



Cours PHP

PHP et XML : SAX, DOM et SimpleXML

Par Gauthier GARNIER

Contact mail et jabber : garnier.gauthier@gmail.com

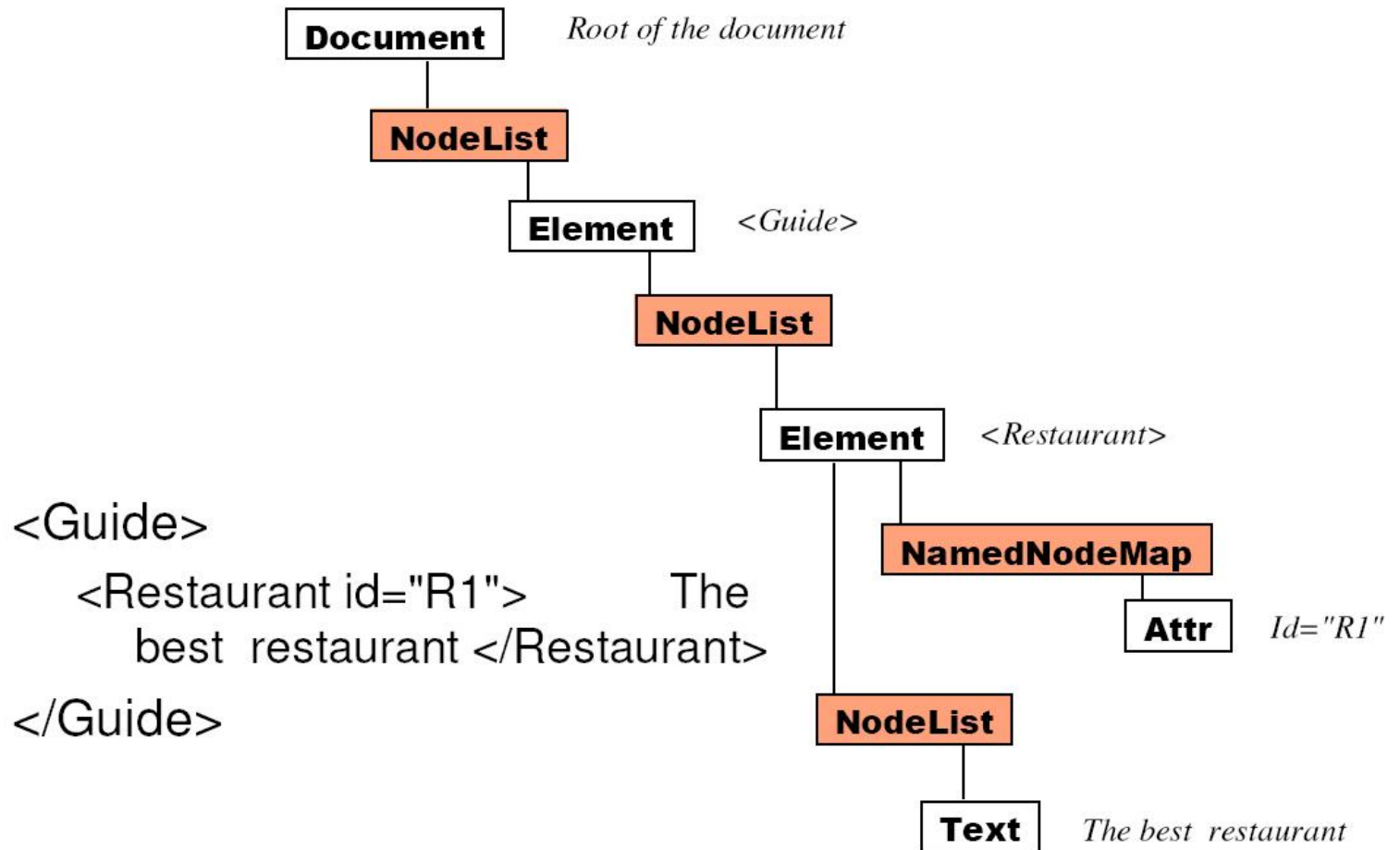


PHP et XML – Introduction

- libxml2 et libxslt : au cœur de PHP5
- Parsers hiérarchiques
 - Fonctionnement
 - Extension DOM
 - Extension SimpleXML
- Parsers évènementiels
 - Fonctionnement
 - Extension XML (basée sur SAX)
 - Extension XMLReader

