PHP5, XML, and XSL

Asst. Prof. Dr. Kanda Runapongsa Saikaew
(krunapon@kku.ac.th)
Department of Computer Engineering
Khon Kaen University

Agenda

- Calling Web services using PHP5 functions
- Writing and reading XML files using PHP5
- Transforming XML files using PHP5

Calling REST WS

- Set header of the content that we want to get as text/xml by using function header
  - header(“Content:type: text/xml”);
- Get the content of the WS by specifying the URL of that WS
  - $xml_content = file_get_content($url);

Example: Calling Local REST WS

```
<?php
   // file Calc_Client.php
   header("Content-type: text/xml");
   $url =
   "http://localhost/rest/calc.php?operati
   on=add&param1=1&param2=3";
   $xml = file_get_contents($url);
   echo $xml;
?>
```

Calling Remote REST Web Service

- Use an array to specify parameters
  - $params = array(
    'paraName1‘ => 'value1',
    'paraName2‘ => 'value2');
- To call a Web service from the outside host that has to pass through the proxy server before connecting to a host outside.
  - We need to define a context for HTTP

Defining a Context for HTTP

```
$aContext = array(
   'http' => array(
      'proxy' => 'tcp://202.12.97.116:8088', //
   This needs to be the server and the port of
   KKU Proxy Server.
      'request_fulluri' => True,
      ),
   );
$cxContext =
   stream_context_create($aContext);
```

Example: calling Remote REST WS (1/3)

```
<?php
// callAmazonRESTWS.php
header("Content-type: text/xml");
$base = 'http://webservices.amazon.com/onca/xml';
$query_string = '';
$params = array(
'Service' =>'AWSECommerceService',
'SubscriptionId' => '16XT8ETKKB7NWHAGCQ02' ,
'Operation' => 'ItemSearch',
'SearchIndex' => 'Books',
'Keywords' => 'Web Services');
```

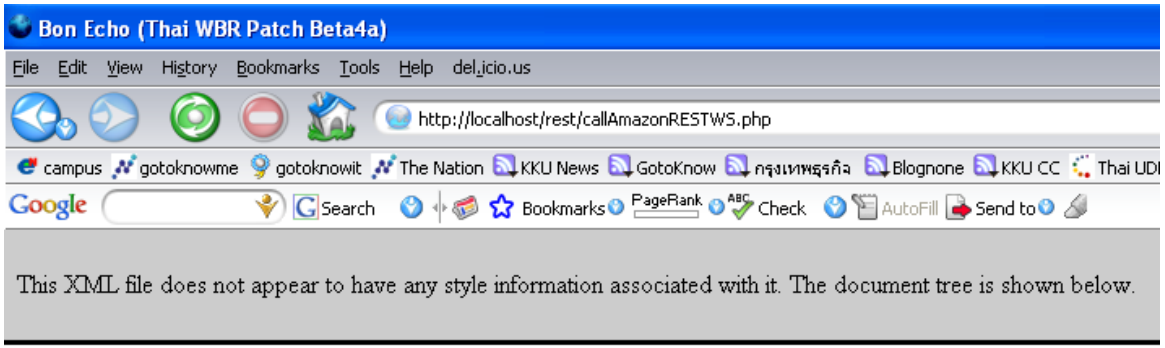
Example: calling Remote REST WS (2/3)

```
foreach ($params as $key => $value) {
    if ($key != 'Keywords')
     $query_string .= "$key=" . urlencode($value) .
    "&";
  else
   $query_string .="$key=".urlencode($value);
}
$url = "$base?$query_string";
// Define a context for HTTP.
$aContext = array(
```

Example: calling Remote REST WS (3/3)

```
'http' => array(
        'proxy' => 'tcp://202.12.97.116:8088',
        'request_fulluri' => True,
        ),
    );
$cxContext = stream_context_create($aContext);
// Now all file stream functions can use this context.
$xml = file_get_contents($url, False, $cxContext);
echo $xml;
?>
```

