

**MASHERY**

# **SOAP Tips, Tricks & Tools**

**March 12, 2008**

**Rob Richards**

**<http://xri.net/=rob.richards>**

**[www.cdatazone.org](http://www.cdatazone.org)**

# Helpful Tools

- soapUI
  - <http://www.soapui.org/>
  - Multiple platforms / Free & enhanced Pro versions
- SOAPSonar
  - <http://www.crosschecknet.com/>
  - Windows only / Free and Professional versions
- Eclipse
  - WSDL Editor
  - XML / XSD Editors & Validators
- Misc. XML Editors
  - XML NotePad 2007 (Microsoft)
  - XML Spy (Altova)

# soapUI Features

- Web Service Inspection
  - WSDL Inspector
  - HTTP data inspection (request & response)
- Web Service Invocation
  - Automatic Request generation
  - Authentication support (Basic, Digest, WS–Security)
  - Custom HTTP Header support
- Web Service Development and Validation
- Web Service Functional Testing
- Web Service Load Testing
- Web Service Simulation

Projects

- Amazon
  - AWSECommerceServicePortType
    - BrowseNodeLookup
    - CartAdd
    - CartClear
    - CartCreate
    - CartGet
    - CartModify
    - CustomerContentLookup
    - CustomerContentSearch
    - Help
    - ItemLookup
    - ItemSearch
      - Request 1
    - ListLookup
    - ListSearch
    - MultiOperation
    - SellerListingLookup
    - SellerListingSearch
    - SellerLookup
    - SimilarityLookup
    - TransactionLookup
  - Amazon Tests
  - MockService 1
- Personal
- PropertyExpansionTip

```

ItemSearch - Request 1
http://soap.amazon.com/onca/soap?Service=AWSECommerceService

XML
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:ItemSearch>
      <ns:SubscriptionId>1853898630001</ns:SubscriptionId>
      <!--Optional:-->
      <ns:AssociateTag/>
      <!--Optional:-->
      <ns:XMLEscaping/>
      <!--Optional:-->
      <ns:Validate/>
      <!--Optional:-->
      <!--Zero or more repetitions:-->
      <ns:Request>
        <ns:Author>Banks</ns:Author>
        <ns:SearchIndex>Books</ns:SearchIndex>
      </ns:Request>
    </ns:ItemSearch>
  </soapenv:Body>
</soapenv:Envelope>
  
```

```

SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
SOAP-ENV:Body
  <ItemSearchResponse xmlns="http://webservices.amazon.com/onca/xml" >
    <OperationRequest>
      <HTTPHeaders>
        <Header Name="UserAgent" Value="Jakarta" />
      </HTTPHeaders>
      <RequestId>0GMDCCZYJHAMK4DGT7CY1</RequestId>
      <Arguments>
        <Argument Name="Service" Value="AWSECommerceService" />
      </Arguments>
      <RequestProcessingTime>0.119747161865234</RequestProcessingTime>
    </OperationRequest>
    <Items>
      <Request>
        <IsValid>True</IsValid>
        <ItemSearchRequest>
          <Author>Banks</Author>
          <SearchIndex>Books</SearchIndex>
        </ItemSearchRequest>
      </Request>
      <TotalResults>6378</TotalResults>
    </Items>
  </ItemSearchResponse>
  
```

XSD (string) Table (1) XML Doc

SOAP Request Request Attachments (0) HTTP Headers (0)

response time: 2140ms (8500 bytes)

XSD Table XML Doc

SOAP Response Response Attachments (0) HTTP Headers (8) S...

1 : 1

Request Properties

Property	Value
Name	Request 1
Encoding	UTF-8
Endpoint	http://soap.amazo...
Enable MTOM/Inline	false
Username	
Password	
Domain	

Log messages:

```

Sun Mar 25 00:10:27 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 286ms (539 bytes)
Sun Mar 25 00:10:40 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 291ms (3042 bytes)
Sun Mar 25 00:12:37 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 251ms (1901 bytes)
Sun Mar 25 00:13:09 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 550ms (8500 bytes)
Sun Mar 25 00:20:30 CET 2007:INFO:Got response for [AWSECommerceServicePortType.ItemSearch:Request 1] in 2140ms (8500 bytes)
  
```

Log filters: soapUI log | http log | jetty log

Show checkboxes: Show Show Show



Data - F:\source\cvs\php5\_2\Debug\_TS\IAnimalSearchWSv2.wsdl - Eclipse Platform

File Edit Source Navigate Search Project Run XML Window Help

testproj

- .project
- IAnimalSearchWSv2.wsdl
- IEventUploadWSv1.wsdl
- import.xsd

IAnimalSearchWSv2.wsdl

```

?xml version="1.0" encoding="UTF-8"
definitions
  name="AnimalSearchWSv2"
  targetNamespace="http://com/darkston/cts/webservice/animal/search/AnimalSearchWSv2.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://com/darkston/cts/webservice/animal/search/AnimalSearchWSv2.wsdl"
  xmlns:ns1="http://com.darkston.cts.webservice.animal.search/IAnimalSearchWSv2.xsd"
  types
  message
    name="executeSearch0Request"
    part
  message
    name="executeSearch0Response"
    part
      name
      type="ns1:com_darkston_cts_webservice_animal_search_value_AnimalSearchResultWSValue"
  portType
  binding
  service
    name="AnimalSearchWSv2"
    documentation="Provides the Service Endpoint (busines logic) for the AnimalSearchWSv1 web service."
    port
      name="AnimalSearchWSv2Port"
      binding="tns:AnimalSearchWSv2Binding"
      soap:address
  
