

PHP: SimpleXML

SimpleXML er en utvidelse av PHP (extension).

Den kan betraktes som en verktøykasse til hjelp når du skal konvertere et XML dokument til et objekt som kan behandles videre. Den gir en enkel måte å få adgang til XML data.

SimpleXML extension forutsetter PHP5 eller senere. Kjør php-funksjonen `phpinfo()`; for å bringe på det rene at SimpleXML er enablert. Hvis den ikke er det, må du redigere `php.ini`.

SimpleXML forutsetter libxml. Denne skal være enablert default. Det er ellers ingen egne direktiver i `php.ini` for simpleXML.

For dokumentasjon av SimpleXML, se <http://no.php.net/manual/en/book.simplexml.php>.

Det er 3 funksjoner (metoder) som brukes til å lage SimpleXMLElement objekter:

Funksjon	Resultat	Parameter
<code>simplexml_import_dom</code>	SimpleXMLElement	DOM node

Denne funksjonen lager et objekt av klassen SimpleXMLElement fra en DOM node.

Funksjon	Resultat	Parameter
<code>simplexml_load_file</code>	SimpleXMLElement	Filnavn

Denne funksjonen lager et objekt av klassen SimpleXMLElement fra en XML fil.

Funksjon	Resultat	Parameter
<code>simplexml_load_string</code>	SimpleXMLElement	Peker til stringen.

Denne funksjonen lager et objekt av klassen SimpleXMLElement fra en string. Denne stringen må selvfølgelig oppfylle alle krav til syntaks i XML.

Disse funksjonene gir deg muligheten til å ta utgangspunkt i et "XML dokument" du har på en eller annen representasjon.

Klassen SimpleXMLElement har følgende attributter og funksjoner tilgjengelig:

Funksjoner	Resultat	Beskrivelse
addAttribute	ingen verdi	Legger til et nytt attributt til SimpleXML elementet som funksjonen anvendes på. Funksjonen tar 2 parametre, navn på attributtet og verdien til attributtet. En evt. tredje parameter kan være en namespace referanse.
addChild	SimpleXMLElement, det child element som er lagt til.	Lager et SimpleXMLElement objekt (dvs. en XML tag), og legger det til som et child element til det elementet funksjonen blir anvendt på. Funksjonen tar 1, 2 eller 3 parametere, et navn, eventuelt en verdi og et namespace.
asXML	string, eller skrijving av stringen til en fil.	Lager en velformet XML string basert på SimpleXML elementet. Hvis det gis et argument, blir dette tolket som et filnavn, og resultatet blir lagret der. Alias: saveXML
attributes	SimpleXMLElement	Identifiserer attributtene til det SimpleXMLElement objektet det blir anvendt på. Hvis dette elementet er et attributt, returneres ikke noe.
children	SimpleXMLElement	Identifiserer child elementene til det SimpleXMLElementet det blir anvendt på.
count	tall	Antallet children i det SimpleXMLElementet det blir anvendt på.
getDocNamespaces	array	Returnerer de namespaces som er deklart i et dokument. Funksjonen kan kalles med en boolsk parameter. Hvis denne er true vil funksjonen også inkludere namespaces brukt i children og parent.
getName	string	Gir navnet på det SimpleXMLElementet det blir anvendt på.
getNamespaces	array	Returnerer de namespaces som brukes i et dokument, evt. med en boolsk parameter (som for getDocNamespaces).
registerXPathNamespace	boolean	Lager et prefix/namespace til den neste XPath query
xpath	array	Utfører en XPath query på SimpleXMLElement objektet, ser etter children som matcher path argumentet, returnerer disse i en array. Hvis det ikke er noen node som oppfyller kriteriet, returneres false.
__construct	SimpleXMLElement	Lager et nytt SimpleXMLElement objekt Parameter skal være en velformet XML string

De funksjonene som er skyggelagt er mindre viktige.

