

Serviço de primeira

Veja como o PHP-GTK facilita a criação de clientes gráficos para qualquer Web Service.
por Pablo Dall'Oglio

No artigo anterior[1] estudamos como armazenar as informações de nossa aplicação PHP-GTK em um banco de dados *SQLite*. No quinto e último

artigo da série, veremos as formas de trocar informações entre aplicações distantes geograficamente, por meio de *Web Services*. Para tal, vamos criar uma aplicação cliente em PHP-GTK que se comunique com um servidor rodando *Apache* e *PHP*. A nossa aplicação cliente PHP-GTK irá oferecer ao usuário uma interface solicitando a execução de um método da aplicação servidora, que também será escrita em PHP. Antes de escrever o programa, vamos explicar alguns conceitos.

componentes de sistemas legados, um serviço online de consulta a cotações da bolsa, uma aplicação cliente para um serviço de consulta a entrega dos correios, uma aplicação cliente para um sistema remoto de registro de horas no ponto ou até um sistema de chat, com um programa cliente que conversa com uma aplicação servidora responsável por registrar as mensagens de todos os participantes. Enfim, as possibilidades são inúmeras.

Na **figura 1** temos uma representação simplificada da aplicação que iremos construir neste artigo. Ambas aplicações cliente e servidora devem rodar em um dispositivo que permita uma conexão à Internet, por meio do protocolo TCP/IP. A aplicação cliente faz uma requisição à servidora por meio de um pacote XML, de acordo com o protocolo SOAP, e em seguida a aplicação servidora processa essa requisição, de acordo com suas regras internas, devolvendo a resposta para a aplicação cliente, também no formato de um pacote XML.

Web Services

Web Services representam uma tecnologia que permite a aplicações comunicarem-se umas com as outras, combinando funcionalidades de forma independente de plataforma ou linguagem. Os Web Services são disponibilizados pela Internet por meio de um conjunto de tecnologias como XML e SOAP (*Simple Object Access Protocol*), que permitem interoperabilidade pela publicação de serviços.

Web Services podem ser utilizados para diversos fins, como criar novas aplicações baseadas em

Aplicação Cliente

O PHP5 possui classes nativas que dão suporte ao protocolo SOAP. Vamos utilizar tais classes para criar nossa aplicação, que será extremamente simples. Ela será formada por uma aplicação cliente, na qual o usuário informará o código de uma pessoa e, ao clicar no botão *Buscar*, um Web Service será acionado. Basicamente, esse Web

Exemplo 1: Aplicação cliente

```
<?php
// cria a janela
$janela = new GtkWindow;
$janela->set_size_request(200,120);
// cria os campos
$codigo = new GtkEntry;
$nome = new GtkEntry;
$nome->set_sensitive(FALSE);
// cria o botão
$botao = new GtkButton('Buscar');
$botao->connect('clicked', 'onBuscar');
// coloca campos na caixa vertical
$vbox = new GtkVBox;
$vbox->add(new GtkLabel('Código'));
$vbox->add($codigo);
$vbox->add($botao);
$vbox->add(new GtkLabel('Nome'));
$vbox->add($nome);
// exibe a janela
$janela->add($vbox);
$janela->show_all();
// aciona o web service
function onBuscar()
{
    global $codigo, $nome;
    // declara parametros do WS
    $param=array();
    $param['encoding'] = 'ISO-8859-1';
    $param['location'] =
        'http://127.0.0.1/server.php';
    $param['exceptions'] = TRUE;
    $param['uri'] = 'http://test-uri/';
    // instancia cliente SOAP
    $client=new SoapClient(NULL, $param);
    // executa o método remoto
    $id = $codigo->get_text();
    $resp=$client->buscar($id);
    $nome->set_text($resp['nome']);
}
Gtk::Main();
?>
```