```

Design Source

Properties

Property	Value
Attributes	
binding	tns:AnimalSearchWSv2Binding
name	AnimalSearchWSv2Port

definitions/service/port

Writable Smart Insert 1:1

Data - testproj/import.xsd - Eclipse Platform

File Edit Source Navigate Search Project Run XSD Window Help

100%

Navigator

- testproj
  - .project
  - IAnimalSearchWSv2.wsdl
  - IEventUploadWSv1.wsdl
  - import.xsd

IAnimalSearchWSv2.wsdl \*IAnimalSearchWSv2.wsdl import.xsd

Schema : (no target namespace specified)

Directives

Elements

- myemptyelement
- tl : anyType

Types

- meetingNote
  - AtoZ : xsd:string
  - Credits : AtoZ | oneToTen
  - CreditType : xsd:decimal
  - EmailType : xsd:string
  - oneToTen : xsd:integer
  - TitleType : xsd:string

Attributes

Groups

- togroup

Design Source

Properties Tasks Problems Error Log Bookmarks

simpleType

General

Name: Credits

Constraints

Variety: union

Documentation

Member types: AtoZ oneToTen

Data - testproj/IAnimalSearchWSv2.wsdl - Eclipse Platform

File Edit Source Navigate Search Project Run WSDL Editor Window Help

testproj

- .project
- IAnimalSearchWSv2.wsdl
- IEventUploadWSv1.wsdl
- import.xsd

IAnimalSearchWSv2

AnimalSearchWSv2Port  
https://www.dia.livestock...

AnimalSearchWSv2PortType

executeSearch			
input	criteria	com_clarkston_cts_webservice_animal_search_value_AnimalSearchCriteriaWSValue	
output	return	com_clarkston_cts_webservice_animal_search_value_AnimalSearchResultWSValue	
NewOperation			
input	NewOperationRequest	string	
output	NewOperationResponse	string	

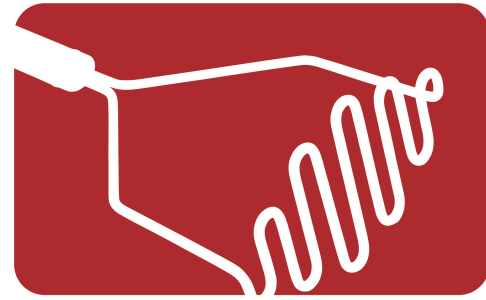
part

**General** Name: NewOperationRequest

**Documentation** Type: string

**Extensions** Reference Kind:  Type  Element

definitions/message/part | Writable | Smart Insert | 92 : 4



**MASHERY**

# Making SOAP Requests

**Tips and Tricks working with the SoapClient**



# Canadian Cattle Identification Agency (CCIA)

- Handles cattle age verification
- Provides a centralized database where producers can maintain cattle information
- Information is readily available to help speed up import/exports
- Information can be correlated with RFID tags
- System is entirely online
- SOAP is used for application integration
- Website: <http://www.canadaid.ca/>

# Canadian Cattle Identification Agency (CCIA)

Access to the CCIA services requires authentication through the use of WS-Security in the means of a UsernameToken.

To simplify the examples, authentication has been omitted.

# CCIA: Retrieve API Information

```
$wsdl = 'http://www.clia.livestockid.ca/CLTS/public/help/en/  
webservices/WebServiceAnimalSearch/IAAnimalSearchWSv2.wsdl';
```

```
$client = new SOAPClient($wsdl);
```

```
$functions = $client->__getFunctions();  
    foreach ($functions AS $function) {  
        print $function."\n\n";  
    }  
}
```

```
$types = $client->__getTypes();  
    foreach ($types AS $type) {  
        print $type."\n\n";  
    }  
}
```

# CCIA: Can We Be Any More Descriptive?

```
com_clarkston_cts_webservice_animal_search_value_AnimalSearchResultWSValue  
executeSearch (com_clarkston_cts_webservice_animal_search_value_  
AnimalSearchCriteriaWSValue $criteria)
```

```
struct  
    com_clarkston_cts_webservice_animal_search_value_AnimalSearchCriteriaWSValue {  
    ArrayOfcom_clarkston_cts_webservice_value_TagRangeWSValue tagRangeList;  
    string publicAccountId;  
    }
```

```
struct com_clarkston_cts_webservice_value_TagRangeWSValue {  
    string tagType;  
    string endTag;  
    string startTag;  
    }
```

```
com_clarkston_cts_webservice_value_TagRangeWSValue  
    ArrayOfcom_clarkston_cts_webservice_value_TagRangeWSValue[]
```

# CCIA: Functions and Types Translated

AnimalResult executeSearch(AnimalCriteria \$criteria)

```
struct TagRange {  
    string tagType;  
    string endTag;  
    string startTag;  
}
```

TagRange ArrayOfTagRange[]

```
struct AnimalCriteria {  
    ArrayOfTagRange    tagRangeList;  
    string              publicAccountId;  
}
```