The Result from Calling Amazon WS




```
- <ItemSearchResponse>
  - <OperationRequest>
    - <HTTPHeaders>
        <Header Name="UserAgent"/>
      </HTTPHeaders>
      <RequestId>0MV2X5G2GJKMSR4KY2B7</RequestId>
    - <Arguments>
        <Argument Name="SearchIndex" Value="Books"/>
        <Argument Name="Service" Value="AWSECommerceService"/>
        <Argument Name="Keywords" Value="Web Services"/>
        <Argument Name="SubscriptionId" Value="16XT8ETKKB7NWHAGCQ02"/>
        <Argument Name="Operation" Value="ItemSearch"/>
      </Arguments>
      <RequestProcessingTime>0.0562698841094971</RequestProcessingTime>
    </OperationRequest>
  - <Items>
    - <Request>
        <IsValid>True</IsValid>
      - <ItemSearchRequest>
          <Keywords>Web Services</Keywords>
          <SearchIndex>Books</SearchIndex>
        </ItemSearchRequest>
      </Request>
```

Agenda

- Calling Web services using PHP5 functions
- **Writing and reading XML files using PHP5**
- Transforming XML files using PHP5

PHP5 and XML

- In PHP 5 working with XML can be done in any number of ways, via a whole series of extensions
  - XML extension
  - DOM
  - XML-RPC
  - Simplexml
  - SOAP
  - XML Reader (PECL)
  - XML Writer (PECL)

Extension Functionality (1/2)

- The PHP's XML extensions can be split into 4 categories, based on their functionality and backend processing mechanisms.
  - Document Object Model (DOM)
    - SimpleXML
    - DOM Evolution of domxml in PHP 4
  - Simple API for XML (SAX)
    - XML Same as in PHP 4
    - XMLReader

Extension Functionality (2/2)

- Web Services
  - SOAP
  - XML-RPC Same as in PHP 4
- XML Transformation
  - XSL Evolution of XLST in PHP 4

LibXML2 vs Expat

- Aside from the addition of the new XML parsing extensions, in PHP5 the underlying XML library was changed as well
- LibXML2 the Good
  - Feature complete
  - Active development
  - Very fast for DOM
- and the Bad
  - Significantly larger then Expat (not bundled with PHP)
  - Slightly slower for SAX

XmlReader

- Very new extension (in PECL)
- a so called Pull Parser
- API derived from C# XmlTextReader
- A replacement for SAX

XmlReader vs SAX

- Sax is Event based
- XmlReader is cursor based
- XmlReader has support for namespaces, validation and entities
- Both do not load the whole document into memory

Advantages of XMLReader

- Unlike SimpleXML, it's a full XML parser that handles all documents, not just some of them
- Unlike DOM, it can handle documents larger than available memory
- Unlike SAX, it puts your program in control
- If your PHP programs need to accept XML input, XMLReader is well worth your consideration

A Sample Problem (1/2)

- An XML-RPC Request (request1.xml)

```
<methodCall>
   <methodName>sqrt</methodName>
   <params>
      <param>
         <value><double>36.0</double></value>
      </param>
   </params>
</methodCall>
```

A Sample Problem (2/2)

- An XML-RPC Response

```
<methodResponse>
   <params>
      <param>
            <value><double>6.0</dobule></value>
      </param>
   </params>
</methodResponse>
```

XMLReader in Action (1/4)

```
<?php
  $reader = new XMLReader();
  $reader-
  >open('http://localhost/phpxml/xmlrea
  der/request1.xml');

  // Reader the document
  while ($reader->read()) {
```

XMLReader in Action (2/4)

```
// $reader properties: localName,
// name, namespaceURI,
// nodeType, prefix, value, hasValue
        /*echo $reader->name;
        if ($reader->hasValue) {
                echo ": " . $reader->value;
        }
        echo "\n"; */
```