Exemplo 2: Aplicação servidora

```
<?php
function buscar($codigo)
{
    // cria a string sql
    $sql = "select nome from pessoas ".
           "where id='$codigo'";
    // conecta ao banco
    $con = pg_connect('host=localhost
                      dbname=linuxmag
                      user=postgres');
    // realiza a consulta
    $res = pg_query($con, $sql);
    // retorna o resultado
    return pg_fetch_array($res);
}
// declara parametros do WS
$params=array();
$params['encoding'] = 'ISO-8859-1';
$params['uri']       = 'http://test-uri/';
// instancia servidor SOAP
$server=new SoapServer(NULL, $params);
// adiciona a função ao WS
$server->addFunction('buscar');
// começa a "ouvir" requisições
$server->handle();
?>
```

Service irá comunicar-se com a aplicação servidora, em que existe um banco de dados de pessoas. A aplicação servidora irá buscar em seu banco de dados interno qual registro possui aquele código, e irá retorná-lo para a aplicação cliente. Então, a aplicação cliente exibirá o nome da pessoa retornada logo abaixo do botão *Buscar*.

Veja a aplicação cliente na **figura 2**. Nela, criamos uma janela e, dentro dessa janela, colocamos uma caixa vertical (**exemplo 1**). Dentro da caixa vertical colocamos nossos objetos `GtkLabel` (Código, Nome), dois `GtkEntry` (`$codigo`, `$nome`) e um botão para buscar o registro. Tornamos o campo para digitação do nome

não-editável, pois ele irá apresentar somente o resultado da pesquisa ao servidor.

Quando o botão *Buscar* for clicado, o método `onBuscar()` será executado. Sua função é acionar o Web Service por meio da classe do PHP `SoapClient`. Essa classe disponibiliza para a aplicação um objeto cujos métodos são exatamente os mesmos disponibilizados pelo servidor do Web Service. Aqui estamos executando o método `buscar()`, passando

como parâmetro o *ID* da pessoa. Veja que no parâmetro `location`, passado ao criar o `SoapClient`, especificamos a localização do servidor do Web Service. No **exemplo 1** é um endereço local, mas poderia ser qualquer endereço IP.

Aplicação Servidora

A aplicação servidora (**exemplo 2**) irá responder à aplicação cliente. Para disponibilizar métodos para a aplicação cliente, utilizamos a classe `SoapServer`. Essa classe possui o método `addFunction()`, que adiciona funções ao Web Service. Essas funções são disponibilizadas como métodos do objeto `SoapClient`.

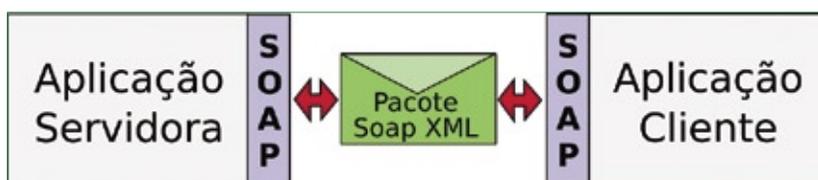


Figura 1 Programa em execução.



Figura 2 Representação de Web Services.

No lado da aplicação servidora, existirá um banco de dados em PostgreSQL contendo uma tabela de pessoas, em que temos campos como `id` e `nome`. O papel do método `buscar()` é receber o código de uma pessoa e retornar o registro da pessoa, composto por seu nome, à aplicação cliente.

Conclusão

Neste último artigo da série vimos como trocar informações entre aplicações por meio de Web Services. Com isso, completamos nossos objetivos, que eram informar o leitor sobre aspectos diferentes relacionados ao PHP-GTK, para que ele pudesse buscar maiores informações acerca de um assunto específico, quando necessário. Espero que a leitura tenha sido agradável e convido a todos a participarem de nossa comunidade virtual PHP-GTK Brasil [2]. ■

Mais informações

[1] Pablo Dall'Oglio, "Fale com o banco": http://www.linuxmagazine.com.br/article/fale_com_o_banco

[2] PHP-GTK Brasil: <http://www.php-gtk.com.br>

[3] Livro PHP-GTK: <http://www.php-gtk.com.br/book>

[4] Site do Autor: <http://www.pablo.blog.br>