iranphp articles

عنوان مقاله :	آشنایی با SimpleXML
نگارنده :	Sterling Hughes
آدرس پست الکترونیک :	.....
تاریخ نگارش :	.....

مقدمه:

این نوشته از سایت zend.com گرفته شده است و از سری مقالاتی است که نویسنده هر extension خود به معرفی قابلیتها و امکانات extensionپرداخته و چه چیز بهتر از این! برای این که ترجمه لفظ به لفظ باشد در بعضی از موارد بعضی از عبارات مورد نظر نویسنده اصلی درون [] قرار گرفته اند تا در عین سادگی متن اصلی حفظ شود .

هنگامی که مردم از من می پرسند SimpleXML چیست من به کنایه به آنها می گویم که XML : راه حلی برای تمام مشکلات ماست و به ما اطمینان می دهد که XML ریشه تمام مشکلات ما خواهد بود !

بعضی از شما که با PHP 4 به پردازش XML پرداخته اید و یا در حال حاضر این کار را انجام می دهید، می دانید که براستی handle کردن متون XMLبا هر درجه پیچیدگی که داشته باشند می تواند تا چه اندازه عذاب آور باشد، شما یا نیاز به استفاده از SAX دارید و برای هر متن XML نیاز دارید تا یک مفسر جداگانه بنویسید و یا اینکه از DOM استفاده کنید (به علاوه تمایل این روش به سقوط [crash] ، نشت [leak] و همچنین درست عمل نکردن در شرایط استفاده سنگین) باید درد و رنج استفاده از یک API که برای کار با برنامه های به شدت OO و کوچکترین خصوصیات XML نوشته شده را به جان بخرید .

قطعه کد XML زیر را که در آن دو کتاب در فرمت XML معرفی شده اند در نظر بگیرید، این کد یک گره ریشه به نام library دارد با یک فرزند مستقیم [Direct Child] به نام shelf که کتابها را در قسمت fiction کلاسه بندی می کند، این shelf دو کتاب را معرفی می کند: موشها و آدمها از جان اشتاین بک و هری پاتر و سنگ فیلسوفان از جی.کی رولینگ (توضیحات مفصل این کد برای این آمده است تا کسانی که به درستی با XMLآشنایی ندارند هم تا حدودی متوجه بحث شوند)

```
<?xml version="1.0"?>
  <library>
   <shelf id="fiction">
    <book>
     <title>Of Mice and Men</title>
     <author>John Steinbeck</author>
    </book>
    <book>
     <title>Harry Potter and the Philosopher's Stone</title>
     <author>J.K. Rowling</author>
    </book>
   </shelf>
  </library>
```

کد بالا به [حد کافی] ساده است: شما می توانید ساختار کلی را به وضوح مشاهده کنید و همچنین شما می توانید راهی را که برای دریافت اطلاعات از این کد لازم است به راحتی به دست آورید و آن را دنبال کنید .

خوب حالا قبل از اینکه ببینیم کار ما با استفاده از SimpleXML تا چه حد ساده می شود بگذارید به پردازش این کد با استفاده از DOM بپردازیم :

```
<?php
  $doc = new domDocument();
  $doc->load('library.xml');

  $library = $doc->documentElement;
  $shelves = $library->childNodes;

  foreach ($shelves as $shelf) {
     if ($shelf instanceof domElement) {
          process_shelf($shelf);
     }
  }

  function process_shelf($shelf)
  {
      printf("Shelf %s\n", $shelf->getAttribute('id'));
```

```
    $books = $shelf->childNodes;
    foreach ($books as $book) {
        if ($book instanceof domElement) {
            process_book($book);
        }
    }
}

function process_book($book)
{
    foreach ($book->childNodes as $child) {
        if (! ($child instanceof domElement)) {
            continue;
        }

        foreach($child->childNodes as $element) {
            $content = trim($element->nodeValue);

            switch ($child->tagName) {
            case 'title':
                printf("Title: %s\n", $content);
                break;
            case 'author':
                printf("Author: %s\n", $content);
                break;

            }
        }
    }
}
?>
```

همانطور که می بینید این کار بدون داشتن یک error handler با یک کد کاملا بهینه سازی شده ۴۷ خط شده است. با استفاده از یک error handlerو گذاشتن comment و یا سایر اطلاعات مورد نیاز در این برنامه طول آن به راحتی به ۷۰ الی ۸۰ خط می رسد. خوب حالا می خواهیم کار بالا را با استفاده از SimpleXML انجام دهیم به گونه ای که خروجی کاملا مشابه با کد بالا باشد :

```
<?php
$library = simplexml_load_file('library.xml');
foreach ($library->shelf as $shelf) {
    printf("Shelf %s\n", $shelf['id']);
    foreach ($shelf->book as $book) {
        printf("Title: %s\n", $book->title);
        printf("Author: %s\n", $book->author);
    }
}
?>
```

با استفاده از SimpleXML نام عناصر به صورت خودکار به خصوصیات [Properties] یک شیء نسبت داده می شود و این کار به صورت بازگشتی صورت می گیرد و آنگاه با استفاده از Iterator این کارها مرتبا انجام می گیرند تا تمام موارد موجود در متن به دست بیایند. همه این کارها در لحظه تقاضا [On-demand] با استفاده از Zend Engine 2 صورت می گیرند، روش کم چربی SimpleXML در تفسیر کد، ۴۷ خط مورد نیاز برای تفسیر با DOMرا به حدود ۱۰ خط رساند! در ثانی کد نوشته شده تا حد بسیار زیادی خوانا تر و قابل فهم تر نوشته شده است . در یک دنیای عالی و بی عیب و نقص تمامی متون XML و اطلاعاتی که شما نیاز دارید تا از آنها خارج کنید باید به همین سادگی مثال بالا باشند، در واقع در بسیاری از موارد این امر صحیح است اما در بعضی از موارد خاص روند نوشته شده در بالا جوابگو نیست .

