使用PHP和Ajax实现实时搜索

# 开发环境

1、环境搭建：Windows 7+Apache 2.4.18+MySQL 5.7.11+PHP 7.1.0。

2、文本编辑器：Sublime 3。

# 二、主要技术

本案例主要通过PHP读取XML文件，同时结合Ajax无刷新技术来实现实时搜索的功能。

# 三、效果图展示

实现效果图，如图1所示。

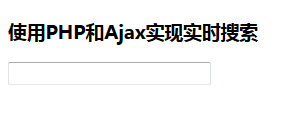


图1 效果图（操作界面）

# 四、具体步骤

## 4.1、创建PHP文件

创建一个名为“index.php”的PHP文件。

## 4.2、表单设计

<h3>使用PHP和Ajax实现实时搜索</h3>

<form>

<input type="text" size="30" onkeyup="showResult(this.value)">

<div id="livesearch"></div>

</form>

## 4.3、创建showResult()函数

### （1）创建一个名为showResult()的函数

function showResult(str){

...

}

### （2）判断用户输入是否为空

if (str.length==0){

document.getElementById("livesearch").innerHTML="";

document.getElementById("livesearch").style.border=

"0px";

return;

}

### （3）Ajax无刷新技术实现实时搜索

// 创建XMLHttpRequest对象

if (window.XMLHttpRequest){

// IE7+, Firefox, Chrome, Opera, Safari 浏览器执行

xmlhttp=new XMLHttpRequest();

}else{

// IE6, IE5 浏览器执行

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange=function(){

if (xmlhttp.readyState==4 && xmlhttp.status==200){

document.getElementById("livesearch").innerHTML=

xmlhttp.responseText; // 获取返回的响应信息

document.getElementById("livesearch").style.border

="1px solid #A5ACB2";

}

}

xmlhttp.open("GET","livesearch.php?q="+str,true); // 以get方式发送一个新请求

xmlhttp.send(); // 发送请求

## 4.4、使用PHP读取XML文件

### （1）创建XML文件

创建一个名为“links.xml”的XML文件。XML文件内容如下所示：

<pages>

<link>

<title>HTML a 标签</title>

<url>http://www.runoob.com/tags/tag-a.html</url>

</link>

<link>

<title>HTML br 标签</title>

<url>http://www.runoob.com/tags/tag-br.html</url>

</link>

<link>

<title>CSS background 属性</title>

<url>http://www.runoob.com/cssref/css3-pr-

background.html</url>

</link>

<link>

<title>CSS border 属性</title>

<url>http://www.runoob.com/cssref/pr-border.html</url>

</link>

<link>

<title>JavaScript Date 对象</title>

<url>http://www.runoob.com/jsref/jsref-obj-

date.html</url>

</link>

<link>

<title>JavaScript Array 对象</title>

<url>http://www.runoob.com/jsref/jsref-obj-

array.html</url>

</link>

</pages>

### （2）创建PHP文件

创建一个名为“livesearch.php”的PHP文件。

### （3）创建XML DOM对象

$xmlDoc=new DOMDocument();

### （4）加载XML文件

$xmlDoc->load("links.xml");

### （5）从XML文件中查找数据

$x=$xmlDoc->getElementsByTagName('link'); // 获取带有指定标签名的对象的集合

// 从URL中获取参数q的值

$q=$\_GET["q"];

// 如果q参数存在则从xml文件中查找数据

if (strlen($q)>0){

$hint="";

for($i=0; $i<($x->length); $i++){

// item() 方法可返回节点列表中处于指定索引号的节点。

$y=$x->item($i)->getElementsByTagName('title'); //

获取带有指定标签名的对象的集合

$z=$x->item($i)->getElementsByTagName('url'); //

获取带有指定标签名的对象的集合

if ($y->item(0)->nodeType==1){

// 找到匹配搜索的链接

if (stristr($y->item(0)->childNodes->item(0)-

>nodeValue,$q)){

if ($hint==""){

$hint="<a href='" .$z->item(0)->

childNodes->item(0)->nodeValue .

"' target='\_blank'>" .

$y->item(0)->childNodes->

item(0)->nodeValue . "</a>";

}else{

$hint=$hint . "<br /><a href='" .

$z->item(0)->childNodes->

item(0)->nodeValue .

"' target='\_blank'>" .

$y->item(0)->childNodes->

item(0)->nodeValue . "</a>";

}

}

}

}

}

// 如果没找到则返回 "no suggestion"

if ($hint==""){

$response="no suggestion";

}else{

$response=$hint;

}

### （6）输出查找结果

echo $response;

# 五、网页测试

①开启Apache服务器后，打开浏览器，在浏览器地址栏中输入文件地址，例如：“localhost:8888/PHParticle/081/Codes/”，可以看到程序运行结果如图2所示。

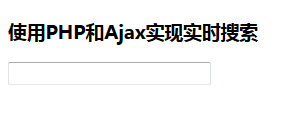


图2 案例结果图（操作界面）

②在文本框内输入“a”，可以看到程序运行结果如图3所示。

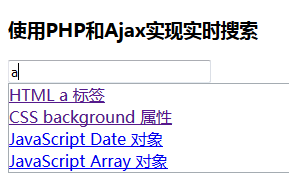


图3 案例结果图（实时搜索“a”）

③在文本框内输入“s”，可以看到程序运行结果如图4所示。

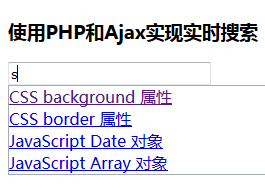


图4 案例结果图（实时搜索“s”）