PHP读取xml

# 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

本案例主要使用PHP 7中的DOM 库、SAX(XML Simple API)和正则表达式来实现xml文件内容的读取功能。

# 三、效果图展示

效果图如图1所示：

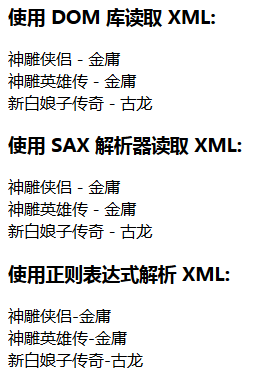


图1 效果图（xml文件读取信息）

# 四、具体步骤

## 4.1、准备xml文件

本实验中采用的xml文件（book.xml）内容如下：

<?xml version="1.0" encoding="UTF-8"?>

<sheelbook>

<book id="p1">

<title>神雕侠侣</title>

<author>金庸</author>

</book>

<book id="p2">

<title>神雕英雄传</title>

<author>金庸</author>

</book>

<book id="p3">

<title>新白娘子传奇</title>

<author>古龙</author>

</book>

</sheelbook>

## 4.2、PHP使用DOM 库读取 XML

echo "<h3>使用 DOM 库读取 XML:</h3>";

$doc = new DOMDocument();

$doc->load( 'book.xml' );

$books = $doc->getElementsByTagName( "book" );

foreach( $books as $book )

{

$authors = $book->getElementsByTagName( "author" );

$author = $authors->item(0)->nodeValue;

$titles = $book->getElementsByTagName( "title" );

$title = $titles->item(0)->nodeValue;

echo $title." - ".$author ."</br>";

}

## 4.3、PHP使用 SAX(XML Simple API) 解析器读取 XML

echo "<h3>使用 SAX 解析器读取 XML:</h3>";

$g\_books = array(); // 在内存中容纳所有图书和图书信息

$g\_elem = null; // 保存脚本目前正在处理的标记的名称

// 定义回调函数

//startElement 标记查找 book 标记，在 book 数组中开始一个新元素

function startElement( $parser, $name, $attrs )

{

global $g\_books, $g\_elem;

if ( $name == 'BOOK' ) $g\_books []= array();

$g\_elem = $name;

}

function endElement( $parser, $name )

{

global $g\_elem;

$g\_elem = null;

}

// textData 函数查看当前元素，看它是不是 title 或 author 标记。如果是，函数就把当前文本放入当前图书。

function textData( $parser, $text )

{

global $g\_books, $g\_elem;

if ( $g\_elem == 'AUTHOR' || $g\_elem == 'TITLE' )

{

$g\_books[ count( $g\_books ) - 1 ][ $g\_elem ]

= $text;

}

}

$parser = xml\_parser\_create(); // 创建XML解析器

// 设置回调句柄

xml\_set\_element\_handler( $parser, "startElement", "endElement" );

xml\_set\_character\_data\_handler( $parser, "textData" );

$f = fopen( 'book.xml', 'r' );

// 脚本读取文件并把文件的大块内容发送到解析器

while( $data = fread( $f, 4096 ) )

{

xml\_parse( $parser, $data );

}

// 删除解析器

xml\_parser\_free( $parser );

// 输出 g\_books 数组的内容

foreach( $g\_books as $book )

{

echo $book['TITLE']." - ".$book['AUTHOR']." <br/> ";

}

// 以数组的形式输出xml中所有内容

// $file="book.xml";

// $xml = simplexml\_load\_file($file);

// echo "<pre>";

// print\_r($xml);

// echo "</pre>";

## 4.4、PHP使用正则表达式解析 XML

echo "<h3>使用正则表达式解析 XML:</h3>";

$xml = "";

$f = fopen( 'book.xml', 'r' );

while( $data = fread( $f, 4096 ) ) { $xml .= $data; } // 读取数据

fclose( $f );

// 匹配最外层标签里面的内容

preg\_match\_all( "/\<sheelbook\>(.\*?)\<\/sheelbook\>/s",

$xml, $bookblocks );

foreach( $bookblocks[1] as $k=>$book )

{

preg\_match\_all("/\<title\>(.\*?)\<\/title\>/",

$book,$title); //匹配出书名

preg\_match\_all("/\<author\>(.\*?)\<\/author\>/",

$book,$author); //匹配出作者

}

// 输出内容

for($i=0;$i<=count($title);$i++){

echo $title[1][$i]."-".$author[1][$i]."<br/>";

}

# 五、网页测试

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进行测试。例如输入：“localhost:8080/test1.php”, 可以看到程序运行结果如图2所示。

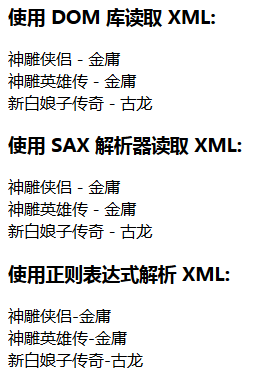


图2 案例结果图（xml文件读取信息）