

► Por: Pablo Dall'Oglio

No site oficial do PHP (www.php.net) você encontra informações sobre o uso de Web Services

The screenshot shows the official PHP website. The main content area features a section titled 'php|tropics' with a sub-header 'php|tropics' and a date '16/04/2005'. The text describes the PHP Tropics conference, mentioning it was held at the Moon Palace All-inclusive Resort in Cancun, Mexico. It highlights that the conference is a 5-day event with sessions and hands-on tutorials by leading PHP experts. A special early-bird discount is mentioned, and attendees are encouraged to sign up before February 28. The page also includes a 'Thanks To' section, 'Related sites', 'Community', and 'Syndication' links. A sidebar on the right lists 'Upcoming Events' for January and 'User Group Events' in various cities like Hamburg, Dallas, and New York.

Os Webservices têm a missão de integrar serviços diferentes por meio de uma plataforma Web. Entenda como isso é possível

WebServices no

WebServices são um conjunto de tecnologias independentes de plataforma, que permitem interoperabilidade através da entrega de serviços publicados e invocados pela Internet para utilização imediata. Ou mesmo composição de novos serviços integrando tudo por meio de padrões como XML, SOAP, WSDL e UDDI.

Ao longo dos anos, a promessa de interoperabilidade tem sido ilusória. Presenciamos o surgimento de padrões como DCOM e CORBA, e mais recentemente, o de Webservices.

A promessa é de que os WS (Webservices) integrarão PCs, dispositivos, bancos de dados e redes em uma única plataforma virtual, a Internet, movendo funcionalidades desde aplicações e dados do desktop para servidores Web. Estes, por sua vez, passarão a controlar segurança, privacidade e acessibilidade, lidando com aplicações complexas e transações críticas.

Ao contrário de tecnologias como Corba, que utilizavam um protocolo específico, WS provêem interoperabilidade por meio de XML e da utilização de padrões abertos como SOAP, que é um

PHP

protocolo para a troca de informações em um ambiente descentralizado pelo protocolo de comunicação HTTP e o formato XML. Para publicar e para descobrir os WS na Internet, as aplicações utilizam a especificação UDDI (Universal Description Discovery and Integration). Para descrever os serviços, é utilizada a linguagem WSDL (Web Services Description Language), baseada no formato XML, que descreve um WS, suas interfaces e os mecanismos de interação.

► Arquitetura

Por meio da figura no topo da página, temos um panorama geral do

funcionamento de WebServices, por meio de um conjunto de tecnologias de padrão aberto interagindo sob uma plataforma de Internet. Temos o papel do *Requerente*, que é a aplicação que interage com o *Serviço*, podendo ser desde um Desktop ou Servidor até um Celular ou Palmtop. Temos o servidor de aplicação, que provê o serviço e se comunica com o Requerente pela camada e pacotes SOAP. E temos o Registro dos WS codificado em WSDL, publicado e descoberto pela especificação UDDI.

Uma das características mais atraentes na adoção de WS é a possibilidade de criar novas aplicações baseadas em

componentes de sistemas legados, preservando investimentos, possibilitando interoperabilidade, disponibilidade no ambiente Web e uma oportunidade para as empresas reverem processos de negócio.

► PHP e Webservices

Para demonstrar uma pequena aplicação utilizando Webservices, utilizaremos a linguagem PHP. As vantagens em utilizar PHP no ambiente Web são sua capacidade de Orientação a Objetos, classes para parser (SAX, DomXML e SimpleXML) e geração de XML, diversas extensões para as mais diversas finalidades, além de bibliotecas para comunicação utilizando protocolos como HTTPS, XML-RPC, LDAP, entre outros. Além de sua sintaxe ser muito similar à de linguagens como Java e C, conseguiremos implementar Webservices por meio de poucas linhas de código.

Para demonstrar o funcionamento de WS em PHP, utilizaremos as classes nativas para manipulação do protocolo SOAP, desenvolvido por Dmitry Stogov. Por meio da utilização de SOAP em PHP, poderíamos escrever um código em PHP que envia uma pesquisa (encapsulada em XML pelo SOAP) para uma aplicação de Banco de Dados em C++ localizada em outro continente para obter o preço de um livro, por exemplo.

O funcionamento é simples: a transação se inicia quando a aplicação cliente realiza a chamada remota de uma função enviando um pacote SOAP que contém a descrição do método a ser invocado via HTTP. O servidor recebe o pacote SOAP, interpreta-o, executa a função correspondente e retorna a resposta da execução também encapsulada via SOAP. Para demonstração, construiremos um exemplo de WebServices em PHP para simular a requisição de dados de um cliente. Sendo que os dados estão armazenados em um banco de dados PostgreSQL ao lado do servidor.

Para construir este Webservice em PHP, o primeiro passo é criar o arquivo WSDL (Web Services Description Language), que irá descrever o funcionamento do Web Service. A seção *message* descreve processos de requisição (*request*) e resposta (*response*) de uma determinada

funcionalidade do Webservice, que neste caso é a função *getNome*. Cada processo possui seus parâmetros, bem como os tipos de dados. A seção *portType* define uma funcionalidade do Webservice e indica quais processos de requisição e resposta serão utilizados para tal. Neste caso, *getNomeRequest* para *input* e *getNomeResponse* para *output*. A seção *binding* define como as mensagens devem ser transmitidas e codificadas. Neste caso, via RPC usando SOAP sobre HTTP. E por fim, a seção *service* define a URL na qual o serviço está rodando. Neste caso, no servidor Apache local.