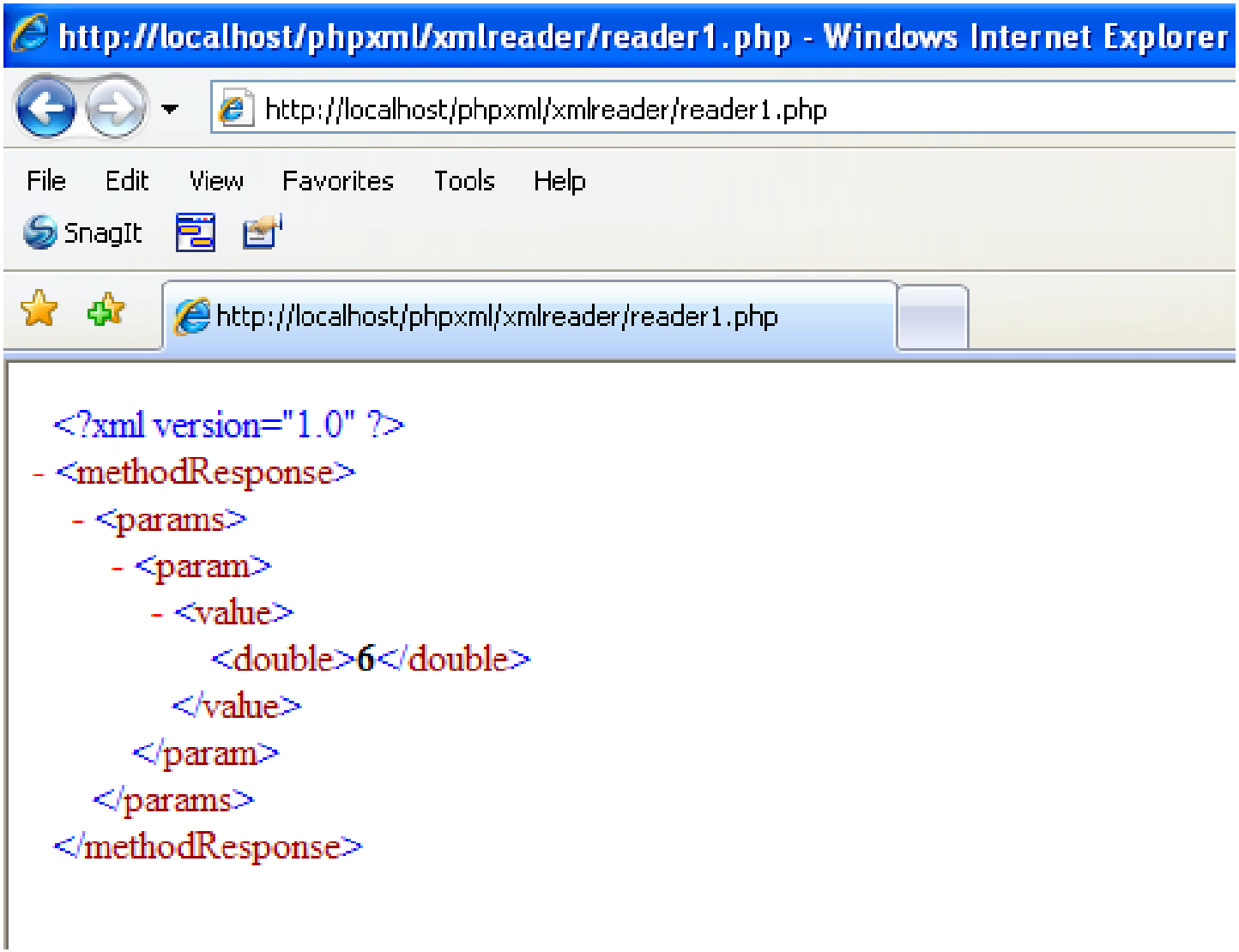
XMLReader in Action (3/4)

```
    if ($reader->name == "double" &&
       $reader->nodeType ==
       XMLReader::ELEMENT) {
            $reader->read();
            respond($reader->value);
       } // end if
 } // end while ($reader->read)
$reader->close();
```

XMLReader in Action (4/4)

```
function respond($input) {
  echo "<?xml version='1.0'?>
<methodResponse><params>
    <param>
      <value><double>"
   .sqrt($input)."</double></value>
    </param></params>
</methodResponse>";
}
```

The Response Result



Writing File XML using PHP5

- Use function fopen, fwrite and fclose to write XML content to XML file
- Example:

```
$fp = fopen(‘filename.xml','w');
fwrite($fp,$xml_content);
fclose($fp);
```

