

XML-RPC Gateway

Realtime XML-RPC Gateway for API requests

Version 1.2
August 20, 2009

Proprietary Information

This document is our property. It may be used by recipient only for the purpose for which it was transmitted and shall be returned upon request or when no longer needed by recipient. It may not be copied or communicated without the prior written consent of us.

COPYRIGHT NOTIFICATION

Copyright © 2009. All rights reserved.

DISCLAIMER AND LIMITATION OF LIABILITY

We made efforts to ensure the accuracy and completeness of all information in this document. However, we make no warranties of any kind (whether express, implied or statutory) with respect to the information herein. We assume no liability to any party for loss or damage (whether direct or indirect) caused by any errors, omissions or statements of any kind contained in this document. Further, we assume no liability arising from the application or use of the product or service described herein and specifically disclaims any representation that the products or services described herein do not infringe upon any existing or future intellectual property rights. Nothing herein grants the reader any license to make, use, or sell equipment or products constructed in accordance with this document. Finally, all rights and privileges related to any intellectual property right described herein are vested in the patent, trademark, or service mark owner, and no other person may exercise such rights without express permission, authority, or license secured from the patent, trademark, or service mark owner.

We reserve the right to make changes to any information herein without further notice.

NOTICE AND CAUTION concerning Patent or Trademark Rights

The inclusion in this document, the associated on-line file, or the associated software of any information covered by any patent, trademark, or service mark rights shall not constitute nor imply a grant of, or authority to exercise, any right or privilege protected by such patent, trademark, or service mark. All such rights and privileges are vested in the patent, trademark, or service mark owner, and no other person may exercise such rights without express permission, authority, or license secured from the patent, trademark, or service mark owner.

This publication was created using OpenOffice.org by Sun Microsystems Inc.

XML-RPC Gateway

Version 1.2

August 20, 2009

TABLE OF CONTENTS

1. The XML-RPC-Gateway-Service.....	4
Methods.....	5
1.1 Method "Api.echo".....	5
1.2 method "Api.xcall".....	6
2. Client-Access / Examples.....	9
2.1 perl-Client using perl-Module SOAP::Lite/XMLRPC::Lite.....	9
2.2 perl-Client using perl-Module SOAP::Lite/XMLRPC::Lite.....	9
2.3 PHP-Client using PEAR::XML_RPC.....	10
3. ChangeLog.....	11

1. The XML-RPC-Gateway-Service

This Realtime XML-RPC Gateway has been developed to provide an easy way to connect existing shopping cart and domain ordering systems to a user account of our system.

The gateway consists of a XML-RPC server that accepts XML requests conforming to a defined syntax and converts them into API calls. All incoming XML-RPC requests are processed immediately and a XML response is generated and returned to the client.

The XML-RPC-Gateway-Service has the following connection parameters:

- IP: 93.190.235.176
- Port: 8081
- Server-/Proxy-URL: "<http://93.190.235.176:8081>"
- available Methods: "Api.echo" / "Api.xcall"

XML-RPC HTTPS (SSL) call parameters:

- IP: 93.190.235.176
- Port: 8083
- Server-/Proxy-URL: "<http://93.190.235.176:8083>"
- available Methods: "Api.echo" / "Api.xcall"

Methods

1.1 Method "Api.echo"

Description:

To test the XML-RPC gateway service the function "Api.echo" can be used. This function simply echos the text string got from the request.

Arguments:

Any argument is accepted and will be echoed to the client.

Sample Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>Api.echo</methodName>
  <params>
    <param><value><string>Hello World.</string></value></param>
  </params>
</methodCall>
```

Sample Result:

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param><value><string>Hello World.</string></value></param>
  </params>
</methodResponse>
```

1.2 method "Api.xcall"

Description:

To access the API by XML-RPC the function "Api.xcall" can be used.

Arguments:

Both authentication parameters (like "s_login", "s_entity", ...) and API commands can be specified using a simple key/value-hash.

