



DOTS Phone Exchange

Developer's Guide

Version 1.0.0
June 20, 2011
Robert Munoz

Table of Contents

Introduction	3
Integration	3
Request Types	4
Analysis of Request Types	4
XML Parsing	5
Operation Definitions	5
GetSMSInfo	5
GetExchangeInfo	6
GetExchangeInfo_V2	7
Errors	9
Integration and FAQ	10
Can Phone Exchange give me information for Canadian phone numbers?	10
Phone Exchange doesn’t find any data for my phone number!	10
The Sample Code is Giving Strange Errors or is Crashing!	10
Conclusion	11

Introduction

DOTS Phone Exchange is an XML web service that provides information about telephone numbers. With DOTS Phone Exchange users can verify telephone numbers, receive geographic location and carrier information of any phone number in the U.S.

Integration

Integration of DOTS Phone Exchange into user applications is generally a straightforward process. For common programming platforms, such as ASP, ASP.NET, ColdFusion, PHP, etc., Service Objects will likely have sample code available online:

<http://www.serviceobjects.com/support/sample-code/category/17->

If the code you seek is not available online, you can ask Service Objects to build a custom example for you. Email support@serviceobjects.com for more details.

Web Service Structure

Web services provide a standard interface to encapsulate tricky business logic. They allow simple integration of applications via the web. Service Objects has followed web services best practices and come up with some of its own standards to ensure that its web services are as easy to integrate, and as accessible as possible.

The host path, or physical location of the web service is here:

<http://trial.serviceobjects.com/pe/PhoneExchange.asmx>

The location of the WSDL, or Web Service Definition Language document, is here:

<http://trial.serviceobjects.com/pe/PhoneExchange.asmx?WSDL>

(This is also accessible via the "Service Definition" link on the web service page.)

The WSDL is an XML document that defines the interaction web service, meaning its inputs, outputs, operations, and the like. Most likely, you will have another tool read this WSDL and make the operations available to you in your application via some type of proxy class. Whenever your utilities or IDE asks for a WSDL path, you can provide this one.

Every web service has *operations* that it offers to subscribers. These operations, also called methods, contain different functionality and return different outputs. DOTS Phone Exchange has the following operations:

GetSMSInfo – Uses the provided phone numbers to return geographic location, carrier information, and sms domain information of any mobile phone number in the U.S.

GetExchangeInfo - Uses the provided phone numbers to return geographic location and carrier information of any phone number in the U.S.

GetExchangeInfo_V2 – This operation is similar to GetExchangeInfo, but also returns VOIP as a possible line type.

Request Types

DOTS Phone Exchange is a public XML web service that supports SOAP(1.1 and 1.2), POST, and GET request methods. A request type is just the type of web (HTTP) request used to interact with our web services.

GET - All of the input data is in the query string appended to the URL. The response is simple XML.

POST - The input parameters are in the body of the request instead of the query string. The response is simple XML.

SOAP - The input parameters are in an XML SOAP message contained within the body of the request. The response is an XML SOAP message.

Analysis of Request Types

GET is the easiest method to implement by hand because you just set up the URL and query string. It is also easy to debug because you can test the URL + query string in a web browser and see the output.

POST is probably the best method to implement by hand because you do not have to know the specifics of SOAP, and is a little cleaner than passing input parameters in the query string via GET.

SOAP is the best method if you are using a platform that supports SOAP. In many programming environments you can call the service's WSDL file (<http://trial.serviceobjects.com/pe/PhoneExchange.asmx?WSDL>) to create a proxy class to help you interact with the web service. In this case you only have to create an instance of the proxy and use its methods. This completely abstracts the programmer from any complications like sending/receiving web requests/responses as well as any XML parsing. This is typically available in newer environments like PHP version 5, ColdFusion version 7, .NET, etc. Older

languages like PHP version 4 and ColdFusion version 5 will require the use of GET or POST.

XML Parsing

If you are not using an environment that provides a proxy class for you to use, then you will have to parse XML. If you do have a proxy, then it uses an XML parser behind the scenes for you. Although XML parsing can be done without a parser, most programming environments provide easy access to several standard ones. **We strongly recommend that you take advantage of an XML parser.** These parsers may take a few more minutes to integrate if the developer is not familiar with them, but will give your application an added level of security against improper parsing. Without them it is very difficult, even for skilled programmers to write robust code that can handle all cases of XML properly. Because we have very consistent XML you could get away without this extra precaution, but we suggest you use an XML parser anyway to ensure your application is of the highest quality.

Operation Definitions

This section defines the input, output and behavior of the operations in DOTS Phone Exchange.

GetSMSInfo

Returns the geographic location and carrier of the US phone number(s) along with SMS contact information for wireless numbers. This operation is capable of processing multiple phone number transactions in a single request when the phone numbers are separated by a comma.

