如何实现 JavaScript RPC 应用

作者：谢声涛 [shishengsoft@gmail.com](mailto:shishengsoft@gmail.com)

**一、一个 JavaScript RPC 的实现**

**1、RPC工作流程**

JavaScript RPC 的简单实现过程是：

在客户端JavaScript脚本中，将要调用的服务端PHP函数名和函数参数（本地的JavaScript变量值）作为要传输的数据，通过 Ajax传输到服务端，同时，Ajax设置一个回调函数，以便使用服务端函数的返回结果。在服务端PHP脚本中，接收Ajax传输的数据，从中取出要执行 的函数名和函数参数。然后执行指定的函数，并将执行函数的返回值作为传输的数据，直接输出到浏览器，以此作为响应AJax的请求。客户端的Ajax在接受 服务端的响应后，把返回的数据传递给AJax的回调函数。到此完成了一个RPC的流程。

**2、传输的数据**

在RPC执行过程中，客户端与服务端需要相互传输数据，我们分别称之为客户端数据和服务端数据。对此的说明如下表：

客户端数据：

函数名：服务端PHP脚本中存在的函数。

函数参数：客户端JavaScript脚本中的变量值。

服务端数据：

函数返回值：执行服务端PHP函数的返回值。

**3、序列化与反序列化**

为了要传输客户端的变量值和服务端的函数返回值，需要对其进行序列化处理。与之相对应的在传输后还要进行反序列化处理。

JavaScript中的变量的类型：基本类型和JSON类型（普通数组、元对象）

PHP中的变量类型：基本类型和数组（普通数组和关联索引数组）

PHP语言具有序列化和反序列化函数，而JavaScript语言中没有，因此我们要自己实现它。

由于我们进行JavaScript和PHP之间的数据交换，所以直接把JavaScript变量序列化为PHP的序列化形式。我们实现Js2PhpSerialize函数来完成这个任务。

Js2PhpSerialize:主要是把JSON对象及嵌套调用的JSON对象和基本类型的数据表示为PHP的序列化形式。

服务端的PHP在执行函数后，把结果序列化为JavaScript能理解的形式，以便于eval动态生成变量。我们实现php2JsSerialize函数来完成这个任务。

**4、功能实现**

我们创建两个类来完成上述功能。一个是客户端的JavaScript类，一个是服务端的PHP类。

文件名：jsrpc.class.js

要实现的功能： A、序列化JavaScript的数据。 B、Ajax传输序列化数据 C、反序列化JavaScript数据。 D、设置回调函数。

文件名：jsrpc.class.php

要实现的功能： A、响应Ajax请求，接收序列化数据。 B、反序列化PHP数据，基本类型和数组。 C、序列化PHP数据为JavaScript数据。

将JavaScript的数据类型转换为PHP可以理解的形式，通过Ajax把转换后的数据传递到服务端，由PHP的eval函数解析为PHP的数据类型。

PHP函数执行后，将执行结果数据转换为JavaScript可以理解的形式，由Ajax的回调函数响应，并由JavaScript的eval函数解析为JavaScript的数据类型。

**二、应用示例**

代码片段：

JavaScript端：

<SCRIPT language=javascript src="jsrpc.class.js"></SCRIPT>      
    
<SCRIPT language=javascript>     
    
var arr = new Array(1,'abc', new Array(3,'xyz'),5);     
    
var rpc = new JsRPC();     
    
rpc.call('ajax\_server.php', 'checkBook', response, arr);     
    
function response(result) {     
    
     alert(result[1]);      
    
}     
    
</SCRIPT>

PHP端（即文件ajax\_server.php）：

<?php

require\_once(&apos;jsrpc.class.php&apos;);

$ar = new JsRPC();

$ar->register(&apos;checkBook&apos;);

$ar->start();

function checkBook($param) {

     return $param;

}

?>

如上所示，它使用方法与一个叫CPaint的类库相似，只是省去了将数据装到一个对象中的步骤。

**三、获取源码**

jsrpc.class.js和jsrpc.class.php文件的源码通过下面网址获取。

 JavaScript RPC Client 源码 <http://blog.csdn.net/shishengsoft/archive/2008/07/12/2643920.aspx>

JavaScript RPC Server 源码 <http://blog.csdn.net/shishengsoft/archive/2008/07/12/2643932.aspx>