یک مورد که SimpleXML با آن مواجه است Namespace ها می باشند، متون XML به شما امکان می دهند تا تگها را درون قسمتهای مشخصی بـه نام namespace ها بریزید، SimpleXMLاین کار را با استفاده از اضافه کردن یک لایه دیگر به متن حل می کند :

```
<?xml version="1.0"?>
  <entries xmlns:blog="http://www.edwardbear.org/serendipity/">
   <blog:entry>
    <blog:name>RPROF - Regular Expression Profiler</blog:name>
   </blog:entry>
   <blog:entry>
    <blog:name>Advanced PHP Programming</blog:name>
   </blog:entry>
  </entries>
```

برای به دست آوردن و نمایش دادن تمام اقلام مربوط به blog شما می توانید از کد زیر استفاده کنید :

```
<?php
  $entries = simplexml_load_file('syndic.xml');
  foreach ($entries->blog->entry as $entry) {
      printf("%s\n", $entry->name);
  }
  ?>
```

این راه حل روشی بود برای بهبود سادگی و بی تکلفی در پردازش متون XML. نکته ای که باید درمورد namespace ها درXML بدانیم این است کـه qualified nameکه در این مثال منظور ما همان blog است تنها یک شناسه است و هیچ رابطه خاصی را ایجاد نمی کند، برای شما که می خواهید به پردازش متون XML بپردازید منبع اصلی جایی نیست جز اینجا [1]
بنابراین روشی که SimpleXML برای پردازش متون با چند namespace به کار می برد هیچ تغییری در روشی که شما به این متون دسترسـی داریـد ایجاد نمی کند، اما به جای اینکه دو متد children و attributes را در اختیار شما قرار دهد، تابع children تمامی هـای children موجود در گره های XML موجود در یک namespace را بر می گرداند، اگر هیچ namespace ای برای تابع children برگردانده نشود آنگـاه تمام عناصر در namespace کلی [global] برگردانده می شوند .
مثال بالا با استفاده از کد زیر به خوبی و آسانی تفسیر می شود :

```
<?php
  $entries = simplexml_load_file('syndic.xml');
  foreach ($entries->children('http://www.edwardbear.org/serendipity/') as $entry) {
      printf("%s\n", $entry->name);
  }
  ?>
```

اما مشکلی که در SimpleXML وجود دارد این است که هرچند یک الگوریتم عالی برای تفسیر یک متن در اختیار قرار می دهد هیچ امکـانی بـرای انجـام کارهای معمولی از قبیل جستجو در متن ندارد، برای مثال چگونه می توان به تمام اولاد یک گره دسترسی پیدا کرد و یا چگونه می توان یک متن را بـرای پیـدا کـردن عبارتهـای برقـرار بـا شـرط جسـتجو کـرد؟ خیلـی از کارهـای معمـول بـا فایلهـای XML وجـود دارنـد کـه هنـوز بایـد سـاده شـوند .
به عنوان راه حلی برای این مشکلها SimpleXML دوباره کاری نکرده است و متود xpath را در اختیار شما قرار داده است که به شما اجازه مـی دهـد پرس و جوی های استاندارد W3C را روی یک متن XML انجام دهید، مشکل دسترسی به تمام اولاد یک گره با استفاده از کـوئری //children در xpathممکن است، هر چند قصد دارم در مقاله ای جداگانه به طور کامل به معرفی xpath بپردازم اما به شما توصیه می کنم که حتما و با جدیت این بحث را ادامه دهید زیرا xpath برای XML همان قدر اهمیت دارد که Regular Expressions برای متون ساده .
هر چند که SimpleXML یک ابزار عالی برای تفسیر متون XML در اختیار شما قرار می دهد اما خالی از اشکال هم نیست، یکی از مهمترین این اشکالات تفسیر متونی است که در آنها XML و متن با هم حضور دارند، برای مثال :

```
<?xml version="1.0"?>
  <flaw>
    <blurb>
      This <italic>is</italic> some sample <bold>text</bold> where
```

```
      SimpleXML <underline>will</underline> not behave well.
    </blurb>
  </flaw>
```

انتساب دادن $document->blurb به توابعی مثل print_r ویا var_dump مواردی مثل bold, italic و underline را برمی گرداند و نه متن موجود در مثال بالا را و این به این دلیل است که هرگاه به SimpleXML این حق انتخاب داده می شود که بین متن و عناصر یکی را انتخاب کند، SimpleXMLعناصر را انتخاب می کند .
SimpleXMLبرای حل این مشکل دو راه حل در کتابخانه همراه خود دارد، اول asXML که یک گره را گرفته و متون موجود در آن و همچنین متون موجود در children باز می گرداند، به طوری که اگر شما در مثال بالا از $document->blob->asXML() استفاده کنید متن با یک فرمت مناسب برای چاپ باز گردانده می شود .
راه حل دوم این است که شما خروجی SimpleXML را به عنوان یک شی در اختیار DOM قرار دهید .

[1]http://www.edwardbear.org/serendipity/