

XML-RPC: Zdalne wykonywanie procedur

Bartłomiej Świercz

Katedra Mikroelektroniki i Technik Informatycznych

Łódź, 28 października 2005 roku

Internet dostarcza wiele możliwości programistą piszącym aplikacje komunikujące się pomiędzy **różnymi** komputerami i systemami. Zazwyczaj projektowanie **rozproszonego** oprogramowania jest procesem złożonym, szczególnie jeżeli oprócz aplikacji projektuje się również protokół komunikacji.

Obecnie jednak podstawowym pytaniem podczas projektowania aplikacji **rozproszonej** jest pytanie, który sposób komunikacji (jaką bibliotekę) należy wybrać.

Internet dostarcza wiele możliwości programistą piszącym aplikacje komunikujące się pomiędzy **różnymi** komputerami i systemami. Zazwyczaj projektowanie **rozproszonego** oprogramowania jest procesem złożonym, szczególnie jeżeli oprócz aplikacji projektuje się również protokół komunikacji.

Obecnie jednak podstawowym pytaniem podczas projektowania aplikacji **rozproszonej jest pytanie, który sposób komunikacji (jaką bibliotekę) należy wybrać.**

Dwa najbardziej popularne protokoły (framework) to:

- CORBA
- DCOM

Obydwie technologie są bardzo silne i pozwalają na zbudowanie dowolnej architektury połączeń. Są jednak **skomplikowane**.

XML-RPC – XML-Remote Procedure Call

XML-RPC jest technologią zdalnego wykonywania procedur. Jest to prosta, przenośna i niezależna od języka programowania technologia komunikacji rozproszonych systemów wykorzystująca protokół **HTTP** jako kanał transportowy język XML do opisu danych.

`http://www.xml-rpc.org`

Hypertext Transfer Protocol jest protokołem warstwy aplikacji przeznaczonym dla rozproszonych i zorientowanych na media systemów informacyjnych. Jest to protokół bardzo ogólny, **bezstanowy**, który może być wykorzystany do dystrybucji informacji hipertekstowej oraz jako **serwer rozproszonych zorientowanych obiektowo systemów (poprzez rozszerzenie metody "request", kodów błędów i nagłówek)**.

`http://www.w3.org/Protocols/rfc2616/rfc2616.html`

Przykład zapytania do serwera HTTP

```
GET / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US;
rv:1.7.10) Gecko/20050911 Firefox/1.0.6 (Debian
package 1.0.6-5)
Accept: text/xml, application/xml,
application/xhtml+xml, text/html;q=0.9,
text/plain;q=0.8,image/png, */*;q=0.5
Accept-Language: pl,en-us;q=0.7,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-2,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

Przykład odpowiedzi serwera HTTP

```
HTTP/1.1 302 Found
Date: Fri, 28 Oct 2005 05:39:29 GMT
Server: Apache/2.0.54 (Debian GNU/Linux)
PHP/4.3.10-15
Location: http://localhost:8000/apache2-default/
Content-Length: 323
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
```


Komunikat XML-RPC jest realizowany jako rządanie HTTP-POST. Treścią wiadomości jest struktura XML. Procedura wykonywana jest na serwerze i wynik jest zwracany jako poprawnie sformatowany dokument XML.

Parametrem procedury mogą być liczby, napisy, struktury i tablice.

Przykład zapytania XML-RPC

POST / HTTP/1.0

Host: localhost:9000

User-Agent: xmlrpcclib.py/1.0.1

Content-Type: text/xml

Content-Length: 187

<?xml version='1.0'?>

<methodCall>

 <methodName>pow</methodName>

 <params>

 <param>

 <value><int>2</int></value>

 </param>

 <param>

 <value><int>8</int></value>

 </param>

 </params>

Przykład odpowiedzi XML-RPC

```
HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.3.5
Date: Wed, 26 Oct 2005 11:44:59 GMT
Content-type: text/xml
Content-length: 123
<?xml version='1.0'?>
<methodResponse>
  <params>
    <param>
      <value><int>256</int></value>
    </param>
  </params>
</methodResponse>
```

Przykład klienta XML-RPC

```
import xmlrpclib

server = xmlrpclib.Server ("http://localhost:8000/")
print server.pow (2, 8)
```

Przykład serwera XML-RPC

```
from SimpleXMLRPCServer import SimpleXMLRPCServer

server = SimpleXMLRPCServer(("localhost", 8000))
server.register_function(pow)
server.serve_forever()
```

Znacznik:	Opis:
<code><i4></code> lub <code><int></code>	czterobajtowa liczba całkowita ze znakiem
<code><boolean></code>	wartość logiczna: 0 (fałsz) lub 1 (prawda)
<code><string></code>	napis
<code><double></code>	liczba zmiennoprzecinkowa podwójnej precyzji
<code><dateTime.iso8601></code>	format daty i czasu: 19980717T14:08:55
<code><base64></code>	wartość binarna zakodowana algorytmem base64

```
<struct>  
  <member>  
    <name>lowerBound</name>  
    <value><i4>18</i4></value>  
  </member>  
  <member>  
    <name>upperBound</name>  
    <value><i4>139</i4></value>  
  </member>  
</struct>
```

```
<array>
  <data>
    <value><i4>12</i4></value>
    <value><string>Egypt</string></value>
    <value><boolean>0</boolean></value>
    <value><i4>-31</i4></value>
  </data>
</array>
```


Przećwiczmy to w praktyce!