

# Xajax 中文手册

原文: <http://xajax.sourceforge.net/>

英文原版: Copyright ? 2005 J. Max Wilson

中文翻译: HonestQiao(乔楚) / 整理: lfj[at]icbuy.net

## 什么是 Xajax?

Xajax 是一个开源的 PHP 类库 它能够让你黏合 HTML、CSS、JavaScript 和 PHP, 轻而易举的开发功能强大、基于 Web 的 Ajax 应用软件. 使用 Xajax 开发的应用软件, 无需重新调入页面, 就能够异步调用服务器端的 PHP 函数和更新内容.

## Xajax 如何工作?

你的应用软件需要异步调用的 PHP 函数, Xajax 的 PHP 对象都生成了对应的封装好了的 JavaScript 函数. 当被调用时, 封装的函数使用 JavaScript 的 XMLHttpRequest 对象与服务器异步通讯, 调用 Xajax 对象对应的 PHP 函数. 调用结束后, PHP 函数由 Xajax 返回一个 Xajax 的 XML 响应传递给应用程序. XML 响应包含了特定的指令和数据, 他们可以被 Xajax 的 JavaScript 消息分析器解析, 并且被用于更新你的应用程序的内容.

## 什么我要使用 Xajax 代替其他 PHP 的 Ajax 库?

你应该选择一个最适合你的项目需要的库. Xajax 提供了以下的功能, 它们使得 Ajax 富有特色而又功能强大:

1. Xajax 特殊的 XML 响应 / Javascript 消息分析系统帮助你做到, 自动的处理函数返回的数据, 按照 PHP 函数返回的指令更新内容或者状态. 因为 Xajax 作了这些工作, 你不需要写 Javascript 的回调处理函数.

2. Xajax 反对将代码和数据紧密地杂糅在一起的主张, 并且保持 Xajax 的代码从与他代码分离. 因为它是对象构造的, 你可以加上自己定制的功能给 Xajax 去扩展 XajaxResponse 类和使用 addScript 方法.

3. Xajax 可以工作在 Firefox, Mozilla, 大部分基于 Mozilla 的浏览器, Internet Explorer, 和 Safari.

4. 除了更新元素的值和内含的 HTML 内容 (innerHTML), Xajax 还能用于更新样式(styles), css 类, 多选和单选按钮选择,甚至可以更新任何元素的属性.

5. Xajax 支持使用一维或者多维数组、关联数组(哈希数组) 作为 Xajax 函数的参数从 Javascript 传送给 PHP. 反之 Additionally, 如果你传送一个 Javascript 的对象给 Xajax 函数, PHP 函数将接受一个描述对象属性的关联数组(哈希数组).

6. Xajax 提供了一种简单的异步表单处理方式. 使用 `xajax.getFormValues()` Javascript 方法, 你可以轻而易举的在表单之中提交一个描绘值的数组作为参数传送到 Xajax 异步处理函数:

```
xajax_processForm(xajax.getFormValues('formId'));
```

它可以处理复杂的 input 元素名称, 例如 "checkbox[][]" 或者 "name[first]" 产生的多维或者关联数组(哈希数组), 就是普通提交表单那样使用 PHP 的 \$\_GET 数组

7. 使用 Xajax 你可以动态的发送附加的 Javascript 作为请求的响应到你的应用软件中运行, 而这就和动态的更新元素的属性一样方便.

8. Xajax 会自动比较 PHP 函数返回的数据与你已经标记需要修改的元素属性. 只有当新的数据确实可以改变现有的属性, 属性才会真的被更新. 这将可消除程序在一定时间间隔内更新与当前内容相同或者不同的内容而出现的明显的闪烁.

9. 每一个通过 Xajax 简单注册的函数都可以有不同的请求方式. 所有的函数默认使用 POST 方式, 除非你明确的指定使用 GET 方式. 如果使用 GET 请求, 你必须非常明确的考虑它确实是你所需要的.

10. 如果没有指定 URI, Xajax 将设法自动确定脚本的 URI. Xajax 的自动确定算法经过了充分的验证, 确保在大部分 HTTPS 或者未知端口的 HTTP 服务器上也可以正常的工作.

11. Xajax 使用 UTF8 编码所有的请求和响应, 以确保支持绝大部分的字符和语言. Xajax 已经成功测试了多种不同的 UNICODE 字符, 包括 Spanish, Russian, Arabic, and Hebrew.

12. 几乎所有 Xajax 生成的 Javascript 都可以通过动态的外部 Javascript 包含到你的 web 程序之中. 当你在浏览器之中查看程序的源代码时, 不会有 JavaScript 的函数定义混杂在 HTML 标记之中.

Xajax 可以使用在 Smarty 模板系统之中, 仅仅需要创建一个 Smarty 变量就可以包含 Xajax 的 Javascript:

```
$smarty->assign('xajax_Javascript', $xajax->getJavascript());
```

然后你可以使用在模版之中使用 `{Xajax_Javascript}` 从而把 Xajax 应用到你的站点.

