

## **Tehnologii bazate pe XML: XML-RPC**

**Lect. dr. Teodor Florin Fortiș**  
Universitatea de Vest Timișoara  
**Asist. drd. Alexandra Fortiș**  
Universitatea “Tibiscus” Timișoara

**ABSTRACT.** XML documents offers an ideal support for transporting data between different applications, even remotely. RPC offers the necessary support for calling remote procedures. By combining these two technologies we have the possibility of making complex remote calls. In this paper we offer a point of view on XML-RPC based applications, following the client-server model.

### **1 Introducere**

În acest articol este urmărită prezentarea modului în care poate fi realizată integrarea tehnologiilor bazate pe XML în programele PHP. În timp ce XML oferă posibilități variate de codificare a documentelor în scopul prelucrării ulterioare a structurilor de date conținute în acestea, limbajul PHP oferă suportul independent de platformă pentru transportul datelor depozitate în documentele XML. RPC a fost dezvoltată pentru a permite aplicațiilor să comunice între ele, indiferent de limbajul sursă al acestora sau locul în care rulează acestea.

#### **1.1. Informații de bază XML**

XML (Extensible Markup Language) este un subset al limbajului SGML (Standard Generalized Markup Language). Acesta din urmă a fost definit în standardul ISO8879 și a fost realizat pentru a ușura schimbul de documente structurate pe WEB. Spre deosebire de HTML, fișierele XML marchează

clar punctul de start și punctul final al fiecărei părți logice dintr-un asemenea document. Fiecare dintre aceste părți logice se numește element.

XML restricționează utilizarea construcțiilor SGML pentru a se asigura împotriva unor situații în care accesul către anumite componente ale documentului nu este posibil. De asemenea, este definit modul în care URL-urile pot fi utilizate pentru a identifica anumite componente ale streamurilor de date XML.

Prin definirea rolului fiecărui element de text într-un model formal, cunoscut sub numele DTD (Document Type Definition), utilizatorii XML pot verifica faptul că fiecare componentă a unui document apare într-o poziție validă în cadrul unui stream de date utilizat pentru schimbul de informații. XML

DTD permite de asemenea calculatorului să verifice, de exemplu, ca utilizatorii să nu utilizeze accidental anumite construcții interzise.

Astfel de verificări nu sunt posibile atunci când este utilizat un limbaj gen HTML, limbajul clasic utilizat pentru a codifica documentele accesibile pe WEB.

Spre deosebire de SGML, XML nu cere prezența unui DTD. În lipsa acestuia, sistemul XML va utiliza o definiție implicită pentru fiecare dintre componentele nedeclarate din documentul curent.

Utilizatorii vor putea folosi XML pentru:

- utilizarea de fișiere multiple pentru formarea de documente compuse;
- identificarea locului în care urmează să fie incorporate ilustrațiile în fișierele text și modul în care este codificată fiecare ilustrație;
- oferirea de informații de control a procesării programelor suport cum ar fi browserele;
- adăugarea de comentarii editoriale fișierelor.

Totuși, este important de notat ce nu este și ce nu oferă XML:

- Un set predefinit de etichete, de tipul celor definite pentru HTML, care pot fi utilizate pentru marcarea documentelor;
- Un șablon standardizat pentru producerea unor anumite tipuri de documente.

XML nu a fost creat pentru a standardiza modalitatea de codificare a textului; o asemenea operație ar fi de fapt imposibilă, atâta vreme cât ar fi de așteptat crearea unei unice scheme de codificare care se va potrivi cu toate limbajele și aplicațiile.

XML poate fi privit mai degrabă ca un limbaj formal care poate fi utilizat pentru a transmite informațiile despre părțile componente ale unui

document către un alt sistem de calcul. XML oferă de fapt destulă flexibilitate pentru a putea descrie orice structură logică a unui text.

### 1.1.1. Componentele XML

XML presupune că fiecare document este format dintr-o serie de entități. Fiecare entitate poate conține una sau mai multe elemente logice, fiecare dintre aceste elemente logice fiind dotat cu anumite proprietăți (atribute) care descriu modul în care urmează să fie procesat documentul.

XML oferă o sintaxă formală pentru descrierea relațiilor dintre entități, elemente și atributele care fac parte dintr-un document XML, putând fi utilizate pentru a comunica sistemului de calcul modul în care pot fi recunoscute diferitele componente ale unui document.

Spre deosebire de alte limbaje de marcare a documentelor, XML nu se rezumă la precizarea modificărilor de stil sau la începutul elementelor noi. Spre deosebire de acestea, XML precizează cu claritate limitele fiecărei părți a unui document, indiferent de tipul acesteia. În plus, pentru a permite calculatorului să verifice structura unui document, este necesară oferirea unui DTD în care sunt declarate entitățile permise, elemente și atributele asociate, dar și relațiile dintre toate acestea.