# CCIA: Working with Structs

## From WSDL

```
struct TagRange {  
    string tagType;  
    string endTag;  
    string startTag;  
}
```

## In PHP

```
array("tagType" => x, "endTag" => y, "startTag" => z);
```

```
class TagRange {  
    public $tagType;  
    public $endTag;  
    public $startTag;  
}
```

# CCIA: Making The Request

```
try {
    $client = new SoapClient($wsdl);
    $stag = '000000124000050000102';
    $etag = '000000124000050000106';

    $tag1 = array('startTag'=>$stag, 'endTag'=>$etag,
                 'tagType'=>'C');
    $tRangeList = array($tag1);

    $res = $client->executeSearch(array('tagRangeList'=>$tRangeList,
                                       'publicAccountId'=>""));

    var_dump($res);
} catch (SoapFault $e) {
    var_dump($e);
}
```

# CCIA: DOH! . . . Denied





# CCIA: DOH! . . . Denied

```
object(SoapFault)#3 (9) {  
  ...  
  ["faultstring"]=> string(273) "No Deserializer  
found to deserialize a 'http://  
com.clarkston.cts.webservice.animal.search/  
IAntimalSearchWSv2.xsd:ArrayOfcom_clarkston_cts_w  
ebservice_value_TagRangeWSValue' using  
encoding style 'http://schemas.xmlsoap.org/soap/  
encoding/'. [java.lang.IllegalArgumentException]"  
  
  ["faultcode"]=> string(15) "SOAP-ENV:Client"  
  ["faultactor"]=> string(22) "/CLTS/  
AnimalSearchWSv2"  
}
```

# CCIA: Capturing the Request

```
$client_options = array ('trace'=>1);  
  
$client = new SoapClient($wsdl, $client_options);  
try {  
    /* code edited for brevity */  
  
    $res = $client->executeSearch( array(  
                                    'tagRangeList'=>$tRangeList,  
                                    'publicAccountId'=>"" ) );  
} catch (SoapFault $e) {  
    print $client->__getLastRequest();  
}
```

# CCIA: Capturing Messages

Be careful when capturing the output from `__getLastRequest()` or `__getLastResponse()`

- Pipe the output to a file  
`php myclient.php > request.php`
- Capture it to a variable  
`$request = $client->__getLastRequest();`  
`file_put_contents('request.xml', $request);`

# CCIA: The SOAP Request (edited)

```
<ns1:executeSearch>
  <criteria xsi:type="ns2:AnimalSearchCriteriaWSValue">
    <tagRangeList SOAP-ENC:arrayType="ns2:TagRange[1]"
      xsi:type="ns2:ArrayOfTagRange">
      <item xsi:type="ns2:TagRange">
        <tagType xsi:type="xsd:string">C</tagType>
        <endTag xsi:type="xsd:string">...00106</endTag>
        <startTag xsi:type="xsd:string">...00102</startTag>
      </item>
    </tagRangeList>
    <publicAccountId xsi:type="xsd:string"/>
  </criteria>
</ns1:executeSearch>
```

# CCIA: So Where's the Problem?

Time to pull out one or more of  
your SOAP tools

Fire up your favorite search  
engine



No Deserializer found to deserialize

Search

[Advanced Search](#)  
[Preferences](#)

Web

Results 1 - 10 of about 89,500 for **No Deserializer found to deserialize**. (0.04 seconds)

### [Re: No Deserializer found to deserialize ArrayOf\\_xsd\\_string ...](#)

org> Sent: Tuesday, July 15, 2003 3:13 AM Subject: RE: **No Deserializer found to deserialize** ArrayOf\_xsd\_string Thanks for your answer. ...

[aspn.activestate.com/ASP/Message/Apache-Soap-Dev/1738165](http://aspn.activestate.com/ASP/Message/Apache-Soap-Dev/1738165) - 31k -

[Cached](#) - [Similar pages](#)

### [\[PMX:#\] RE: Help, please! No Deserializer found to deserialize ...](#)

**No Deserializer found to deserialize** I got the following error when I send SOAP RPC request to server which running on my local machine. ...

[aspn.activestate.com/ASP/Message/Apache-Soap-Users/1546026](http://aspn.activestate.com/ASP/Message/Apache-Soap-Users/1546026) - 31k -

[Cached](#) - [Similar pages](#)

[More results from aspn.activestate.com »](#)

### [Java Technologies for Web Services - Soap client No Deserializer ...](#)

[SOAPException: faultCode=SOAP-ENV:Client; msg=**No Deserializer found to deserialize** a 'http://services.pagedownweb.com/:ZipToStateResult' using encoding ...

[forum.java.sun.com/thread.jspa?threadID=336781&messageID=1379379](http://forum.java.sun.com/thread.jspa?threadID=336781&messageID=1379379) - 28k -

[Cached](#) - [Similar pages](#)

### [jGuru: SOAPException: faultCode=SOAP-ENV:Client; msg=No ...](#)

IllegalArgumentException: **No Deserializer found to d** ... Re: SOAPException: faultCode=SOAP-ENV:Client; msg=**No Deserializer found to deserialize** Topic: SOAP ...

[www.jguru.com/forums/view.jsp?EID=1285190](http://www.jguru.com/forums/view.jsp?EID=1285190) - 35k - [Cached](#) - [Similar pages](#)

### [No Deserializer found to deserialize a &apos](#)

**No Deserializer found to deserialize** a &apos. Posted by: Steve Watt on Tue Mar 11 17:35:17 EST 2003 in response to Message #76215. I believe the &apos; ...

