

FreeSWITCH XML-RPC

Test program for connecting to the FreeSWITCH XML-RPC interface.

Ruby Example

```
#!/usr/bin/env ruby

require 'xmlrpc/client'
require 'xmlrpc/marshal'

exit unless ARGV[0]

port = 8080
server = '192.168.190.248'
directory = '/RPC2'

def fixup(value)
  value.gsub(/&lt;/, "<").
  gsub(/&gt;/, ">")
end

server = XMLRPC::Client.new(server, directory, port, nil, nil, 'freeswitch', 'works', nil, 10)
marshal = XMLRPC::Marshal.new()

puts "Executing command #{ARGV.join(' ')}"
result = server.call("freeswitch.api", "#{ARGV.shift}", "#{ARGV.join(' ')}")

puts fixup(marshal.dump_response(result))
```

PHP Example

Uses the xmlrpc-2.1 library from <http://phpxmlrpc.sourceforge.net/>

```

<?php

//
// Kludged together by Ken Rice - SwK on #freeswitch/freenode
//

include("xmlrpc.inc");

$username = "freeswitch";
$password = "works";

// Play nice to PHP 5 installations with REGISTER_LONG_ARRAYS off

$stateno=(integer)$HTTP_POST_VARS["stateno"];
$f=new xmlrpcmsg('freeswitch.api',
    array(new xmlrpcval("show", "string"),new xmlrpcval("channels", "string"))
);
$c=new xmlrpc_client("/RPC2", "localhost", 8080);
$c->setCredentials($username,$password,NULL);

// $c->setDebug(2); // Uncomment a Value of 1 Outputs Received XML,
// Gets you Both Sent and Returned XML

$r=&$c->send($f);
echo "<PRE>";
if(!$r->faultCode())
{
    $v=$r->value();
    $foo = explode("\n" , $v->scalarval());

    $x=0;
    foreach ($foo as $bar){
        $foobar = explode(",", $bar);
        if (preg_match("/created/", $foobar[1])) {
            $nope =1 ; // dont show the first line
        } elseif (sizeof($foobar) < 2) {
            $nope =1 ; // dont lines at the end
        } else {
            $calls[$x] = $foobar; // grab the exploded lines into an array
            $x++;
        }
    }
    print_r($calls);
} else {
    print "An error occurred: ";
    print "Code: " . htmlspecialchars($r->faultCode())
        . " Reason: '" . htmlspecialchars($r->faultString());
}
echo "</PRE>";
?>

```

Perl Example

```
#!/usr/bin
/perl

# by Brian West (bkw_) or #freeswitch
use RPC::XML::
Client;

use Data::
Dumper;

my $client = new RPC::XML::Client('http://localhost:8080
/RPC2');

my $req = RPC::XML::request->new('freeswitch.
api',

'show',

'channels');

$client->credentials ("freeswitch" , "freeswitch",
"works");

$res = $client->send_request
($req);

my $value = $res-
>value;

chomp
($value);

print Dumper
$req;

print Dumper $value;
```

Python Example

```
#!/usr/bin/python

from xmlrpclib import ServerProxy

host = 'localhost'
username = 'freeswitch'
password = 'works'
port = '8080'

server = ServerProxy("http://%s:%s@%s:%s" % (username, password, host, port))
print server.freeswitch.api("show", "channels")
```

Java Example

Using apache XmlRpcClient

```
XmlRpcClientConfigImpl config = new XmlRpcClientConfigImpl();
XmlRpcClient client = new XmlRpcClient();
try {
    config.setServerURL(new URL("http://localhost:8080/RPC2"));
    config.setBasicUserName("freeswitch");
    config.setBasicPassword("works");

    client.setConfig(config);
    client.execute("freeswitch.api", new Object[]{"originate", "sofia/internal/1001 &park()"});

} catch (Exception ex) {
    ex.printStackTrace();
}
```

Drupal Example, Using Drupal 6 API

Create a new node and make sure it can evaluate PHP-code (select the PHP filter under 'input format'). Enter the following in the body of the node:

```
// this code will call the Freeswitch 'help' command and output the (unformatted) result of the command on the
new drupal node/page.
// Drupal expects two arguments, thus the empty argument ("") at the end.

$xmlrpc_call = xmlrpc("http://freeswitch:works@localhost:8080/RPC2","freeswitch.api","help", "")

return ($xmlrpc_call);
```

See Also

- [Documentation/Developer_Documentation](#)
- [mod_xml_rpc](#)
- [Features](#)