PHP et XML – Parsers hiérarchiques

- Fonctionnement



PHP et XML – Parsers hiérarchiques

- Extension DOM (Document Object Model)
 - Documentation :
<http://fr.php.net/manual/fr/ref.dom.php>

PHP et XML – Parsers hiérarchiques

- Extension DOM (Document Object Model)

- Créer un document avec DOM

```
$dom = new DOMDocument();  
$dom->preserveWhiteSpace = FALSE;  
$dom->load('xmldata.xml');
```

- Afficher le contenu du DOM

```
$dom->formatOutput = TRUE;  
print $dom->saveXML();
```


PHP et XML – Parsers hiérarchiques

- Extension DOM (Document Object Model)

- Accéder au contenu du document

```
$root = $dom->documentElement;
```

- Afficher les attributs d'un nœud

```
echo "rootAttribs [".$root->attributes->length."] : \n";  
for ($i = 0; $i < $root->attributes->length; $i++)  
{  
    echo "    [".$root->attributes->item($i)->nodeName."] => ";  
    echo      $root->attributes->item($i)->nodeValue."\n";  
}
```

PHP et XML – Parsers hiérarchiques

- Extension DOM (Document Object Model)

- Accéder aux sous-éléments d'un noeud

```
$channels = $root->firstChild;  
echo "channelsChilds [\".$channels->childNodes->length.\"] : \n";  
for ($x = 0; $x < $channels->childNodes->length; $x++)  
{  
    echo "    [\".$channels->childNodes->item($x)->nodeName.\"] => ";  
    echo    $channels->childNodes->item($x)->nodeValue."\n";  
}
```


PHP et XML – Parsers hiérarchiques

- Extension SimpleXML – Parser type DOM

- Accéder au contenu du document

```
$xml = simplexml_load_file('test.xml');  
echo "rootName : ".$xml->getName()."\n";  
echo "rootType : ".gettype($xml)."\n";
```


PHP et XML – Parsers hiérarchiques

- Extension SimpleXML – Parser type DOM
 - Accéder aux attributs et sous éléments

```
echo "rootAttribs [".count($xml->attributes())."] : \n";
foreach ($xml->attributes() as $attrib => $value)
{
    echo "    [".$attrib."] => ";
    echo    $value."\n";
}
foreach ($xml->channel->children() as $child => $value)
{
    echo "    [".$child."] => ";
    echo    $value->title."\n";
}
```