Et lite eksempel på bruk av dette

Anta at du har en fil doc.xml som inneholder:

```
<doc>
    <kapittel nr=1><avsnitt>Dette er første avsnitt</avsnitt></kapittel>
</doc>
```

La oss se på følgende php kode:

```
$xml = simplexml_load_file("doc.xml");
$kap = $xml->addChild('kapittel');
$kap->addAttribute('nr','2');
$kap->addChild('avsnitt','Dette er det andre avsnittet. ');
$xml->asXML('resultat.txt');
```

Resultatet av å evaluere denne php koden bør bli at det skrives en fil resultat.txt (under forutsetningen av at "du" har skriverett på den aktuelle katalog). Denne filen vil inneholde:

```
<?xml version="1.0"?>
<doc>
    <kapittel nr="1"><avsnitt>Dette er første avsnitt</avsnitt></kapittel>
    <kapittel nr="2"><avsnitt>Dette er det andre avsnittet.</avsnitt></kapittel>
</doc>
```

Den første linjen lager et simpleXMLElement fra fila.

Andre linje legger til en ny node for taggen kapittel, og denne noden refereres med \$kap.

Taggen i den nye noden gis et attributt nr med verdi "2" på linje 3.

På linje 4 legges det til en subtag med navn avsnitt og innhold "Dette er det andre avsnittet.".

På linje 5 skrives dette til fila resultat.txt.

På denne måten kan vi altså skape nye elementer i eksisterende elementer. Vi kan også skape nye elementer og legge til innhold fra eksisterende strukturer. Vi kan f.eks. lage en node hvor vi setter inn innholdet fra flere eksisterende filer. På denne måten kan du f.eks. lage en bok fra filer som representerer hvert sitt kapittel.

Klassen XSLTProcessor

XSL er en extension til PHP. XSL extension implementerer XSL standarden. Med den kan du utføre XSLT transformasjoner ved hjelp av biblioteket libxslt.

Det kan tenkes at dette ikke er enablet i php.ini; i så fall søker du etter *extension=php_xsl.dll*, og sørger for at den ikke er kommentert bort. Hvis du ikke finner den, legg til denne linja i php.ini.

XSL extension bruker også libxml, så de samme betraktninger som gjaldt simpleXML gjelder også XSL extension. XSL extension inneholder klassen XSLTProcessor.

Hvis du ikke finner den angitt extension i din php.ini fil, må du legge den til. Før du får den til å virke, må du restarte webserveren.

Hvis du skal anvende en XSL fil på et XML dokument, uten at du ønsker å legge inn spesifikasjon i XML fila om dette, kan du benytte klassen XSLTProcessor. Denne har følgende funksjoner:

Funksjon	Resultat	Beskrivelse
importStylesheet		Funksjonen importerer et XSL stylesheet i det XSLTProcessor objektet det anvendes på. Argumentet er et DOMDocument objekt, eller et SimpleXMLElement objekt. Dette objektet kan du ha fått ved å anvende simplexml_load_file på en XSL fil.
transformToDoc	DOM dokument, ellers false.	Funksjonen tar ett argument, en DOM node. Funksjonen anvender et importert stylesheet og transformerer denne DOM node (XML node) til et DOM dokument.
transformToURI	Antall byte skrevet, eller false.	Funksjonen tar ett argument, en DOM node. Funksjonen anvender et importert stylesheet og transformerer denne DOM node til et URI dokument, dvs. en fil.
transformToXML	string eller false	Funksjonen anvender et importert stylesheet og transformerer DOM node til et XML dokument i form av en string.
setParameter		Kan sette en ny verdi til en parameter.
getParameter	string eller null	Hent verdien til en XSLT parameter, første argument er NS, andre er navn på parameter.
removeParameter		Denne funksjonen "fjerner" en parameter i betydningen at default verdi skal benyttes.
hasExsltSupport	true/false	Sjekker om PHP har EXSLT lib inkludert.
registerPHPFunctions		Denne funksjonen gjør det mulig å bruke PHP funksjoner som XSLT funksjoner.
setProfiling		Angir fil som profiling informasjon skal skrives til.