[www.theserverside.com/discussions/thread.tss?thread\\_id=18285](http://www.theserverside.com/discussions/thread.tss?thread_id=18285) - 44k -

[Cached](#) - [Similar pages](#)

Done

# CCIA: The Problem

SOAP 1.1 Specs:

SOAP arrays are defined as having a type of "SOAP-ENC:Array" or a type derived there from

ArrayOfcom\_clarkston\_cts\_webservice\_value\_TagRange  
WSValue derives from SOAP-Enc:Array

SOAP Server does not accept the derived type but rather requires SOAP-ENC:Array type

The SOAP Server breaks the WS-I rules surrounding arrays

# CCIA: The Solution

xsi:type of tagRangeList needs to be changed from ns2:ArrayOfTagRange to SOAP-ENC:Array

But, How do we alter the message?



# CCIA: Accessing The Request

```
class CustomSoapClient extends SoapClient {  
    function __doRequest($request, $location,  
                        $saction, $version) {  
        $doc = new DOMDocument();  
        $doc->loadXML($request);  
  
        /* Modify SOAP Request held within $doc */  
        return parent::__doRequest($doc->saveXML(),  
                                    $location, $saction, $version);  
    }  
}
```

# CCIA: Modifying The Request

```
$root = $doc->documentElement;
$encprefix = $root->lookupPrefix('...schemas.xmlsoap.org/soap/encoding/');
$xmlpath = new DOMXPath($doc);
$xmlpath->registerNamespace("myschem", 'http://.../XMLSchema-instance');
$xmlpath->registerNamespace("myxsi",
                           'http://schemas.xmlsoap.org/soap/...');
```

```
$query = '/**/**/**/*[@myxsi:arrayType and @myschem:type]';
```

```
if ($nodelists = $xmlpath->query($query)) {
```

```
    foreach ($nodelists AS $node) {
```

```
        if ($attnode =
```

```
            $node->getAttributeNodeNS('http://.../XMLSchema-instance', 'type'))
```

```
        {
```

```
            $attnode->nodeValue = $encprefix.':Array';
```

```
            ...
```

# CCIA: The Altered Request

```
<ns1:executeSearch>  
  <criteria xsi:type="ns2:AnimalSearchCriteriaWSValue">  
    <tagRangeList SOAP-ENC:arrayType="ns2:TagRange[1]"  
      xsi:type="SOAP-ENC:Array">  
      <item xsi:type="ns2:TagRange">  
        <tagType xsi:type="xsd:string">C</tagType>  
        <endTag xsi:type="xsd:string">..00106</endTag>  
        <startTag xsi:type="xsd:string">..00102</startTag>  
      </item>  
    </tagRangeList>  
    <publicAccountId xsi:type="xsd:string"/>  
  </criteria>  
</ns1:executeSearch>
```

# CCIA: Inspecting Altered Request

```
$client_options = array ('trace'=>1);  
$client = new CustomSoapClient($wsdl, $client_options);  
try {  
    /* code edited for brevity */  
    $res = $client->executeSearch(array('tagRangeList'=>$tRangeList,  
                                       'publicAccountId'=>""));  
    $res->__getLastRequest() // Still shows bad message
```

- Messages altered within `__doRequest()` are not reflected when `__getLastRequest()` is called
- View the raw XML from the DOMDocument within `__doRequest`  

```
print $doc->saveXML();
```

# CCIA: Handling SOAP-ENC:Array w/ PHP 5.2

```
$client_options = array(  
    'features'=>SOAP_USE_XSI_ARRAY_TYPE);  
$client = new SoapClient($wsdl, $client_options);  
$stag = '000000124000050000102';  
$etag = '000000124000050000106';  
  
$tag1 = array('startTag'=>$stag, 'endTag'=>$etag,  
             'tagType'=>'C');  
$tRangeList = array($tag1);  
  
$res = $client->executeSearch( array(  
    'tagRangeList'=>$tRangeList,  
    'publicAccountId'=>""));
```

# ExactTarget

- Provide tools and services for email communications
- Focus on Email Marketing
- On-Demand Access
- Personal Interaction via Web-based GUI
- Application integration via REST and/or SOAP
- SOA based SOAP interface

# ExactTarget: Retrieve API information

```
$wsdl = 'https://webservice.exacttarget.com/etframework.wsdl';
```

```
@$client = new SOAPClient($wsdl, array('trace'=>1));
```

```
$types = $client->__getTypes();
```

```
foreach ($types AS $type) {
```

```
    print $type."\n\n";
```

```
}
```

```
$functions = $client->__getFunctions();
```

```
foreach ($functions AS $function) {
```

```
    print $function."\n\n";
```

```
}
```

# ExactTarget Functions

CreateResponse Create(CreateRequest \$parameters)

RetrieveResponseMsg Retrieve(RetrieveRequestMsg \$parameters)

UpdateResponse Update(UpdateRequest \$parameters)

DeleteResponse Delete(DeleteRequest \$parameters)

ExecuteResponseMsg Execute(ExecuteRequestMsg \$parameters)