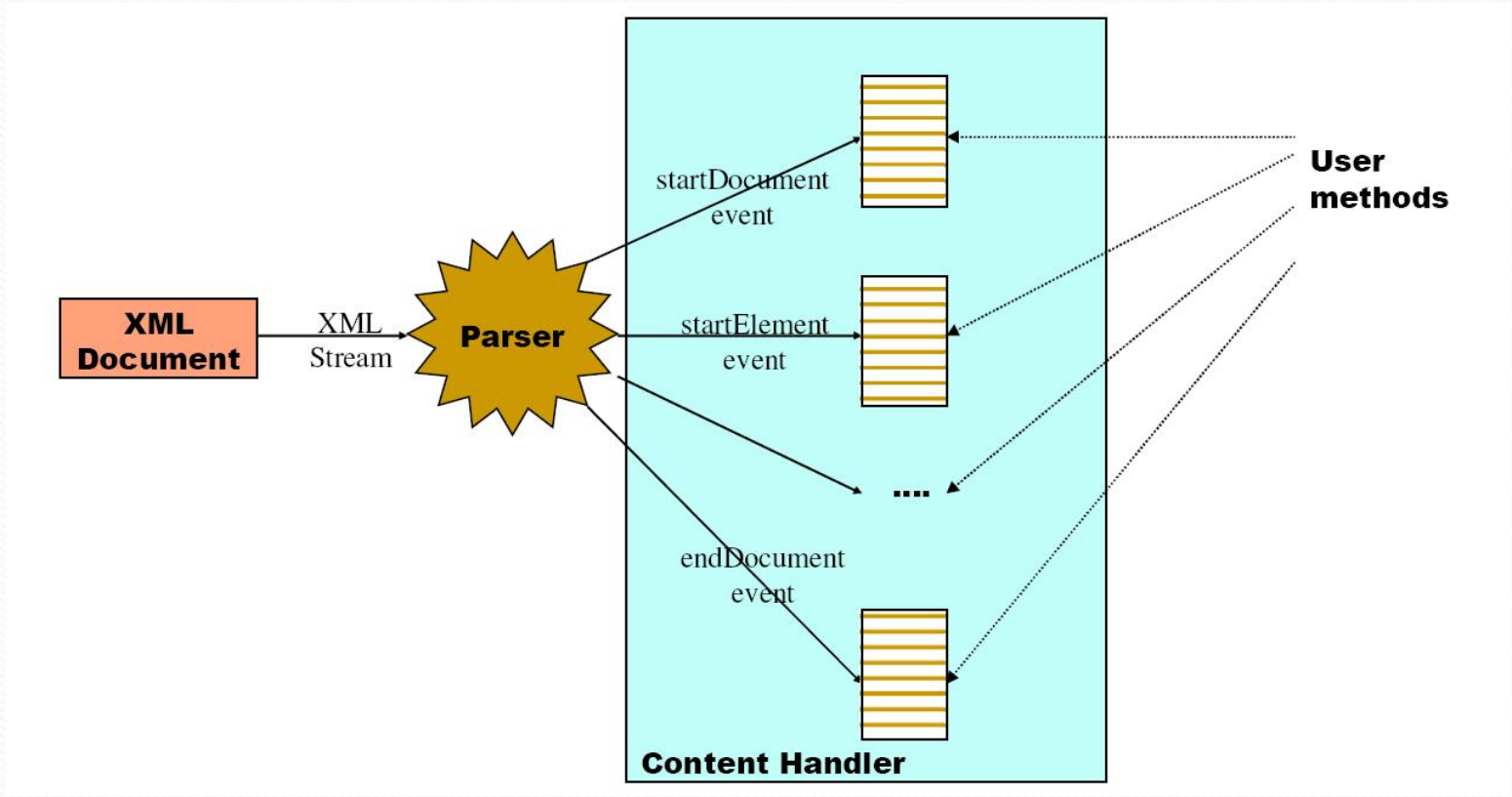
- Extension SimpleXML – Parser type DOM

- ```
print_r($xml);
```

```
<pre>SimpleXMLElement Object
(
 [@attributes] => Array
 (
 [version] => 2.0
)
 [channel] => SimpleXMLElement Object
 (
 [title] => Le Monde.fr : A la une
 [link] => http://www.lemonde.fr
 [description] => Toute l'actualité au moment de la connexion
 [language] => en
 [copyright] => Copyright Le Monde.fr
 [pubDate] => Wed, 05 Mar 2008 19:18:07 GMT
 [ttl] => 30
 [image] => SimpleXMLElement Object
 (
 [title] => Le Monde.fr : A la une
 [url] => http://medias.lemonde.fr/mmpub/img/lgo/lemondefr_rss.gif
 [link] => http://www.lemonde.fr
)
 [item] => Array
 (
 [0] => SimpleXMLElement Object
 (
 [title] => Salon automobile de Genève : les constructeurs automobiles s'inquiètent du ralentissement économique
 [link] => http://rss.feedsportal.com/c/205/f/3050/s/b2009e/story01.htm
 [description] => Le Salon de Genève grave;ve s'ouvre alors qu'une crise du crédit plane sur
les marchés européns. <div class='mf-enclosure' => SimpleXMLElement Object
```