Listagem: exemplo.wsdl

```
<?xml version='1.0'
encoding='ISO-8859-1' ?>
<definitions name='Exemplo'
targetNamespace='http://example.org/Exemplo'
xmlns:tns='http://example.org/Exemplo'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wSDL='http://schemas.xmlsoap.org/wsdl/'
xmlns='http://schemas.xmlsoap.org/wsdl/'>
<message
name='getNomeRequest'>
<part name='codigo'
type='xsd:string' />
</message>
<message
name='getNomeResponse'>
<part name='resultado'
type='xsd:string[]' />
</message>
<portType
name='ExemploPortType'>
<operation name='getNome'>
<input message='tns:
getNomeRequest' />
<output message=
'tns:getNomeResponse' />
</operation>
</portType>
<binding
name='ExemploBinding'
```

```
type='tns:ExemploPortType'>
<soap:binding style='rpc'
transport='http://schemas.xml
soap.org/soap/http' />
<operation name='getNome'>
<soap:operation
soapAction='exemplo#getNome'
/>
<input>
<soap:body
use='encoded' namespace=
'exemplo' encodingStyle=
'http://schemas.xmlsoap.
org/soap/encoding/' />
</input>
<output>
<soap:body
use='encoded'
namespace='exemplo'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
</output>
</operation>
</binding>
<service
name='ExemploService'>
<port name='ExemploPort'
binding='ExemploBinding'>
<soap:address
location='http://127.0.0.1/s
ervidor.php' />
</port>
</service>
</definitions>
```

O programa servidor recebe o código de um cliente, consulta seus dados no banco de dados e retorna todas as informações em forma de vetor. Caso contrário, lança os devidos erros (pelo *throw*), comentados mais adiante.

Listagem: servidor.php

```
<?php
function getNome($codigo)
{
// verifica a passagem
do parâmetro
if (!$codigo)
{
throw new
SoapFault('Client',
'Parametro nao preenchido');
}
}
```

```
// conecta ao Banco
de Dados
$id = @pg_connect
("dbname=samples
user=postgres");
if (!$id)
throw new
SoapFault("Server", "Conexao
nao estabelecida");

// realiza consulta ao
Banco de Dados
$result = pg_query($id,
"select * from clientes " .
"where codigo=$codigo");
$matriz = pg_fetch_all
($result);
if ($matriz == null)
throw new
SoapFault("Server", "Cliente
nao encontrado");

// retorna os dados.
return $matriz[0];
}

// instancia servidor SOAP
$server = new
SoapServer("exemplo.wsdl",
array('encoding'=>'ISO-8859-
1'));
$server->addFunction
("getNome");
$server->handle();
?>
```

A seguir, temos o código da aplicação cliente, que estabelece conexão com o servidor da aplicação (*servidor.php*, listado



Para saber mais sobre as funcionalidades do PHP, visite o site oficial www.php.net

acima), informa os parâmetros da função remota e exibe os dados de retorno em formato HTML (no browser).

Listagem: cliente.php

```
<?php
// instancia cliente SOAP
$client = new
SoapClient("exemplo.wsdl",
array('encoding'=>'ISO-8859-
1'));
try
{
// realiza chamada
remota de método
$resultado = $client-
>getNome(3);
// Imprime os dados
de retorno
echo '<table
border=1>';
echo '<tr
bgcolor=gray><td>Coluna
</td> <td> conteúdo
```

```
</td></tr>';
echo '<tr><td>Código
</td> <td> ' .
$resultado['codigo'] .
'</td></tr>';
echo '<tr><td>Nome
</td> <td> ' .
$resultado['nome'] .
'</td></tr>';
echo '<tr><td>Telefone
</td> <td> ' .
$resultado['telefone'] .
'</td></tr>';
echo '<tr><td>Rua
</td> <td> ' .
$resultado['rua'] .
'</td></tr>';
echo '<tr><td>Idade
</td> <td> ' .
$resultado['idade'] .
'</td></tr>';
echo '</table>';
}
catch (SoapFault $excecao)
// ocorrência de erro
{
echo "Erro: ";
echo "<b> {$excecao-
>faultstring} </b>";
}
?>
```

Caso não ocorram erros, serão exibidos no browser, os dados requisitados:

Coluna	Conteúdo
Código.....	3
Nome.....	PabloDall'Oglio
Telefone.....	55 3434-9595
Rua.....	Rua Conceicao
Idade.....	24

Caso o parâmetro do método remoto não seja informado, a mensagem de erro retornada será a seguinte:

Erro: Parâmetro não preenchido

Caso o banco de dados não seja conectado devidamente, a mensagem de erro retornada será a seguinte:

Erro: conexão não estabelecida

Agora que você já sabe o que são Webservices e a forma de usá-los no PHP, basta estudar mais sobre o assunto. Uma boa pedida para tal é visitar o site <http://webservicexml.com>.

O Site webservicexml.com tem muita documentação sobre os Webservices além de fóruns de discussão sobre tecnologia voltada à Internet