Etichetele XML fiind bazate pe structura logică a documentului, acestea vor fi ceva mai ușor de înțeles în contextul documentului curent decât alte scheme de marcare a documentelor. Puteți codifica un memo (sau chiar un mesaj de poștă electronică) în XML în felul următor:

```
<memo>
  <catre>Toți cititorii</catre>
  <dela>Florin Fortiș</dela>
  <data>26 Noiembrie 2002</data>
  <subiect>Bibliografia</subiect>
  <text>Trimiteți bibliografiile până luni,
    vă rog.</text>
</memo>
```

Observați că fiecare element logic al documentului din acest fișier este marcat clar printr-o pereche de etichete de început/sfârșit (cum ar fi <subiect>...</subiect>). Observați, de asemenea, că nu există nici un fel de indicație asupra documentului final. Modul în care urmează să fie utilizată informația astfel codificată este lăsat la alegerea utilizatorului. Acest document va putea la fel de bine să fie afișat pe ecran, tipărit într-o formă prestabilită sau utilizat mai departe într-o altă aplicație.

### 1.1.2. Utilizarea textului XML

Un fișier XML va conține trei tipuri de marcaje, după cum urmează:

- Instrucțiunea de procesare XML. Aceasta identifică versiunea de XML utilizată, codificarea sau precizează dacă acest fișier se referă la alte fișiere. Acest marcaj este opțional.

```
<?xml version="1.0" encoding="UCS2" standalone="yes">
```

- Instrucțiuni referitoare la DTD (DTD nu face obiectul acestui raport). Această zonă va conține chiar declarațiile formale sau o referință către un fișier care conține DTD.

```
<!DOCTYPE memo SYSTEM "http://www.myco.com/dtlds/memo.dtd">
```

- instanță a unui document complet etichetat, care este format dintr-un element rădăcină, identic cu cel desemnat în DTD ca element de bază. Toate celelalte elemente se vor găsi în interiorul acestui element.

Un document este valid când sunt prezente toate cele trei componente iar instanța documentului se conformează regulilor definite în DTD. Atunci când este prezentă doar ultima dintre cele trei componente, un procesor XML va fi capabil doar să verifice dacă instanța documentului este bine construită, adică dacă fiecare element este corect precizat și dacă atributele acestora sunt însoțite de valori, acestea din urmă fiind prezentate între ghilimele.

Fișierele XML fiind orientate către obiecte și ierarhice, acestea devin candidați potriviți pentru depozitarea unor baze de date. Totuși, uneori trebuie luate măsurile necesare pentru a depozita suficiente informații de structură în baza de date pentru a putea reconstitui corect datele originale. Fișierele XML sunt capabile să identifice poziția fiecărei bucăți de date atât în structura documentului final cât și înaintea utilizării propriu-zise a acestora.

Depozitarea datelor într-un format bine definit prin intermediul XML asigură transferul corect al datelor către o mare varietate de platforme hard sau soft, fără a fi nevoie de o modificare a structurii logice a documentului. În plus, adăugarea unor atribute va presupune doar împerspătarea modelului pentru a include noile atribute ale bazei de date și inserarea marcajelor potrivite în instanța documentului.

## 1.2. Ce este XML-RPC

Tehnologia XML-RPC a fost dezvoltată pentru a permite aplicațiilor să comunice între ele, indiferent de limbajul sursă al acestora sau locul în care rulează acestea. Dezvoltat inițial de UserLand Software, în 1998, această tehnologie a devenit un punct important în cadrul altor tehnologii, incluzând Microsoft .NET sau Apple Mac OS X.

Prin intermediul XML-RPC este posibil ca aplicații XML-RPC să apeleze metode ale altor aplicații XML-RPC, astfel încât o aplicație PHP va fi capabilă să apeleze o funcție C sau o metodă Python. Prin urmare, XML-RPC oferă posibilitatea comunicării unor aplicații eterogene fără eforturi deosebite din partea programatorilor.

## 2. Utilizarea XML-RPC în PHP

### 2.1. Apeluri XML-RPC

Un apel XML-RPC este transmis de un client către server prin intermediul protocolului HTTP, o cerere de tip POST. În acest mod aplicațiile XML-RPC pot funcționa și comunica extrem de simplu chiar și din spatele unui firewall, având în vedere că portul pe care se bazează comunicarea este, în general, totdeauna deschis.

