How to make a PHP SOAP call using the SoapClient class

I'm used to writing PHP code, but do not often use Object-Oriented coding. I now need to interact with SOAP (as a client) and am not able to get the syntax right. I've got a WSDL file which allows me to properly set up a new connection using the SoapClient class. However, I'm unable to actually make the right call and get data returned. I need to send the following (simplified) data:

- Contact ID
- Contact Name
- General Description
- Amount

There are two functions defined in the WSDL document, but I only need one ("FirstFunction" below). Here is the script I run to get information on the available functions and types:

```
$client = new SoapClient("http://example.com/webservices?wsdl");
var_dump($client->__getFunctions());
var_dump($client->__getTypes());
```

And here is the output it generates:

```
array(
  [0] => "FirstFunction Function1(FirstFunction $parameters)",
  [1] => "SecondFunction Function2(SecondFunction $parameters)",
);

array(
  [0] => struct Contact {
    id id;
    name name;
  }
  [1] => string "string description"
  [2] => string "int amount"
}
```

Say I want to make a call to the FirstFunction with the data:

- Contact ID: 100
- Contact Name: John
- General Description: Barrel of Oil
- Amount: 500

What would be the right syntax? I've been trying all sorts of options but it appears the soap structure is quite flexible so there are very many ways of doing this. Couldn't figure it out from the manual either...

UPDATE 1: tried sample from MMK:

```
$client = new SoapClient("http://example.com/webservices?wsdl");

$params = array(
  "id" => 100,
  "name" => "John",
  "description" => "Barrel of Oil",
  "amount" => 500,
);
$response = $client->__soapCall("Function1", array($params));
```

But I get this response: `Object has no 'Contact' property`. As you can see in the output of `getTypes()`, there is a `struct` called `Contact`, so I guess I somehow need to make clear my parameters include the Contact data, but the question is: how?

UPDATE 2: I've also tried these structures, same error.

```
$params = array(
  array(
    "id" => 100,
    "name" => "John",
  ),
  "Barrel of Oil",
  500,
);
```

As well as:

```
$params = array(
  "Contact" => array(
    "id" => 100,
    "name" => "John",
  ),
  "description" => "Barrel of Oil",
  "amount" => 500,
);
```

Error in both cases: Object has no 'Contact' property`

php soap

edited Jul 23 '12 at 20:13

asked Jul 21 '12 at 15:52
user1305445

8 Answers

This is what you need to do.

Just to know, I tried to recreate your situation...

- For this example, I made a .NET sample webservice with a `WebMethod` called `Function1` and these are the parameters:

> Function1(Contact Contact, string description, int amount)

- In which `Contact` is just a bean `class` that has getters and setters for `id` and `name` like in your case.
- You can download this .NET sample webservice at:

https://www.dropbox.com/s/6pz1w94a52o5xah/11593623.zip

The code.

This is what you need to do at **PHP** side:

(Tested and working)

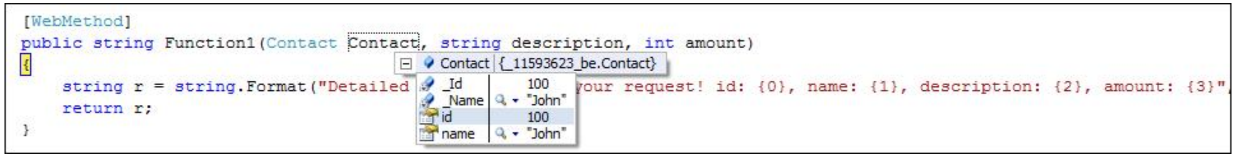
```php
<?php
/* Create a class for your webservice structure, in this case: Contact */
class Contact {
    function Contact($id, $name)
    {
        $this->id = $id;
        $this->name = $name;
    }
}

/* Initialize webservice with your WSDL */
$client = new SoapClient("http://localhost:10139/Service1.asmx?wsdl");

/* Fill your Contact Object */
$contact = new Contact(100, "John");

/* Set your parameters for the request */
$params = array(
  "Contact" => $contact,
  "description" => "Barrel of Oil",
  "amount" => 500,
);

/* Invoke webservice method with your parameters, in this case: Function1 */
$response = $client->__soapCall("Function1", array($params));

/* Print webservice response */
var_dump($response);

?>
```

How I know this is working?

- If you do `print_r($params);` you will see this output, as your webservice expects:

> Array ([Contact] => Contact Object ([id] => 100 [name] => John) [description] => Barrel of Oil [amount] => 500)

- When I debugged the sample .NET webservice I got this:



```
[WebMethod]
public string Function1(Contact Contact, string description, int amount)
{
    string r = string.Format("Detailed        your request! id: {0}, name: {1}, description: {2}, amount: {3}",
    return r;
}
```

(As you can see, `Contact` object is not null and also other parameters, that means your request was successfully done from PHP side).