Example content of the key/value-hash:

Parameter	Value
s_login	<API-LOGIN>
s_pw	<API-PASSWORD>
s_entity	<API-ENTITY>
command	QueryDomainList

Sample request (XML):

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>Api.xcall</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>s_entity</name>
            <value>
              <string>API-ENTITY</string>
            </value>
          </member>
          <member>
            <name>s_pw</name>
            <value>
              <string>API-PASSWORD</string>
            </value>
          </member>
          <member>
            <name>command</name>
            <value>
              <string>QueryDomainList</string>
            </value>
          </member>
          <member>
            <name>s_login</name>
            <value>
              <string>API-LOGIN</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Sample response (XML):

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param>
      <value><struct><member><name>PROPERTY</name>
      <value><struct><member><name>FIRST</name>
      <value><array><data><value><int>0</int></value></data></array></value></member>
      <member>
        <name>TOTAL</name>
        <value>
          <array>
            <data><value><int>1</int></value></data>
          </array>
        </value>
      </member>
      <member>
        <name>COUNT</name>
        <value>
          <array>
            <data><value><int>1</int></value></data>
          </array>
        </value>
      </member>
      <member>
        <name>DOMAIN</name>
        <value>
          <array>
            <data><value><string>mydomain.tld</string></value></data>
          </array>
        </value>
      </member>
      <member>
        <name>LAST</name>
        <value>
          <array>
            <data><value><int>0</int></value></data>
          </array>
        </value>
      </member>
      <member>
        <name>LIMIT</name>
        <value>
          <array>
            <data><value><int>1000</int></value></data>
          </array>
        </value>
      </member>
    </struct></value></member>
    <member>
      <name>DESCRIPTION</name>
      <value><string>Command completed successfully</string></value>
    </member>
    <member>
      <name>QUEUETIME</name>
      <value><int>0</int></value>
    </member>
    <member>
      <name>CODE</name>
      <value><int>200</int></value>
    </member>
  </member>
</methodResponse>
```

```
        <name>RUNTIME</name>
        <value><double>0.036</double></value>
</member>
</struct></value>
</param>
</params>
</methodResponse>
```


2. Client-Access / Examples:

2.1 perl-Client using perl-Module SOAP::Lite/XMLRPC::Lite

Client requests a QueryDomainList command to the API and prints a list of the domains

```
#!/usr/bin/perl -w

use Data::Dumper;
use XMLRPC::Lite; # +trace => 'debug';

my $xmlrpc_url = "http://93.190.235.176:8081/RPC2";

my %command;
$command{"s_entity"} = "API-ENTITY";
$command{"s_login"} = "API-LOGIN";
$command{"s_pw"} = "API-PASSWORD";
$command{"command"} = "QueryDomainList";
my @xcommand;
push(@xcommand, \%command);

my $xmlrpc = XMLRPC::Lite
    ->readable(1)
    ->proxy($xmlrpc_url);
my $som = $xmlrpc->call( "Api.xcall", @xcommand );

print Dumper($som->result) . "\n";
```

2.2 perl-Client using perl-Module SOAP::Lite/XMLRPC::Lite

Client requests a QueryDomainList command to the API and prints a list of the domains

```
#!/usr/bin/perl

use Frontier::Client;
use Data::Dumper;

my $server = Frontier::Client->new('url'=>'http://93.190.235.176:8081/RPC2', "encoding"
=> "UTF-8");

my %command;
$command{"s_entity"} = "API-ENTITY";
$command{"s_login"} = "API-LOGIN";
$command{"s_pw"} = "API-PASSWORD";
$command{"command"} = "QueryDomainList";
my @xcommand;
push(@xcommand, \%command);

my $result = $server->call('Api.xcall', @xcommand);

print Dumper($result);
```

2.3 PHP-Client using PEAR::XML_RPC

Client requests a QueryDomainList command to the API and prints a list of the domains

```
<?php
require_once 'XML/RPC.php';

$xml_rpc_host = "93.190.235.176";
$xml_rpc_port = 8081;
$xml_rpc_uri  = "/RPC2";

$command["s_entity"] = "API-ENTITY";
$command["s_login"]  = "API-LOGIN";
$command["s_pw"]     = "API-PASSWORD";
$command["command"]  = "QueryDomainList";
$params = array(XML_RPC_encode($command));

$msg = new XML_RPC_Message('Api.xcall', $params);
$client = new XML_RPC_Client($xml_rpc_uri, $xml_rpc_host, $xml_rpc_port);
#$client->setDebug(1);
$response = $client->send($msg);
$v = XML_RPC_decode($response->value());
echo "<pre>"; print_r($v); echo "</pre>";

if ($response->faultCode()) {
    // Error-Handling
}

// ...
?>
```

3. ChangeLog

Date	Old / new version	Description of changes
2007-12-24	1.0 / 1.1	Added description for connection to gateway using SSL encryption
2009-08-20	1.1 / 1.2	Replaced old IP addresses by new IP addresses