SimpleXML

- Using function simplexml_load_file for reading an XML file.
- The function returns a SimpleXMLElement object
  - Use print_r to print this object
- Example:

```
$xml_content_obj =
  simplexml_load_file(‘filename.xml');
print_r($xml_content_obj);
```

Example: test.xml

```
<?xml version="1.0"?>
<people>
   <person>
      <firstname>Kanda</firstname>
      <lastname>Runapongsa</lastname>
   </person>
   <person>
      <firstname>Kanda</firstname>
      <lastname>Saikaew</lastname>
   </person>
</people>
```

Example: readxml.php

```
<?php
if (file_exists('test.xml')) {
    $xml = simplexml_load_file('test.xml');
    print_r($xml);
    echo "<br/>";
    echo $xml->person[1]->firstname;
    echo $xml->person[1]->lastname;
} else {
    exit('Failed to open test.xml.');
}
?>
```

Processing XML using DOM (1/4)

```
<?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);
     }
  }
```

Processing XML using DOM (2/4)

```
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);
        }
    }
}
```

Processing XML using DOM(3/4)

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

Processing XML using DOM(4/4)

```
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;
 }
}}} ?>
```

Creating XML using DOM (1/3)

```
<?php
    // create the root node of DOM
    $doc = new DomDocument('1.0');

    $root = $doc->createElement('nation');
    $root = $doc->appendChild($root);
    $root->setAttribute('id','th');
```

Creating XML using DOM (2/3)

```
$name = $doc->createElement('name');
$nameValue =
    $doc->createTextNode('Thailand');
$name->appendChild($nameValue);
$location =
    $doc->createElement('location');
$locationValue =
$doc->createTextNode('Southeast Asia');
```

Creating XML using DOM (3/3)

```
    $location->appendChild($locationValue);
  $root->appendChild($name);
  $root->appendChild($location);
  $xml_string = $doc->saveXML();
  echo $xml_string;
?>
```

Agenda

- Calling Web services using PHP5 functions
- Writing and reading XML files using PHP5
- **Transforming XML files using PHP5**

DOMDocument

- Use a constructor DOMDocument() to create an object for loading an XML file
  - Example:

```
$doc = new DOMDocument();
$xml_filename = "test.xml";
$doc->load($xml_filename);
```

XSLTProcess()

- Use XSLTProcessor to create an XSLT process object for importing stylesheet and transforming
- Example:

```
$xsl = new XSLTProcessor();
$xsl_filename = "test.xsl";
$doc->load($xsl_filename);
$xsl->importStyleSheet($doc);
$xml_filename = "test.xml";
$doc->load($xml_filename);
echo $xsl->transformToXML($doc);
```

Input File: test.xml

```
<?xml version="1.0"?>
<people>
   <person>
      <firstname>Kanda</firstname>
      <lastname>Runapongsa</lastname>
   </person>
   <person>
      <firstname>Kanda</firstname>
      <lastname>Saikaew</lastname>
   </person>
</people>
```

Input File: test.xsl (1/2)

```
<?xml version='1.0'?>
<xsl:stylesheet
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version='1.0'>
    <xsl:output method="html"/>
    <xsl:template match="/people">
      <html><head><title>Test.html</title></head>
          <body>
                    <h1><xsl:value-of
                    select="person/firstname"/></h1>
           <xsl:apply-templates/>
        </body>
      </html>
    </xsl:template>
```