## 如何在我的 PHP 脚本之中使用 Xajax?

Xajax 的设计是如此的富有特色,以至于不管是已有的 Web 程序还是新的项目,它都能够被极其简单的部署和应用. 仅仅需要七步,你就可以在几乎任何 PHP 脚本之中加入 Xajax 的强大功能:

1. 包含 Xajax 类库:

```
require_once("xajax.inc.php");
```

2. 实例化 Xajax 对象:

```
$xajax = new xajax();
```

3. 注册你需要通过 Xajax 调用的 PHP 函数的名称:

```
$xajax->registerFunction('myFunction');
```

4. 编写注册的 PHP 函数,并且在函数之中使用 XajaxResponse 对象返回 XML 指令:

```
function myFunction($arg)
{
    // 对$arg 做一些基本处理例如从数据库检索数据
    // 然后把结果赋值给变量,例如$newContent

    // 实例化 XajaxResponse 对象
    $objResponse = new xajaxResponse();

    // 添加指令到响应之中,用于指派
    // 指定元素(例如 id="SomeElementId")的 innerHTML 属性的新的内容
    $objResponse->addAssign("SomeElementId","innerHTML",
    $newContent);

    //返回 XajaxResponse 对象生成的 XML 响应
    return $objResponse->getXML();
}
```

5. 在你的脚本输出任何信息之前,调用 Xajax 用于接管请求:

```
$xajax->processRequests();
```

6. 在页面的 <head></head> 标签之间,告诉 Xajax 生成所必需的 JavaScript:

```
<?php $xajax->printJavascript(); ?>
```

7. 在程序中, 从 JavaScript 事件或者函数调用前面注册的函数:

```
<div id="SomeElementId"></div>  
<button onclick="xajax_myFunction(SomeArgument);">
```

就这么简单. Xajax 会处理其他所有的事情. 你所要做的主要工作就是编写 PHP 函数, 然后从函数之中返回 Xajax 的 XML 响应. 而后者通过 XajaxResponse 类可以非常简单的生成.

## 如何异步更新内容?

Xajax 最富有特色的功能或许就是 XajaxResponse 类. 其他的 Ajax 库需要你编写 JavaScript 的回调句柄去处理一个异步请求返回的数据并更新内容. Xajax, 从另外一个角度来说, 允许你使用 PHP 简单的控制内容. XajaxResponse 让你在 PHP 函数之中创建 XML 指令返回给你的程序. XML 将被 Xajax 的消息分析器解析, 指令将告诉 Xajax 如何更新程序的内容和状态. XajaxResponse 类目前提供了以下指令:

### 1. addAssign(\$sTargetId,\$sAttribute,\$sData)

给命名为 \$sTargetId 的元素的 \$sAttribute 属性赋值 \$sData

```
$objResponse->addAssign("contentDiv","innerHTML","Some Text");  
$objResponse->addAssign("checkBox1","checked","true");
```

### 2. addAppend(\$sTargetId,\$sAttribute,\$sData)

给命名为 \$sTargetId 的元素的 \$sAttribute 属性追加值 \$sData

```
$objResponse->addAppend("contentDiv","innerHTML","Some Text");
```

### 3. addPrepend(\$sTargetId,\$sAttribute,\$sData)

预备给命名为 \$sTargetId 的元素的 \$sAttribute 属性赋值 \$sData

```
$objResponse->addPrepend("contentDiv","innerHTML","Some Text");
```

### 4. addReplace(\$sTargetId,\$sAttribute,\$sSearch,\$sData)

替换命名为 \$sTargetId 的元素的 \$sAttribute 属性的值之中的 \$sSearch 为 \$sData

```
$objResponse->addReplace  
("contentDiv","innerHTML","text","<strong>text</strong>");
```

### 5. addClear(\$sTargetId,\$sAttribute)

清空命名为 \$sTargetId 的元素的 \$sAttribute 属性的值

```
$objResponse->addClear("Input1","value");
```

#### 6. addCreate(\$sParentId, \$sTagName, \$sId, \$sType)

在已经存在的命名为 \$sParentId 的元素下添加一个名叫 \$sTagName 的子元素，并且把他的 id 赋值为 \$sId，可选的类型赋值为 \$sType.

```
$objResponse->addCreate("form1","input", "pass", "password");
```

#### 7. addRemove(\$sElementId)

从你的程序之中移除命名为 \$sElementId 的元素

```
$objResponse->addRemove("div1");
```

#### 8. addAlert(\$sMsg)

显示一个内容为 \$sMsg 的警告框(JavaScript 的 Alert)

```
$objResponse->addAlert("This is some text");
```

#### 9. addScript(\$sJS)

执行 JavaScript 代码 \$sJS (演示代码原文有误)

```
$objResponse->addScript("var txt = prompt('get some text');");
```