Un apel XML-RPC este încapsulat într-o etichetă <methodCall>, care conține toate informațiile legate de identificarea funcției sau metodei apelate la distanță, împreună cu eventualii parametri care urmează să fie utilizați. Parametrii sunt precizați între etichete purtând numele tipului de date al acestora, valorile posibile cuprinzând: <int>...</int>, <string>...</string>, <boolean>...</boolean>, <double>...</double>. În plus, pot fi construite structuri sau tablouri, situație în care elementele acestor structuri de date vor fi precizate, din nou, prin etichete de tip.

Iată un exemplu simplu de apel XML-RPC, primele linii caracterizând cererea (POST) HTML:

```
POST xmlrpc_server.php HTTP/1.0
User-Agent: xmlrpc-epi-php/0.2 (PHP)
Host: localhost:80
Content-type: text/html
Content-length: ...
<?xml version="1.0" encoding="iso-8859-1" ?>
<methodCall>
```

```
<methodName>hello</methodName>
<params>
<param>
<value>
<string>World!</string>
</value>
</param>
</params>
</methodCall>
```

## 2.2. Răspunsuri XML-RPC

Răspunsul la o cerere XML-RPC va sosi într-un format similar. De data aceasta, elementul <methodResponse> înglobează întreaga informație cuprinsă în răspuns. Răspunsul este, la rândul lui, construit în același stil cu apelul unei metode, un exemplu fiind destul de concludent:

```
HTTP/1.1 200 OK
Connection: close
Content-length: ...
Content-type: text/html
Date: Tue, 13 Jun 2004 13:13:13 GMT
Server: Apache/1.3.28
<?xml version="1.0" encoding="iso-8859-1" ?>
<methodResponse>
<params>
<param>
<value>
<string>Hello, indeed!</string>
</value>
</param>
</params>
</methodResponse>
```

## 3 XML-RPC: client - server

Tehnologia XML-RPC presupune existența a două tipuri de aplicații înrudite: un server și unul sau mai mulți clienți. Filosofia aplicațiilor XML-RPC este extrem de simplă: clientul formulează cereri în timp ce serverul procesează aceste cereri și emite răspunsurile necesare.

### 3.1. Un server XML-RPC

Implementarea PHP pentru XML-RPC oferă metodele necesare pentru crearea acestuia și apelul metodelor, precum și pentru "înregistrarea" metodelor pentru ca acestea să poată fi apelate de un client. Orice metodă care urmează să fie apelată prin intermediul XML-RPC trebuie să fie înzestrată cu trei parametri:

- numele metodei XML-RPC apelată,
- referință către un tablou conținând parametri transmiși din partea clientului,
- date suplimentare transmise acestei metode.

```
<?php
function hello_func($method_name, $params, $app_data) {
    $name=$params[0] ;
    return ="Hello, $name!" ;
}
$xmlrpc_server = xmlrpc_server_create () ;
xmlrpc_server_register_method($xmlrpc_server, "hello",
"hello_func") ;
$request_xml = $HTTP_RAW_POST_DATA ;
$response = xmlrpc_server_call_method ($xmlrpc_server,
$request_xml, '' ) ;
print $response ;
?>
```

### 3.2. Un client XML-RPC

Crearea unui client este ușurată de existența unui pachet de funcții care sosesc odată cu pachetul în care este definită extensia XML-RPC pentru PHP.

Întreaga muncă legată de descifrarea răspunsului oferit de server este rezolvată prin funcția `xu_rpc_http_concise()`.

```
<?php
include (xmlrpcutils/utils.php) ;
$host = "my.local.host.name" ;
$uri = "xmlrpc_server.php";
$name = "myself" ;
$request_xml = $HTTP_RAW_POST_DATA ;
$response = xu_rpc_http_concise( array (
    '`method'`=>"hello",
    '`args'`=>array($name),
    '`host'`=>"$host",
```

```
'`uri'`=>"uri",  
'`port'`=>80)) ;  
print $response ;  
?>
```

### **Bibliografie**

- [WWWa] XML-RPC.com, <http://www.xmlrpc.com>
- [WWWb] Manual PHP, <http://www.php.net/manual>
- [WWWc] xmlrpc API, [http://xmlrpc-epi.sourceforge.net/main.php?t=php\\_api](http://xmlrpc-epi.sourceforge.net/main.php?t=php_api)
- [WWWd] W3C DOM, <http://www.w3.org/DOM>
- [WWWe] PHP DOM XML, <http://www.php.net/manual/en/ref.xml.php>
- [WWWf] PHP XML, <http://www.phpxml.org>
- [WWWg] PHP DOM, <http://devil.medialab.at>