# ExactTarget: RetrieveRequestMsg

```
struct RetrieveRequestMsg {  
    RetrieveRequest RetrieveRequest;  
}  
  
struct RetrieveRequest {  
    ClientID ClientIDs;  
    string ObjectType;  
    string Properties;  
    FilterPart Filter;  
    AsyncResponse RespondTo;  
    APIProperty PartnerProperties;  
    string ContinueRequest;  
    boolean QueryAllAccounts;  
    boolean RetrieveAllSinceLastBatch;  
}
```

# ExactTarget: Basic List Retrieval

```
@$client = new ExactTargetSoapClient($wsdl, $options);  
/* WS-Security Username handling */  
  
$request = new ExactTarget_RetrieveRequestMsg();  
  
$rr = new ExactTarget_RetrieveRequest();  
$rr->ObjectType = "List";  
$rr->Properties = array("ListName");  
  
$request->RetrieveRequest = $rr;  
  
$response = $client->Retrieve($request);
```

# ExactTarget: List Retrieval Response

```
object(stdClass)#6 (3) {
  ["OverallStatus"]=> string(2) "OK"
  ["RequestID"]=> string(36) "6bb27c71-16e6-4167-
  a57c-32df045174c4"
  ["Results"]=>
  array(2) {
    [0]=> object(stdClass)#12 (3) {
      ["PartnerKey"]=> NULL
      ["ObjectID"]=> NULL
      ["ListName"]=> string(8) "RobsList"
    }
    [1]=> object(stdClass)#8 (3) {
      ["PartnerKey"]=> NULL
      ["ObjectID"]=> NULL
      ["ListName"]=> string(12) "New Rob List"
```

# ExactTarget: FilterPart Definition

**From RetrieveRequest Definition:**

FilterPart Filter;

**FilterPart Definition:**

```
struct FilterPart { }
```

# ExactTarget: Using a FilterPart

```
<complexType name="SimpleFilterPart">
  <complexContent>
    <extension base="tns:FilterPart">
      <sequence>
        <element name="Property" type="xsd:string"
          minOccurs="1" maxOccurs="1" />
        <element name="SimpleOperator" type="tns:SimpleOperators"
          minOccurs="1" maxOccurs="1" />
        <element name="Value" type="xsd:string"
          minOccurs="0" maxOccurs="unbounded" />
        <element name="DateValue" type="xsd:dateTime"
          minOccurs="0" maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

# ExactTarget: Filtering Requests

```
$sfp = new ExactTarget_SimpleFilterPart();  
$sfp->Property = "ListName";  
$sfp->SimpleOperator =  
    ExactTarget_SimpleOperators::equals;  
  
$sfp->Value = array("RobsList");  
  
$rr->Filter = $sfp;
```

# ExactTarget: Filter Results

```
object(stdClass)#7 (2) {  
  ["OverallStatus"]=>  
  string(60) "Error: Object reference not set  
  to an instance of an object."  
  ["RequestID"]=>  
  string(36) "9c7ebf66-  
  d211-43cf-9984-0adfb4b1e476"  
}
```

**This Doesn't Look Right!**

# ExactTarget: Examining The Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=". . ."
    xmlns:ns1="http://exacttarget.com/wsd/partnerAPI">
<SOAP-ENV:Body>

    <ns1:RetrieveRequestMsg>
        <ns1:RetrieveRequest>
            <ns1:ObjectType>List</ns1:ObjectType>
            <ns1:Properties>ListName</ns1:Properties>

            <ns1:Filter/> <!-- Where is our Filter??? -->

        </ns1:RetrieveRequest>
    </ns1:RetrieveRequestMsg>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



# ExactTarget: The Problem

RetrieveRequest calls for a FilterPart type

FilterPart is an abstract type

SimpleFilterPart type passed in

PHP leaves type as default

## Now What Do We Do?

# SoapVar To The Rescue!

- Set Encoding for data
- Set Type information
- Set Node information

# ExactTarget: Using The SoapVar Class

```
class SoapVar {  
    __construct ( mixed $data,  
                 int $encoding  
                 [, string $type_name  
                 [, string $type_namespace  
                 [, string $node_name  
                 [, string $node_namespace  
                 ]]) )  
}
```

# ExactTarget: Using The SoapVar Class

```
$soapvar = new SoapVar(  
    $sfp, /* data */  
    SOAP_ENC_OBJECT, /* encoding */  
    'SimpleFilterPart', /* type_name */  
    "http://exacttarget.com/wsdl/partnerAPI"  
    /* type ns */  
);  
  
$rr->Filter = $soapvar;
```

# ExactTarget: Filtered Results

```
object(stdClass)#8 (3) {  
  ["OverallStatus"]=>string(2) "OK"  
  ["RequestID"]=>string(36) "638f2307-xxxxxx"  
  ["Results"]=>  
    object(stdClass)#14 (3) {  
      ["PartnerKey"]=>NULL  
      ["ObjectID"]=>NULL  
      ["ListName"]=>string(8) "RobsList"  
    }  
}
```

# ExactTarget: Examining The Request (Take 2)

```
<!-- Only internal struct of envelope -->  
<ns1:RetrieveRequest>  
  <ns1:ObjectType>List</ns1:ObjectType>  
  <ns1:Properties>ListName</ns1:Properties>  
  
  <ns1:Filter xsi:type="ns1:SimpleFilterPart">  
    <ns1:Property>ListName</ns1:Property>  
    <ns1:SimpleOper...>equals</ns1:SimpleOper...>  
    <ns1:Value>RobsList</ns1:Value>  
  </ns1:Filter>
```

# SOAP Client Headers

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