- The response from .NET sample webservice was the expected one and shown at PHP side:

> object(stdClass)[3] public 'Function1Result' => string 'Detailed information of your request! id: 100, name: John, description: Barrel of Oil, amount: 500' (length=98)

Hope this helps :-)

edited Jul 30 '12 at 15:52 answered Jul 30 '12 at 1:37

Oscar Jara

9,277 7 38 80

1 Perfect! I acted like I knew a little bit more about SOAP services than I really did and this got me where I needed to be. – chapman84 Mar 27 '13 at 13:52

4 @chapman84 you never accepted his answer. – taco Aug 26 '13 at 15:28

1 I didn't ask the question, otherwise I would have. The question and this answer got an upvote from me though. – chapman84 Aug 26 '13 at 16:14

3 @user should accept it :) BTW, very nice answer, complete and very clear. +1 – Yann39 Aug 29 '13 at 13:37

You can use SOAP services this way too:

```
<?php
//Create the client object
$soapclient = new SoapClient('http://www.webservicex.net/globalweather.asmx?WSDL');

//Use the functions of the client, the params of the function are in
//the associative array
$params = array('CountryName' => 'Spain', 'CityName' => 'Alicante');
$response = $soapclient->getWeather($params);

var_dump($response);
```

This is an example with a real service, and it works.

Hope this helps.

answered Jun 12 '13 at 14:00

Salvador P.

914 9 11

First initialize webservices:

```
$client = new SoapClient("http://example.com/webservices?wsdl");
```

Then set and pass the parameters:

```
$params = array (
    "arg0" => $contactid,
    "arg1" => $desc,
    "arg2" => $contactname
);

$response = $client->__soapCall('methodname', array($params));
```

Note that the method name is available in WSDL as operation name, e.g.:

```
<operation name="methodname">
```

edited Jun 21 '16 at 15:05
dotancohen
11k 15 75 126

answered Jul 21 '12 at 16:18
MMK
849 4 19

Thanks! I've tried this but I get the error "Object has no 'Contact' property". Will update my question with full details. Any ideas? – user1305445 Jul 21 '12 at 17:29

@user16441 Can you post the WSDL and schema of the service? I usually start by figuring out what XML the service expects, then using WireShark to figure out what my client is actually sending. – davidfmatheson Jul 30 '12 at 13:51

I don't know why my web service has the same structure with you but it doesn't need Class for parameter, just is array.

For example: - My WSDL:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:ns="http://www.kiala.com/schemas/psws/1.0">
    <soapenv:Header/>
    <soapenv:Body>
        <ns:createOrder reference="260778">
            <identification>
                <sender>5390a7006cee11e0ae3e0800200c9a66</sender>
                <hash>831f8c1ad25e1dc89cf2d8f23d2af...fa85155f5c67627</hash>
                <originator>VITS-STAELENS</originator>
            </identification>
            <delivery>
```

```
                <from country="ES" node=””/>
                <to country="ES" node="0299"/>
            </delivery>
            <parcel>
                <description>Zoethout thee</description>
                <weight>0.100</weight>
                <orderNumber>10K24</orderNumber>
                <orderDate>2012-12-31</orderDate>
            </parcel>
            <receiver>
                <firstName>Gladys</firstName>
                <surname>Roldan de Moras</surname>
                <address>
                    <line1>Calle General Oraá 26</line1>
                    <line2>(4º izda)</line2>
                    <postalCode>28006</postalCode>
                    <city>Madrid</city>
                    <country>ES</country>
                </address>
                <email>gverbruggen@kiala.com</email>
                <language>es</language>
            </receiver>
        </ns:createOrder>
    </soapenv:Body>
</soapenv:Envelope>
```

I var_dump:

```
var_dump($client->getFunctions());
var_dump($client->getTypes());
```

Here is result:

```
array
  0 => string 'OrderConfirmation createOrder(OrderRequest $createOrder)'
(length=56)

array
  0 => string 'struct OrderRequest {
 Identification identification;
 Delivery delivery;
 Parcel parcel;
 Receiver receiver;
 string reference;
}' (length=130)
  1 => string 'struct Identification {
 string sender;
 string hash;
 string originator;
}' (length=75)
  2 => string 'struct Delivery {
 Node from;
 Node to;
}' (length=41)
  3 => string 'struct Node {
 string country;
 string node;
}' (length=46)
  4 => string 'struct Parcel {
 string description;
 decimal weight;
 string orderNumber;
 date orderDate;
}' (length=93)
  5 => string 'struct Receiver {
 string firstName;
 string surname;
 Address address;
 string email;
 string language;
}' (length=106)
  6 => string 'struct Address {
 string line1;
 string line2;
 string postalCode;
 string city;
 string country;
}' (length=99)
  7 => string 'struct OrderConfirmation {
 string trackingNumber;
 string reference;
}' (length=71)
  8 => string 'struct OrderServiceException {
 string code;
 OrderServiceException faultInfo;
 string message;
}' (length=97)
```

So in my code:

