- Object处理成{ }
- 字符串和数字类型直接拼接字符串即可，数组类型则继续进行转换。
- 要确定数组是否存在关联索引，比如[1,2]直接转换成[1,2]的字符串，而 [1,'a'=>2]要转换成{0:1,"a":2}的形式。

实现方式

- 最直接想到的方式：递归。
- 对于数组类型，则递归操作，这种方式实现起来简单并且代码量少。
- 实现考虑因素：一般的接口返回JSON数据不会太大，所以递归操作一般不会导致栈空间不足，而以目前硬件的发展水平，CPU和内存完全能够支持大量递归操作。

如何判断数组是否存在索引

- 使用array_keys函数：

```
function is_ind($a) {  
    if (is_array($a)) {  
        $keys = array_keys($a);  
        return ($keys !== array_keys($keys));  
    }  
    return false;  
}
```

递归实现方式

```
function arr_to_json($arr) {
    $ii = is_ind($arr);
    $json = ($ii?'{'':'[');
    if ($ii) {
        foreach($arr as $k=>$v) {
            if (is_array($v)) {
                $json .= '"".$k.'":' . arr_to_json($v) . ',';
            } else {
                $json .= '""' . $k . '":' . json_num_str($v);
            }
        }
    } else {
        foreach($arr as $v) {
            $json .= (is_array($v)?(arr_to_json($v) . ','):json_num_str($v));
        }
    }
    return rtrim($json,',') . ($ii?'}':'']');
}
```

json_num_str函数

- 此函数处理object以及字符串和数字类型，对不同类型返回不同的字符串格式，对于字符串要进行转义处理。

```
function json_num_str($v)
{
    //转换" -> |" | -> ||, 但是 ' 不能是 |' 要变成 ', object直接转换成{}
    return (is_object($v) ? '{},' :
            (is_numeric($v) ? ($v . ',') :
            ('"' . str_replace("\\" , "'", addslashes($v)) . '",')
            )
    );
}
```

处理结果和json_encode对比

```
{"a":"abc","b":123,"c":[1,2,3]}  
{"a":"abc","b":123,"c":[1,2,3]}
```

```
[1,2,3,4]  
[1,2,3,4]
```

```
{"sdf":["\"sdf\"sdf\"\", \"sdfewer\"\"], \"dch\":[\":sdf:\", \":\\"]}  
{"sdf":["\"sdf\"sdf\"\", \"sdfewer\"\"], \"dch\":[\":sdf:\", \":\\"]}
```

```
{"100":["a","b",12,344,5,6], "list":[{"id":12,"name":"dwoerj"}, {"id":13,"name":"dw456y"}, {"id":14,"name":"dwtreg"}, {"id":15,"name":"dw345t"}], "h":{"a":[1,2,34], "b":{"vh":12, "sdf":[2,3,4,5]}}, "101":[["abc", "bcd", "cde"], ["xyz", "aui", "qwe"]]}  
{"100":["a","b",12,344,5,6], "list":[{"id":12,"name":"dwoerj"}, {"id":13,"name":"dw456y"}, {"id":14,"name":"dwtreg"}, {"id":15,"name":"dw345t"}], "h":{"a":[1,2,34], "b":{"vh":12, "sdf":[2,3,4,5]}}, "101":[["abc", "bcd", "cde"], ["xyz", "aui", "qwe"]]}
```

```
{"redis":{},"memcache":{}}  
{"redis":{},"memcache":{}}
```