```
  <SOAP-ENV:Header />
```

```
  <SOAP-ENV:Body>
```

```
    </SOAP-ENV:Body>
```

```
  </SOAP-ENV:Envelope>
```

# SOAP Client Headers

```
<SOAP-ENV:Header>
```

```
  <wsse:Security xmlns:wsse="...wss-wssecurity-  
secext-1.0.xsd">
```

```
    <wsse:UsernameToken>
```

```
      <wsse:Username>myname </wsse:Username>
```

```
      <wsse:Password>mypass </wsse:Password>
```

```
    </wsse:UsernameToken>
```

```
  </wsse:Security>
```

```
</SOAP-ENV:Header>
```



# SOAP Client Headers: Our Structure

```
class wsUsernameToken {  
    public $Username;  
    public $Password;  
}  
  
class wsSec {  
    public $UsernameToken;  
}
```

# SOAP Headers: Assembling

```
$token = new wsUsernameToken;  
$token->Username = 'myname';  
$token->Password = 'mypass';  
$wsec = new wsSec;  
$wsec->UsernameToken = $token;  
$header = new SoapHeader($wssens, "Security", $wsec,  
                        false);  
$soapClient->__setSoapHeaders(array($header));
```

# SOAP Headers: Not What We Want

```
<SOAP-ENV:Body . . . xmlns:ns2="...wss-wssecurity-secext-1.0.xsd">  
  <SOAP-ENV:Header>  
    <ns2:Security>  
      <UsernameToken>  
        <Username>myname</Username>  
        <Password>mypass</Password>  
      </UsernameToken>  
    </ns2:Security>  
  </SOAP-ENV:Header>
```

# SOAP Headers: Our Way

```
$wsec = new wsSec;  
$token = new wsUsernameToken;  
$token->Username = new SoapVar("myname", XSD_STRING,  
                                NULL, NULL, NULL, $wssens);  
$token->Password = new SoapVar("mypass", XSD_STRING,  
                                NULL, NULL, NULL, $wssens);  
  
$var = new SoapVar($token, SOAP_ENC_OBJECT,  
                  NULL, NULL, NULL, $wssens);  
$wsec->UsernameToken = $var;  
  
$header = new SoapHeader($wssens, "Security", $wsec, false);  
$soapClient->__setSoapHeaders(array($header));
```

# SOAP Header: Finally!

```
</SOAP-ENV:Header>
```

```
<ns2:Security>
```

```
<ns2:UsernameToken>
```

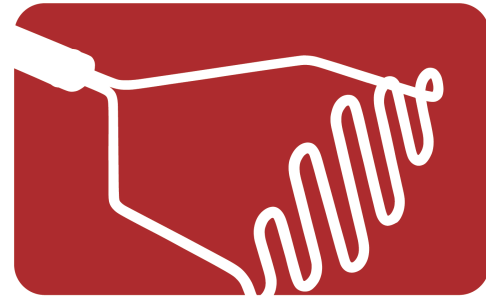
```
<ns2:Username>myname</ns2:Username>
```

```
<ns2:Password>mypass</ns2:Password>
```

```
</ns2:UsernameToken>
```

```
</ns2:Security>
```

```
</SOAP-ENV:Header>
```



**MASHERY**

# Server Sided SOAP

**Working through server sided issues**

# SoapServer: WSDL Caching

**DISABLE WSDL CACHING  
WHEN DEVELOPING!**

```
ini_set("soap.wsdl_cache_enabled", "0");
```

# SoapServer: Generating The WSDL

- Integrated Generators
  - Services\_Webservices
    - [http://pear.php.net/package/Services\\_Webservice](http://pear.php.net/package/Services_Webservice)
  - PRADO
    - <http://pradosoft.com/demos/quickstart/?page=Services.SoapService>
  - WSO2 WSF/PHP (<http://wso2.org/projects/wsf/php>)
- External Tools
  - Eclipse WSDL Editor in Web Tools Platform (WTP)
  - Zend Studio (<http://www.zend.com/en/products/studio/>)



# WSDL: Services\_WebServices

```
include_once('Services/Webservice.php');
class myService extends Services_Webservice {
    /**
     * Says "Hello!"
     *
     * @param int
     * @return string
     */
    public function hello($i )
    {
        //create some logic here
        return 'myString';
    }
}
```

# SoapServer: Serving Up The WSDL

## Server Script: myserv.php

```
/* Helper functions here */  
  
$soapSrv = new SoapServer('helloworld.wsdl');  
  
/* Register functions */  
  
$soapSrv->handle();
```

## Using the SoapClient:

```
$soapClient = new SoapClient(  
    'http://localhost/myserv.php?wsdl');
```

# SoapServer: Handling Requests

- The typical method to process a SOAP request  
`$soapServer->handle();`
- Request may be passed to handler  
`$soapServer->handle($request);`
- Passing in the request can be handy
  - You can be guaranteed of the same request while debugging
  - Debugging can be performed via CLI
  - Requests can be modified prior to being processed

# SoapServer: Handling Requests

```
/* create the initial request to be re-used */  
$request = file_get_contents("php://input");  
file_save_contents('debugging.xml', $request);  
  
/* retrieve saved request on subsequent calls  
$request = file_get_contents('debugging.xml');  
*/  
  