```
$client  = new SoapClient('http://packandship-ws.kiala.com/psws/order?wsdl');

$params = array(
    'reference' => $orderId,
    'identification' => array(
        'sender' => param('kiala', 'sender_id'),
        'hash' => hash('sha512', $orderId . param('kiala', 'sender_id') . param('kiala', 'password')),
        'originator' => null,
    ),
    'delivery' => array(
        'from' => array(
            'country' => 'es',
            'node' => '',
        ),
        'to' => array(
            'country' => 'es',
            'node' => '0299'
        ),
    ),
    'parcel' => array(
        'description' => 'Description',
        'weight' => 0.200,
        'orderNumber' => $orderId,
        'orderDate' => date('Y-m-d')
    ),
    'receiver' => array(
        'firstName' => 'Customer First Name',
        'surname' => 'Customer Sur Name',
        'address' => array(
            'line1' => 'Line 1 Adress',
            'line2' => 'Line 2 Adress',
            'postalCode' => 28006,
            'city' => 'Madrid',
            'country' => 'es',
            ),
        'email' => 'test.ceres@yahoo.com',
        'language' => 'es'
    )
);
$result = $client->createOrder($params);
var_dump($result);
```

but it successfully!

answered Jul 17 '13 at 3:17

Tín Phạm
341 3 13

Your example is more helpfull, cause it shows structure dependecies – vladkras Jun 10 '16 at 10:07

read this;-

http://php.net/manual/en/soapclient.call.php

Or

This is a good example, for the SOAP function "__call". However it is deprecated.

```
<?php
    $wsdl = "http://webservices.tekever.eu/ctt/?wsdl";
    $int_zona = 5;
    $int_peso = 1001;
    $cliente = new SoapClient($wsdl);
    print "<p>Envio Internacional: ";
    $vem = $cliente->__call('CustoEMSInternacional',array($int_zona, $int_peso));
    print $vem;
    print "</p>";
?>
```

answered Jul 29 '12 at 19:52

Abid Hussain
6,051 1 17 37

If you create the object of SoapParam, This will resolve your problem. Create a class and map it with objecttype given by WebService, Initialize the values and send in request. See the sample below.

```
struct Contact {

    function Contact ($pid, $pname)
    {
      id = pid;
      name = pname;
  }
}

$struct = new Contact(100,"John");

$soapstruct = new SoapVar($struct, SOAP_ENC_OBJECT,
"Contact","http://soapinterop.org/xsd");
```

```
$ContactParam = new SoapParam($soapstruct, "Contact")

response = $client->Function1($ContactParam);
```

edited Apr 29 '14 at 9:52
ManseUK
31.6k 7 56 85

answered Jul 25 '12 at 14:19
Umesh Chavan
550 2 11

You need declare class **Contract**

```
class Contract {
  public $id;
  public $name;
}

$contract = new Contract();
$contract->id = 100;
$contract->name = "John";

$params = array(
  "Contact" => $contract,
  "description" => "Barrel of Oil",
  "amount" => 500,
);
```

or

```
$params = array(
  $contract,
  "description" => "Barrel of Oil",
  "amount" => 500,
);
```

Then

```
$response = $client->__soapCall("Function1", array("FirstFunction" => $params));
```

or

```
$response = $client->__soapCall("Function1", $params);
```

answered Jul 29 '12 at 23:17

Ramil Amr
1,099 7 22

You need a multi-dimensional array, you can try the following:

```
$params = array(
   array(
      "id" => 100,
      "name" => "John",
   ),
   "Barrel of Oil",
   500
);
```

in PHP an array is a structure and is very flexible. Normally with soap calls I use an XML wrapper so unsure if it will work.

EDIT:

What you may want to try is creating a json query to send or using that to create a xml buy sort of following what is on this page: http://onwebdev.blogspot.com/2011/08/php-converting-rss-to-json.html

edited Aug 8 '12 at 19:29

answered Jul 23 '12 at 19:29

James Williams
3,733 1 11 29

thanks, but it did not work either. How do you use XML wrappers exactly, perhaps that is easier to use than this... – user1305445 Jul 23 '12 at 20:13

First you have to make sure that your WSDL can handle XML wrappers. But it is similar, you build the request in XML and in most cases use curl. I have use SOAP with XML to process transactions through banks. You can check out these as a starter point. forums.digitalpoint.com/showthread.php?t=424619#post4004636 w3schools.com/soap/soap_intro.asp – James Williams Jul 23 '12 at 20:44

and this one w3schools.com/soap/soap_intro.asp – James Williams Jul 23 '12 at 20:44