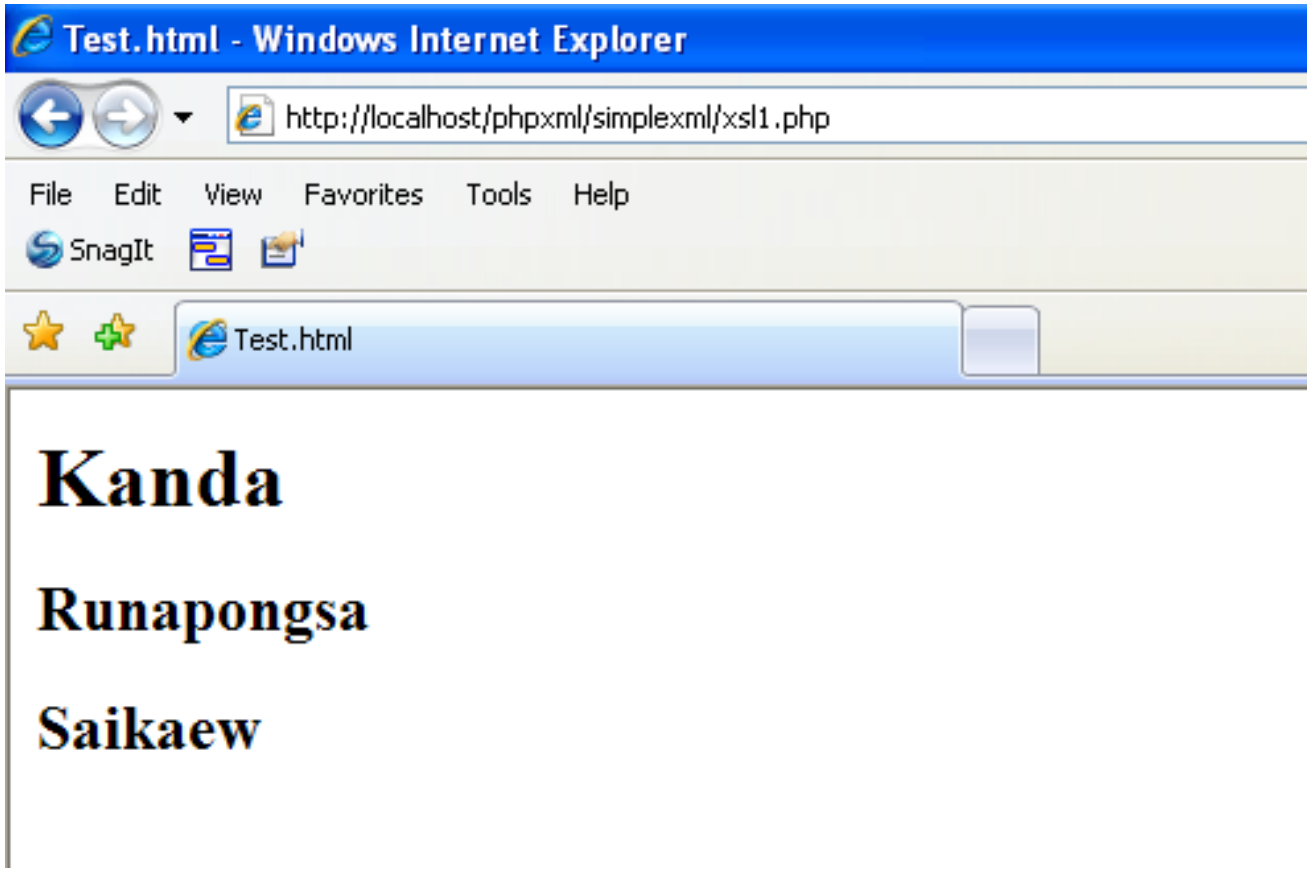
Input File: test.xsl (2/2)

```
<xsl:template match="person">
    <xsl:apply-templates
select="lastname"/>
 </xsl:template>
<xsl:template match="lastname">
    <h2><xsl:value-of select="."/></h2>
</xsl:template>
</xsl:stylesheet>
```

PHP File

```
<?php
  $doc = new DOMDocument();
  $xsl = new XSLTProcessor();
  $xsl_filename = "test.xsl";
  $doc->load($xsl_filename);
  $xsl->importStyleSheet($doc);
  $xml_filename = "test.xml";
  $doc->load($xml_filename);
  echo $xsl->transformToXML($doc);
?>
```

Output File: test.html



References

- PHP5 SimpleXML, http://us2.php.net/simplexml
- PHP5 DOMDocument, http://www.php.net/dom
- PHP5 and XML

http://talks.php.net/show/php5xml

- Pull Parsing XML in PHP http://www.ibm.com/developerworks/library/x-pullparsingphp.html

References

- PHP: XML Reader Manual

http://th.php.net/xmlreader