$server = new SoapServer($wsdl);  
/* Setup function handlers */  
$server->handle($request);
```

# SoapServer: Processing Undefined Headers

```
ini_set("soap.wsdl_cache_enabled", 0);  
  
$data = file_get_contents('php://input');  
  
/* Handle headers here */  
  
function doEcho() {  
    return array("out" => time());  
}  
  
$soapSrv = new SoapServer('helloworld.wsdl');  
$soapSrv->addFunction('doEcho');  
  
$soapSrv->handle($data);
```

# SoapServer: Processing Undefined Headers

```
$sxe = simplexml_load_file('php://input');  
$sxe->registerXPathNamespace('mywsse', $wsens);  
  
$secnode = $sxe->xpath('//mywsse:Security');  
if (! empty($secnode)) {  
    $sec = $secnode[0];  
    $token = $sec->children($wsens)->UsernameToken;  
    if ($token->Username != 'myname' &&  
        $token->Password != 'mypass') {  
        throw new SoapFault("Server", 'Bad Credentials');  
    }  
}  
$data = $sxe->asXML();
```

# SoapServer: Handling Headers

There has to be an easier way!



# SoapServer: Handling Headers

```
<wsdl:message name="GetEchoAuthenticationInfo">  
  <wsdl:part name="AuthenticationInfo" element="tns:AuthenticationInfo" />  
</wsdl:message>
```

```
<soap:operation soapAction="http://www.example.org/  
helloworld/doEcho"/>
```

```
<wsdl:input>  
  <soap:body use="literal"/>
```

```
<soap:header  
  message="tns:GetEchoAuthenticationInfo"  
  part="AuthenticationInfo" use="literal" />
```

```
</wsdl:input>
```



# SoapServer: Our Client Request

```
<SOAP-ENV:Envelope  
xmlns:SOAP-ENV="...xmlsoap.org/soap/envelope/"  
xmlns:ns1="http://www.example.org/helloworld/">
```

```
<SOAP-ENV:Header>
```

```
<ns1:AuthenticationInfo>
```

```
<username>myname-a</username>
```

```
<password>mypass</password>
```

```
</ns1:AuthenticationInfo>
```

```
</SOAP-ENV:Header>
```

```
<SOAP-ENV:Body>
```

```
<ns1:doEcho/>
```

```
</SOAP-ENV:Body>
```

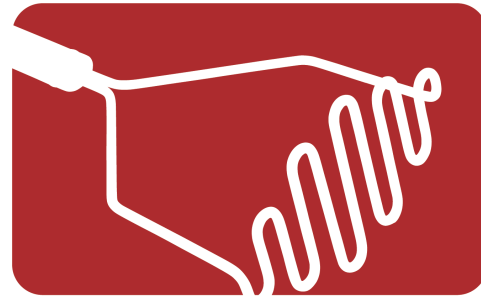
```
</SOAP-ENV:Envelope>
```

# SoapServer: Handling The Header

```
function AuthenticationInfo($auth) {  
    if (($auth->username != "myname") ||  
        ($auth->password != "mypass")) {  
        return new SoapFault("Server", "Bad Credentials");  
    }  
}
```

```
function doEcho() {  
    return array("out" => time());  
}
```

```
$soapSrv = new SoapServer('helloworld-auth.wsdl');  
$soapSrv->addFunction(array('doEcho', 'AuthenticationInfo'));  
$soapSrv->handle();
```



**MASHERY**

# Swiss Army SOAP

**When All Else Fails**

# Typemap: Custom Serial/De-Serialization

- A SOAP option for both client and server
- Specifies functions to use to convert objects to XML (to\_xml) and XML to objects (from\_xml) based on type

```
$typemap = array(  
    array("type_name" => <name of type>,   
        "type_ns"     => <namespace of type>,   
        "to_xml"      => <func to convert obj to xml>,   
        "from_xml"   => <func to convert xml to obj>   
    )  
);
```

# Typemap: SoapClient Example

**RESTful / SOAP translator**

# TypeMap: Rest / SOAP Translator

```
if (empty($_GET['fname']) || empty($_GET['lname'])) {  
    echo '<error>Missing parameters</error>';  
    exit;  
}
```

```
$soapClient = new SOAPClient('http://  
172.16.183.129/transformer.wsdl');
```

```
$myperson = array('fname' => $_GET['fname'],  
                  'lname' => $_GET['lname']);
```

```
$res = $soapClient->getPerson($myperson);  
var_dump($res);
```

# TypeMap: Rest / SOAP Translator Output

`http://localhost/translatorclient.php?  
fname=Joe&lname=Schmoe`

`object(stdClass)[2]`

`public 'fullname' => string 'Joe Schmoe' (length=10)`

`public 'age' => int 94`

`public 'sex' => string 'M' (length=1)`

# TypeMap: Rest / SOAP Translator

## Raw SOAP Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:ns1="http://www.example.org/transformer/">  
  <SOAP-ENV:Body>  
    <ns1:getPersonResponse>  
      <fullname>Joe Schmoe</fullname>  
      <age>49</age>  
      <sex>F</sex>  
    </ns1:getPersonResponse>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```



# TypeMap: Rest / SOAP Translator

```
function getResults($person) {  
    return $person;  
}  
$myperson = array('fname' => $_GET['fname'],  
                 'lname' => $_GET['lname']);  
  
