

Perl: How to work with methods with multiple outbound parameters



Joe Pruitt, 2004-15-11

I've seen this question asked in several forms so I thought I'd try to shed some light on how to handle accessing methods with multiple outbound parameters using Perl's [SOAP::Lite](#): Let's take a look at these function prototypes:

1. void func1(out string param1);
2. void func2(out string param1, out string [] param2);
3. string func3(out string param1, out string [] param2);
4. string[] func4(out string param1, out string [] param2);
5. string func5(out string [] param1, out string [][] param2);

I know there are more combinations than this but this should cover most of the bases (other object types can take the place of string in these examples). I'll take each of these point by point and note how to access the returned parameters. Keep in mind that SOAP::Lite doesn't distinguish a returned value vs an outbound parameter, it uses the **FIRST** returned value whether it's the return value or the first outbound parameter. Then it stuffs all subsequent parameters into it's paramsout array.

1. void func1(out string param1);

Here we have no return value but a single outbound parameter. Since the outbound parameter is the first returned value, it will be stored in the **result**.

```
$soapResponse = $soap->func1();
$paramsout = $soapResponse->paramsout;
$result = $soapResponse->result; # Scalar value
print "param1 => $result\n";
```

2. void func2(out string param1, out string [] param2)

This method has no return value but two outbound parameters so **param1** will be stored in the **result** and param2 will be the first element of the **paramsout** array.

```
$soapResponse = $soap->func2();
$paramsout = $soapResponse->paramsout;
$result = $soapResponse->result; # Scalar value
@param2 = @{$paramsout[0]}; # Array
print "param1 => $result\n";
print "param2 => {";
foreach $val (@param2)
{
    print "$val, ";
}
print "}\n";
```

3. string func3(out string param1, out string [] param2);

This method returns a string as a return value and has 2 outbound parameters. So, the returned value will be in the **result** and params 1 and 2 will be stored in the **paramsout** array.

```
$soapResponse = $soap->func3();
$params = $soapResponse->paramsout;
$retval = $soapResponse->result;
$params1 = @params[0]; # Scalar value
@param2 = @{$params[1]}; # Array
print "retval => $retval\n";
print "param1 => $params1\n";
print "param2 => {";
foreach $val (@param2)
{
    print "$val, ";
}
print "}\n";
```

4. string[] func4(out string param1, out string [] param2);

This method is similar to number 2, but it returns an array which will be stored in the **result** and, again, params 1 and 2 will be stored in the **paramsout** array.

```
$soapResponse = $soap->func4();
$params = $soapResponse->paramsout;
@retval = @{$soapResponse->result};
$params1 = @params[0]; # Scalar value
@param2 = @{$params[1]}; # Array
print "retval => {";
foreach $val (@retval)
{
    print "$val, ";
}
print "}\n";
print "param1 => $params1\n";
print "param2 => {";
foreach $val (@param2)
{
    print "$val, ";
}
print "}\n";
```

5. string func5(out string[] param1, out string [][] param2);

Here we've introduced an Array of Arrays as an outbound parameter. the Return value will again be returned in the **result** with the parameters stored in the **paramsout** array.

```
$soapResponse = $soap->func5();
$params = $soapResponse->paramsout;
$retval = @{$soapResponse->result};
@param1 = {@params[0]}; # Array
@param2 = {@params[1]}; # Array of Array
print "retval => $retval\n";
print "param1 => {";
foreach $val (@param1)
{
    print "$val, ";
}
print "}\n";
print "param2 => {\n";
for $i ( 0 .. $#param2 )
{
    print " {";
    for $j ( 0 .. ${param2[$i]} )
    {
        print "$param2[$i][$j], ";
    }
    print "}\n";
}
print "}\n";
```

You could have used the **foreach** statement for the array of array but this just shows another way to access the data by index.

-Joe

References

[SOAP::Lite](#)

[SOAP::Lite Perldocs](#)

[Manipulating Arrays of Arrays in perl](#)

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com

©2016 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at [f5.com](#). Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113