

IT5404

Internet Application Development

Extensible
Markup
Language

XML

XML

DOCUMENT PROCESSING

Introduction

- There are two ways to process XML documents.
 - Using DOM with JavaScript
 - Using SAX/ simple API for XML - PHP

Using DOM

- This is also called “**Tree-based approach**”. Because it uses DOM tree.

Using SAX

- This is also called “**event-based approach**”. Because it uses a set of events of API which are triggered when finding specific segment in the XML document.

DOM vs. SAX

DOM	SAX
Should be wait until the DOM has been completed.	No need to wait untill the DOM is completed.
Any node of the DOM can be accessed directly	Document is processed from the begin to end.

- **An event-based parser Parser is preferable for large files**, because tree-based parsers must fully load the file into memory in order to parse the XML.Event-based parsers do not need to load the entire file into memory to begin parsing.
- The DOM is the preferred method to use when the XML document must be modified in place or when you need to keep the document around in memory to do more advanced processing of the XML data

- **An event-based parser is preferable for large files**, because tree-based parsers must fully load the file into memory in order to parse the XML. Event-based parsers do not need to load the entire file into memory to begin parsing.
- **The DOM is the preferred method to use when the XML document must be modified in place** or when you need to keep the document around in memory to do more advanced processing of the XML data.
- **The SAX API is better when you only need to run through the XML file once and process each XML tag individually, but does not provide a way to edit the contents of the document.**

Loading XML Document

```
var xmlDoc ;  
if(window.ActiveXObject){  
    xmlDoc = new ActiveXObject("Microsoft.XMLDOM") ;  
}else{  
    xmlDoc = document.implementation.createDocument("", "", null);  
}  
  
xmlDoc.load("order.xml") ;
```

properties

- nodeName
- nodeValue
- text //text content of the node
- parentNode
- childNodes[x]
- firstChild/ lastChild
- nextSibling
- hasChildNodes // returns true or false
- attributes
- documentElement //root element

methods

- `getElementsByTagName("tag_name")`
- `appendChild(node)`
- `removeChild(node)`
- `createElement(element)`
- `createTextNode(text)`
- `getAttribute(attribute)`
- `setAttribute(attribute,value)`
- `removeAttribute(attribute)`

Example : display the details of the node

```
function displayNode(node){  
  
    document.write("node: " + node.nodeName + "<br/> Value: " + node.nodeValue) ;  
  
}
```



```
<?php
$xmlstr = "<?xml version='1.0' ?>
    <sentence> aset of names </sentence>
        <male age='25'> kamal </male>
        <male age='37'> nimal</male>"
        <female age='36'> niyomi</female>
        <female age='27'> shimali </female>
        <male age='27'> dias</male>
    </sentence>";

// creating the DOM
if($doc = xmldoc($xmlstr)){
    die("error parsing XML") ;
}

//accessing the root
$root = $doc->root() ;
```

```
// accessing the child nodes
$children = $root -> children() ;

for each ($children as $child){
    if($child -> tagName == "male"){
        $text = $child -> children() ;
        echo $text[0] -> content . "<br/>" ;
    }
}
```

PHP DOM consists of four classes

1. DomDocument class - represents the document itself

- Name, version
- root() , add_root(node)

2. DomElement class - represents the elements

- tagName
- Children(), parent(), attributes(), new_child(node), get_attribute(attribute), set_attribute(attribute,value), set_content(content)

3. DomText class – represents the text contents

- content

4. DomAttribute class – represents the attributes

- Value, name

Creating XML DOM

```
<?php
```

```
    $doc = new xmldoc("1.0") ;
```

```
    $root = $doc -> add_root("article") ;
```

```
    $root->set_attribute("id","3340") ;
```

```
    $title = $root->new_child("title","introduction to DOM") ;
```

```
    .....
```

```
?>
```


Explain the advantage(s) of using SAX (Simple API for XML) over DOM (Document Object Model).i.e For what occasions SAX is better than DOM.

- *SAX processes the entire document at once from begin to end and no need to wait until the DOM is created. It is useful when need a large file to process from begin to end.*
- *DOM enables to access any part directly but after the DOM is created. Which is useful if accessing several parts again and agin is needed and editing the DOM is needed aswell*

Explain the correctness or otherwise of the following statement.

Explain your answer.

“A tree-based XML parser is preferable for large files than an event-based XML Parser”

- *It is incorrect*
- *DOM parser should be waiting until the entire DOM is created which takes some duration of time for large file. So for this case the event based parser such as SAX is more preferable.*