

# PHP XML Expat 解析器

内建的 Expat 解析器使在 PHP 中处理 XML 文档成为可能。

## XML 是什么？

XML 用于描述数据，其焦点是数据是什么。XML 文件描述了数据的结构。

在 XML 中，没有预定义的标签。您必须定义自己的标签。

如需学习更多关于 XML 的知识，请访问我们的 [XML 教程](#)。

## Expat 是什么？

如需读取和更新 - 创建和处理 - 一个 XML 文档，您需要 XML 解析器。

有两种基本的 XML 解析器类型：

- 基于树的解析器：这种解析器把 XML 文档转换为树型结构。它分析整篇文档，并提供了对树中元素的访问，例如文档对象模型 (DOM)。
- 基于事件的解析器：将 XML 文档视为一系列的事件。当某个具体的事件发生时，解析器会调用函数来处理。

Expat 解析器是基于事件的解析器。

基于事件的解析器集中在 XML 文档的内容，而不是它们的结构。正因为如此，基于事件的解析器能够比基于树的解析器更快地访问数据。

请看下面的 XML 片段：

```
<from>Jani</from>
```

基于事件的解析器把上面的 XML 报告为一连串的三个事件：

- 开始元素：from
- 开始 CDATA 部分，值：Jani
- 关闭元素：from

上面的 XML 实例包含了形式良好的 XML。不过这个实例是无效的 XML，因为没有与它关联的文档类型声明 (DTD)。

然而，在使用 Expat 解析器时，这没有区别。Expat 是不检查有效性的解析器，忽略任何 DTD。

作为一款基于事件、非验证的 XML 解析器，Expat 快速且轻巧，十分适合 PHP 的 Web 应用程序。

**注释：**XML 文档必须形式良好，否则 Expat 会生成错误。

## 安装

XML Expat 解析器函数是 PHP 核心的组成部分。无需安装就可以使用这些函数。

## XML 文件

下面的 XML 文件将应用在我们的实例中：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

## 初始化 XML 解析器

我们要在 PHP 中初始化 XML 解析器，为不同的 XML 事件定义处理器，然后解析这个 XML 文件。

### 实例

```
<?php
//Initialize the XML parser
$xml_parser = xml_parser_create();

//Function to use at the start of an element
function start($parser, $element_name, $element_attrs)
{
    switch($element_name)
    {
        case "NOTE":
            echo "-- Note --<br>";
            break;
        case "TO":
            echo "To: ";
            break;
        case "FROM":
            echo "From: ";
            break;
        case "HEADING":
            echo "Heading: ";
            break;
        case "BODY":
            echo "Message: ";
            break;
    }
}

//Function to use at the end of an element
function stop($parser, $element_name)
{
    echo "<br>";
}

//Function to use when finding character data
function char($parser, $data)
{
    echo $data;
}

//Specify element handler
xml_set_element_handler($xml_parser, "start", "stop");

//Specify data handler
xml_set_character_data_handler($xml_parser, "char");

//Open XML file
$xml_file = fopen("test.xml", "r");
```

```
//Read data
while ($data=fread($fp,4096))
{
xml_parse($parser,$data,feof($fp)) or
die (sprintf("XML Error: %s at line %d",
xml_error_string(xml_get_error_code($parser)),
xml_get_current_line_number($parser)));
}

//Free the XML parser
xml_parser_free($parser);
?>
```

以上代码将输出：

```
-- Note --
To: Tove
From: Jani
Heading: Reminder
Message: Don't forget me this weekend!
```

工作原理：

1. 通过 `xml_parser_create()` 函数初始化 XML 解析器
2. 创建配合不同事件处理程序的函数
3. 添加 `xml_set_element_handler()` 函数来定义，当解析器遇到开始和结束标签时执行哪个函数
4. 添加 `xml_set_character_data_handler()` 函数来定义，当解析器遇到字符数据时执行哪个函数
5. 通过 `xml_parse()` 函数来解析文件 "test.xml"
6. 万一有错误的话，添加 `xml_error_string()` 函数把 XML 错误转换为文本说明
7. 调用 `xml_parser_free()` 函数来释放分配给 `xml_parser_create()` 函数的内存

## 更多 PHP Expat 解析器的信息

如需了解更多关于 PHP Expat 函数的信息，请访问我们的 [PHP XML Parser 参考手册](#)。

← PHP 数据库 ODBC

PHP XML DOM →

 点我分享笔记