一个独立的 XML 响应可能包含多个指令，他们将按照加入响应的顺序执行。让我们用一个用户在你的程序之中点击按钮为例来进行说明。OnClick 事件调用 PHP 函数对应的 Javascript 封装。这个封装通过 XMLHttpRequest 发送异步请求到服务器给 Xajax 调用 PHP 函数。PHP 函数做了一次数据库查询，处理了一些数据，或者序列化。然后你使用 XajaxResponse 类生成包含多个指令的 Xajax 的 XML 响应，并回送给 Xajax 的消息分析器执行：

```
$objResponse = new xajaxResponse();  
$objResponse.addAssign("myInput1","value",$DataFromDatabase);  
$objResponse.addAssign("myInput1","style.color","red");  
$objResponse.addAppend("myDiv1","innerHTML",$DataFromDatabase2);  
$objResponse.addPrepend("myDiv2","innerHTML",$DataFromDatabase3);  
$objResponse.addReplace  
    ("myDiv3","innerHTML","xajax","<strong>Xajax</strong>");  
$objResponse.addScript("var x = prompt("Enter Your Name");");
```

```
return $objResponse->getXML();
```

Xajax 消息分析器将会解析 XML 消息，并执行以下工作:

- ✓ id 为 myInput1 的元素的值将被赋值为 \$DataFromDatabase 的数据。
- ✓ id 为 myInput1 的元素的属性将会变为 red。
- ✓ \$DataFromDatabase2 的数据会被追加到 id 为 myDiv1 的元素 innerHTML 之中。
- ✓ \$DataFromDatabase3 的数据会被预先赋值给 id 为 myDiv2 的元素 innerHTML 之中。
- ✓ id 为 myDiv3 的元素的 innerHTML 之中所有的 "xajax" 将被替换为 "<strong>xajax</strong>"; 使得所有的单词 xajax 显示加粗。
- ✓ 一个提示框将会显示，用来询问用户姓名，从提示框返回的值会被命名为 x 的 Javascript 变量接收。

所有这些都由构成的 PHP 函数在服务器端执行并返回 Xajax 的 XML 响应。

## 如何异步处理表单数据?

Xajax 使得异步处理表单数据非常非常的简单. xajax.getFormValues() 方法会自动的从表单提取数据，并作为一个参数提交给 Xajax 注册的 PHP 函数。

xajax.getFormValues() 仅仅需要一个参数，可以是你需要处理的表单的 id，或者是一个实际的表单对象。你也可以使用 Xajax.getFormValues 作为一个参数给 Xajax 函数，例如：

```
xajax_processFormData(xajax.getFormValues('formId'));
```

Xajax 会生成一个与表单数据对应的请求字符串给 Xajax 服务器解析，然后以一个与表单数据对应的数组传递给 PHP 函数，就像你提交表单使用 PHP 的 \$\_GET 数组那么简单。

Xajax 可以处理类似普通多维数组或者联合数组(哈希数组)等形式的复杂输入名字。例如，如果一个表单有三个多选框 (checkboxes) 并且都命名为 "checkbox[]"，但是值分别为 "check1", "check2", 和 "check3", 然后使用 Xajax.getFormValues 函数作为参数传递给 Xajax 函数，则 PHP 函数会接受到一个如下的数组：

```
array (
  'checkbox' =>
  array (
    0 => 'check1',
    1 => 'check2',
    2 => 'check3',
  ),
)
```

)

作为函数参数的数组的结构与传统意义上提交表单之后的 \$\_GET 数组的结构相同. 你可以访问数组之中的 checkbox 的数据: \$aData['checkbox'][0]

## 如何给Xajax增加定制功能?

Xajax 可以使用各种附加的用户定制功能进行扩展. 正因为 Xajax 是完全面向对象的, 并且可以使用 XajaxResponse 的 addScript() 方法, 所以他具有无限扩展的可能. 你可以创建你自己的 Xajax 响应类, 来继承 XajaxResponse 类以及它的方法, 并加上你自己定制响应. 让我们用一个定制的增加选择组合框(select combo boxes)选项的响应指令的例子来说明. 你可以象下面这样扩展 XajaxResponse 类:

```
class myXajaxResponse extends xajaxResponse
{
    function addAddOption($sSelectId, $sOptionText, $sOptionValue)
    {
        $sScript = "var objOption = new
Option('".$sOptionText."', '".$sOptionValue."');";
        $sScript .=
"document.getElementById('".$sSelectId."').options.add(objOption);";
        $this->addScript($sScript);
    }
}
```

现在, 取代 XajaxResponse 对象的初始化, 把你自己的 myXajaxResponse 对象的初始化定义到你的 Xajax PHP 函数之中:

```
$objResponse = new myXajaxResponse();
$objResponse->addAssign("div1", "innerHTML", "Some Text");
$objResponse->addAddOption("select1", "New Option", "13");

return $objResponse->getXML();
```

被调用时, 这个方法将会发送需要的 Javascript 到页面并执行. 当然你也有另外一种做法 Alternatively, 你可以在你的程序之中创建一个如下的 Javascript 函数:

```
<script type="text/Javascript">
function addOption(selectId,txt,val)
{
    var objOption = new Option(txt,val);
    document.getElementById(selectId).options.add(objOption);
}
```

```
</script>
```

并且使用 addScript() 调用这个方法:

```
$objResponse->addScript("addOption('select1','New Option','13');");
```

## 我能在私有或者收费产品之中使用Xajax吗?

简而言之: 能, 只要你愿意. Xajax PHP 类库的发布遵循 GNU Lesser General Public License (LGPL).