$tmap = array(  
    array('type_name' => 'getPersonResponse',  
        "type_ns"    => '...example.org/transformer/',  
        "to_xml"     => NULL,  
        "from_xml"  => 'getResults'));  
  
$soapClient = new SOAPClient($mywsdl,  
                             array('typemap' => $tmap));  
  
echo $soapClient->getPerson($myperson);
```

# TypeMap: Rest / SOAP Translator

## New Output

```
<ns1:getPersonResponse
xmlns:ns1="...example.org/transformer/">
  <fullname>Joe Schmoe</fullname>
  <age>45</age>
  <sex>M</sex>
</ns1:getPersonResponse>
```

# TypeMap: SoapServer Example

Stock Quote service on top of an XML supported DB

- DB2
- Oracle
- X-Hive/DB
- eXist-db

# Typemap: SoapClient Request

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
  envelope/"
  xmlns:ns1="http://example.org/StockQuote">
  <SOAP-ENV:Body>
    <ns1:Companies xmlns="...ample.org/StockQuote">
      <Symbol>msft</Symbol>
    </ns1:Companies>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# Typemap: WSDL function and Types

```
struct Quote {  
    string Symbol;  
    float Price;  
}
```

```
struct Quotes {  
    Quote Quote;  
}
```

```
struct Companies {  
    string Symbol;  
}
```

```
Quotes getQuote(Companies $parameters)
```

# Typemap: SoapServer Example

```
function quoteToXML($obj) {
    return '<Quotes xmlns="http://example.org/StockQuote">
        <Quote> <Symbol>yhoo</Symbol> <Price>35</Price> </Quote>
        </Quotes>';
}

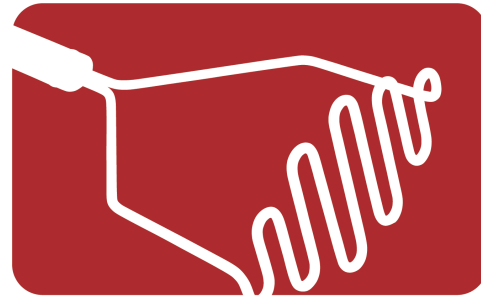
function getQuote($objs) {
    return array("Quote"=>array(array('Symbol'=>'ibm', 'Price'=>1)));
}

$tmap = array(array("type_name"=>"Quotes",
                    "type_ns"=>"http://example.org/StockQuote",
                    "to_xml"=>"quoteToXML"));

$server = new SoapServer('local.wsdl', array("typemap"=>$tmap));
$server->addFunction("getQuote");
$server->handle();
```

# Typemap: SoapServer Response

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
  envelope/"
  xmlns:ns1="http://example.org/StockQuote">
  <SOAP-ENV:Body>
    <ns1:Quotes xmlns="http://example.org/StockQuote">
      <Quote>
        <Symbol>yhoo</Symbol>
        <Price>35</Price>
      </Quote>
    </ns1:Quotes>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



**MASHERY**

**WSO<sub>2</sub>**

**Web Services Framework for PHP**

**WSF/PHP**

**<http://wso2.org/projects/wsf/php>**



# WSO2 WSF/PHP

- Web Service Provider and Consumer
  - SOAP
  - REST
- WS-\* Support
  - WS-Addressing
  - WS-Security / WS-SecurityPolicy
  - WS-ReliableMessaging
- MTOM Support for attachments
- WSDL Generation
- PHP Class generation from WSDL

# WSF/PHP: Raw Requests

```
$client = new WSClient(array("to" =>
    "http://172.16.183.129/transpers.php"));

$message = <<< EOXML
<ns1:getPerson xmlns:ns1="http://www.example.org/
transformer/">
    <fname>Joe</fname>
    <lname>Schmoe</lname>
</ns1:getPerson>
EOXML;

$res = $client->request($message);
var_dump($res);
```

# WSF/PHP: WSDL Request

```
$client = new WSClient(array("wsdl" =>
    "http://172.16.183.129/transformer.wsdl"));

$proxy = $client->getProxy();

$myperson = array('fname' => 'Joe',
    'lname' => 'Schmoe');

$res = $proxy->getPerson($myperson);
var_dump($res);
```

# WSF/PHP: Raw Response

```
function getPerson($message) {
    $sxe = simplexml_load_string($message->str);

    $response = "<getPersonResponse>
        <fullname>{$sxe->fname} {$sxe->lname}</fullname>
        <age>".rand(18,100)."</age>
        <sex>".(rand(0,1)?'M':'F')."</sex>
    </getPersonResponse>";
    return new WSMMessage($response);
}

$soapSrv = new WSService(array("operations" =>
    array("getPerson")));
$soapSrv->reply();
```

# WSF/PHP: WSDL Response

```
function getPersonfunc($fname, $lname) {  
    return array('fullname' => $fname." ".$lname,  
                'age' => rand(18,100),  
                'sex' => (rand(0,1)?'M':'F'));  
}  
  
$operations = array("getPerson"=>"getPersonfunc");  
$opParams = array("getPersonfunc"=>"MIXED");  
  
$service = new WSService(array("wsdl"=>"transformer2.wsdl",  
                               "operations" => $operations,  
                               "opParams"=>$opParams));  
$service->reply();
```

# Questions?

## SOAP Tips, Tricks & Tools

Rob Richards

<http://xri.net/=rob.richards>

<http://www.cdatazone.org>