

## Problem with set\_participant\_properties() with XML-RPC

(Limesurvey version 6.15.24)

When sending an XML request to the method set\_participant\_properties() of the remotecontrol\_handle.php file, we receive the following error "Calling parameters do not match signature".

### #### Steps to reproduce

Create an XML-RPC client and send the following request to the server. This request contains 4 params, the fourth one being an associative array (struct in XML-RPC).

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>set_participant_properties</methodName>
  <params>
    <param>
      <value>
        <string>ZTF~bMZs6aE~BuQW115MVCpn0~gNFccC</string>
      </value>
    </param>
    <param>
      <value>
        <int>366661</int>
      </value>
    </param>
    <param>
      <value>
        <int>11</int>
      </value>
    </param>
    <param>
      <value>
        <struct>
          <member>
            <name>firstname</name>
            <value>
              <string>John</string>
            </value>
          </member>
          <member>
            <name>lastname</name>
            <value>
              <string>Hardaker</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

```

        </member>
        <member>
            <name>email</name>
            <value>
                <string>test@gmail.es</string>
            </value>
        </member>
    </struct>
</value>
</param>
</params>
</methodCall>

```

#### #### Expected result

The previous should work as the method expects an associative array as forth parameters. Limesurvey tests are programmed with this in mind and the pass. It also works with JSON-RPC.

#### #### Actual result

We receive the following XML with the error: "Calling parameters do not match signature"

```

<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
    <fault>
        <value>
            <struct>
                <member>
                    <name>faultCode</name>
                    <value>
                        <int>623</int>
                    </value>
                </member>
                <member>
                    <name>faultString</name>
                    <value>
                        <string>Calling parameters do not match signature</string>
                    </value>
                </member>
            </struct>
        </value>
    </fault>
</methodResponse>

```

## XML-RPC Client

To quickly create an XML-RPC client use the **phpxmlrpc** project from <https://github.com/gggeek/phpxmlrpc>

Install it with:

```
$ composer require phpxmlrpc/phpxmlrpc:^4.9
```

And then use the following php code to send the request to the server (enable XML-RPC in the interface tab of the global settings in Limesurvey).

```
<?php

require_once __DIR__ . '/vendor/autoload.php';

use PhpXmlRpc\Client;
use PhpXmlRpc\Encoder;
use PhpXmlRpc\Request;

$userPassword = array("<USERNAME>", "<PASSWORD>");

// Create xml-rpc parameters from the input array
$encoder = new Encoder();
$v = $encoder->encode($userPassword);

// create client and request objects
$req = new Request('get_session_key', $v);
$client = new Client("http://YOUR_SERVER/index.php/admin/remotecontrol");

// set maximum debug level, to have the complete communication printed to screen
$client->setDebug(2);

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

// check response for errors, and take appropriate action
if (!$res->faultCode()) {
    $value = $res->value();
    $sessionKey = $encoder->decode($value);
}

$params = array(
```

```

    $sessionKey,
    <A VALID SURVEY ID>,
    <AN EXISTING PARTICIPANT ID>,
    array('firstname' => 'John', 'lastname' => 'Hardaker', 'email' =>
'test@gmail.com',),
);
$v = $encoder->encode($params);
$req = new Request('set_participant_properties', $v);
$res = $client->send($req);

```

Running this against a recent Limesurvey version, you get the above XML as the response:

```

.....
    <name>faultString</name>
    <value>
        <string>Calling parameters do not match signature</string>
    </value>
.....

```

## The problem

The problem is the mismatch between what the method accepts as the fourth param (an associative array) and what the XML-RPC server infers from the method param list (an array).

The method is defined as follows:

```

/**
 * Set properties of a survey participant (RPC function)
 *
 * Allow to set properties about a specific participant, only one
participant can be updated.
 * @see \Token for available properties
 *
 * @access public
 * @param string $sSessionKey Auth credentials
 * @param int $iSurveyID Id of the Survey that participants belong
 * @param array|int $aTokenQueryProperties of participant properties used
to query the participant, or the token id as an integer
 * @param array $aTokenData Data to change
 * @return array Result of the change action
 */
public function set_participant_properties($sSessionKey, $iSurveyID,
$aTokenQueryProperties, $aTokenData)

```

In the param list, the fourth param is stated as an array, though an associative array.

The problem is that the XML-RPC server uses reflection and inspection to know what parameter the method accepts. And it infers that is a normal array, instead of an associative array.

To hint the XML-RPC to consider it as an associative array, it should be stated that it is a "struct".

Limesurvey uses the Zend XML-RPC server.

In its documentation, in the Conventions section (see this link <https://github.com/zendframework/zend-xmlrpc/blob/bfd5e938d9aa898b0346ebcd51eedefca31e040c/docs/book/server.md#conventions>) we see the explanation and an example.

*XML-RPC types do not necessarily map one-to-one to PHP types. However, the code will do its best to guess the appropriate type based on the values listed in @param and @return annotations. Some XML-RPC types have no immediate PHP equivalent, however, and should be hinted using the XML-RPC type in the PHPDoc. These include:*

- `dateTime.iso8601`, a string formatted as 'YYYYMMDDTHH:mm:ss'
- `base64`, base64 encoded data
- `struct`, **any associative array**

*An example of how to hint follows:*

```
/**
 * This is a sample function
 *
 * @param base64 $val1 Base64-encoded data
 * @param dateTime.iso8601 $val2 An ISO date
 * @param struct $val3 An associative array
 * @return struct
 */
function myFunc($val1, $val2, $val3)
{}
```

## The solution

As seen above, the solution is as easy as replacing the fourth parameter of the method with the bold line

```
/**
 * Set properties of a survey participant (RPC function)
```

```
*
* Allow to set properties about a specific participant, only one
participant can be updated.
* @see \Token for available properties
*
* @access public
* @param string $sSessionKey Auth credentials
* @param int $iSurveyID Id of the Survey that participants belong
* @param array|int $aTokenQueryProperties of participant properties used
to query the participant, or the token id as an integer
* @param struct $aTokenData Associative array of data to change
* @return array Result of the change action
*/
public function set_participant_properties($sSessionKey, $iSurveyID,
$aTokenQueryProperties, $aTokenData)
```

If you agree, I can send a PR to solve the issue.

Since it is a small documentation change it should not impact any test nor should imply any regression problem.