De funksjoner som er skyggelagt er mindre viktige.

Et lite eksempel på bruk av dette

Anta at du har en xml-fil doc.xml som ovenfor, og en xsl-fil som inneholder:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" version="1.0" indent="yes" encoding="iso-8859-1"/>
<xsl:strip-space elements="doc chapter section"/>
<xsl:template match="doc">
  <html>
    <head><title><xsl:value-of select="title"/></title></head>
      <body><xsl:apply-templates /></body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

Vi kunne legge til en kommando i doc.xml for å vise fram fila med denne xsl fila.

I stedet skal vi lage en php-fil med følgende innhold:

```
$hxml = simplexml_load_file("doc.xml");
$hxml = simplexml_load_file("doc.xml");
$xmlt = new XSLTProcessor();
$xmlt->importStylesheet($hxml);
$xmlt->transformToURI($hxml,'results.txt');
include('results.txt');
```

Vi laster de to filene inn i hvert sitt simpleXMLElement på linje 1 og 2.

Vi lager et objekt av klassen XSLTProcessor på linje 3, og importerer stylesheetet til dette objektet på linje 4.

Så transformerer vi xml-dokumentet med dette stylesheet på linje5 og lagrer det i en fil.
Denne fila viser vi deretter som del av denne sida,

Tips til debugging

Husk funksjonen **var_dump()**, når du skal debugge XSLT transformasjoner. Den skriver ut alt innholdet en variabel eller et objekt peker på.

Noen steder som inneholder interessante xml filer som kan prøves:

<http://svn.bitflux.ch/repos/public/slideml/slideml.xml>

Du kan alternativt benytte, i stedet for SimpleXML:

```
$dom = new DOMDocument();  
$dom->load('https://username:parola@url.com/xml.xml');
```

<http://www.nytimes.com/services/xml/rss/userland/HomePage.xml>

Et annet alternativ er å bruke libxml_use_internal_errors, f.eks. som følger:

```
<?php  
libxml_use_internal_errors(true);  
$sxe = simplexml_load_string("<?xml version='1.0'><broken><xml></broken>");  
if ($sxe === false) {  
    echo "Failed loading XML\n";  
    foreach(libxml_get_errors() as $error) {  
        echo "\t", $error->message;  
    }  
}  
?>
```

Se <http://no2.php.net/manual/en/simplexml.examples-errors.php>

Sammenligning mellom SimpleXML og DOMDocument

SimpleXML		DOMDocument	
Funksjon		Funksjon	
simplexml_load_file	lager et SimpleXMLElement fra en fil	load	laget en DOMDocument node fra en fil
simplexml_load_string	lager et SimpleXMLElement fra en en string	loadXML	laget en DOMDocument node fra en string
simplexml_import_dom	lager et SimpleXMLElement fra en DOM node (som er superklassen til DOMDocument!)		
asXML (saveXML, alias)	Lag en string av et SimpleXMLElement. Hvis funksjonen kalles med en parameter, er dette et filnavn som får resultatet.	saveXML	Lag en string av det DOMDocument funksjonen anvendes på. En parameter vil angi at stringen skal lages fra en node.
		save	Lag en fil av det DOMDocument funksjonen anvendes på. Parameter angir filnavn.
addAttribute		createAttribute	

PHP kode som lager et DOM document og bruker det videre i simpleXML:

```
$domDocument = new DOMDocument('1.0', "UTF-8");
// Kode for å manipulere dette objektet, før behandling med simpleXML ligger her
$smple = simplexml_import_dom($domDocument);
// Nå har du et SimpleXMLElement objekt, og du kan f.eks. benytte XSLTProcessor.
```