# PHP et XML – Parsers évènementiels

- Fonctionnement





# PHP et XML – Parsers hiérarchiques

- Extension libxml – parser type SAX (Simple Api for Xml)
  - Simple à comprendre et à utiliser
  - Procédure pour utiliser cette extension :
    1. Définir les handlers évènementiels
    2. Créer le parser
    3. Paramétrer les options du parser
    4. Définir les fonctions qui interceptent les évènements
    5. Commencer le parsing
    6. Vérifier les erreurs
    7. Libérer la mémoire

# PHP et XML – Parsers hiérarchiques

- Extension libxml – parser type SAX (Simple Api for Xml)

## 1. Définir les handlers évènementiels

```
function startElement($parser, $name, $attrs) {
 print "<$name";
 foreach ($attrs AS $attName=>$attValue) {
 print " $attName=".'"'.$attValue.'"';
 }
 print ">";
}
function endElement($parser, $name) {
 print "</$name>";
}
function chandler($parser, $data) {
 print $data;
}
```



# PHP et XML – Parsers hiérarchiques

- Extension libxml – parser type SAX (Simple Api for Xml)

## 2. Créer le parser

```
$xml_parser = xml_parser_create();
```

## 3. Paramétrer les options du parser

```
xml_parser_set_option ($xml_parser, XML_OPTION_CASE_FOLDING, 0);
```

## 4. Définir les fonctions qui interceptent les évènements

```
xml_set_element_handler($xml_parser, "startElement", "endElement");
xml_set_character_data_handler ($xml_parser, "chandler");
```



# PHP et XML – Parsers hiérarchiques

- Extension libxml – parser type SAX (Simple Api for Xml)

5. Commencer le parsing

6. Vérifier les erreurs

```
if (!xml_parse($xml_parser, $xml, 1)) {
 die(sprintf("XML error: %s at line %d",
 xml_error_string(xml_get_error_code($xml_parser)),
 xml_get_current_line_number($xml_parser)));
}
```

7. Libérer la mémoire

```
xml_parser_free($xml_parser);
```

# PHP et XML – Parsers hiérarchiques

- Extension XMLReader – parser type SAX
  - **Définition** : L'extension XMLReader un analyseur XML Pull. Le lecteur fonctionne comme un curseur qui avance sur le flux du document et s'arrête à chaque noeud sur le chemin.
  - Fonctionnement proche de la libxml mais orientée objet
  - Exemple d'utilisation :

<http://fr.php.net/manual/fr/ref.xmlreader.php#61929>



# PHP et XML – DOM vs SAX

- DOM utilise SAX pour construire l'arbre d'un document XML
- SAX (Simple API for XML)
  - API dirigée par les événements
  - Envoie les événements à l'application au fur et à mesure de la lecture du document
  - Adaptée pour de gros documents et des dispositifs à mémoire contrainte
- DOM (Document Object Model)
  - Lit le document en entier en mémoire sous forme d'arbre
  - Adapté lorsque l'on veut modifier le document ou faire des manipulations complexes
  - SAX ne se souvient pas des événements antérieurs (sauf programmation explicite)



# PHP et XML – TP

- Fichier de référence :  
[http://www.automobile-magazine.fr/automagazine/export\\_rss.xml](http://www.automobile-magazine.fr/automagazine/export_rss.xml)
- Parser le fichier fourni avec les libs suivantes et affichez en html la liste des news, leur lien et leur vignette (champs « enclosure ») : **(4 points par lib)**
  - Avec DOM
  - Avec SimpleXML
  - Avec libxml (SAX)
  - Avec XMLReader
- Modifier le fichier avec DOM et le sauvegarder sur le disque (séparer en deux le champs Author au niveau du premier caractère « espace ») dans deux balises de la forme : **(4 points)**
  - `<author>Guntram Fiala</author>`
  - ➔ `<author>`
    - `<first_name>Guntram</first_name>`
    - `<last_name>Fiala</last_name>`
  - `</author>`