GetSMSInfo - Inputs

Name	Type	Description
PhoneNumber	String	The ten-digit phone number to validate. Multiple phone numbers are accepted and should be comma separated.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

GetSMSInfo - Outputs

If no errors are encountered a SMSExchangeInfo element will be returned with the following information. If there is an error, an Error object will be returned (explained in next section).

Name	Type	Description
PhoneNumber	String	The verified phone number.
Name	String	The name of the phone carrier associated with the phone number.
City	String	The city in which the validated phone number is registered.
State	String	The state in which the validated phone number is registered.
LineType	String	The line type of the validated phone number. <i>(Possible values are Wireless or Landline).</i>
Latitude	String	The latitude of the registered carrier location.
Longitude	String	The longitude of the registered carrier location.
SMSDomain	String	The registered SMS Domain for the validated mobile number
SMSEmail	String	The SMS email address associated to the mobile number.

GetExchangeInfo

Returns the geographic location and carrier of the US phone number(s). This operation is capable of processing multiple phone number transactions in a single request when the phone numbers are separated by a comma.

GetExchangeInfo- Inputs

Name	Type	Description
PhoneNumber	String	The ten-digit phone number to validate. Multiple phone numbers are accepted and should be comma separated.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

GetExchangeInfo – Outputs

If no errors are encountered an ExchangeInfo element will be returned with the following information. If there is an error, an Error object will be returned (explained in next section).

Name	Type	Description
PhoneNumber	String	The verified phone number.
Name	String	The name of the phone carrier associated with the phone number.
City	String	The city in which the validated phone number is registered.
State	String	The state in which the validated phone number is registered.
LineType	String	The line type of the validated phone number. (<i>Possible values are Wireless or Landline</i>).
Latitude	String	The latitude of the registered carrier location.
Longitude	String	The longitude of the registered carrier location.

GetExchangeInfo_V2

Returns the geographic location and carrier of the US phone number(s). This operation is capable of processing multiple phone number transactions in a single request when the phone numbers are separated by a comma.

GetExchangeInfo_V2 adds the capability to return VOIP as LineType.

GetExchangeInfo_V2 - Inputs

Name	Type	Description
PhoneNumber	String	The ten-digit phone number to validate. Multiple phone numbers are accepted and should be comma separated.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com .

GetExchangeInfo_V2 - Outputs

If no errors are encountered an ExchangeInfo element will be returned with the following information. If there is an error, an Error object will be returned (explained in next section).

Name	Type	Description
PhoneNumber	String	The verified phone number.
Name	String	The name of the phone carrier associated with the phone number.
City	String	The city in which the validated phone number is registered.
State	String	The state in which the validated phone number is registered.
LineType	String	The line type of the validated phone number. <i>(Possible values are Wireless, Landline, or VOIP).</i>
Latitude	String	The latitude of the registered carrier location.
Longitude	String	The longitude of the registered carrier location.

Errors

Generally, an error is anything that happens during a run of DOTS Phone Exchange that causes the service to fail. If an error occurs, an XML error message, similar to the one below, will result instead of the GetExchangeInfo output described above:

Example:

```
<Error>
  <Desc>Your license key does not work on this service.</Desc>
  <DescCode>2</DescCode>
</Error>
```

Error codes in Phone Exchange are the same for all operations. They are as follows:

Error Code 1 – “Input cannot be less than zero length”

This error means the web service did not get any input. The connection to the service was made, and data was transferred, but no parameters were passed that the service could understand.

This error often happens when input is passed to the service with namespaces that the service does not understand. Applying a namespace to any of the parameters (PhoneNumbers or LicenseKey, in this service) will cause this error. Additionally, requests made in the “rpc/encoded” format will cause this error. The only namespace that should appear in any element is the “<http://www.serviceobjects.com>” namespace on the root element as so:

```
<GetExchangeInfo xmlns="http://www.serviceobjects.com/">
```

Error Code 2 – Various descriptions

This error code appears when various errors occur, but are of the expected nature. The most common occurrence is when there is no business contact listing found. Additionally, malformed or incomplete input will cause an error 2. Again, check the Error's Desc element for the exact error.

Failed Authentication Errors:

- “Please provide a valid license key for this web service.”
- “The daily allowable number of transactions for this license key has been exceeded.”
- “The monthly allowable number of transactions for this license key has been exceeded.”
- “The total allowable number of transactions for this license key has been exceeded.”
- “There are not enough transactions available. Check your daily/monthly transaction limits.”
- “This license key has expired.”
- “This license key has not yet been activated.”

“This operation is limited to specific usage. Please contact customer service to have your key enabled.”

“Your license key does not work on this service.”

Web Service Invocation Errors:

“Error initializing service”

“Please input a valid 10 digit phone number.”

Error Code 3/4/5 – Various descriptions

An error code 4 is a fatal error and it means something has seriously gone wrong. You should never see an error code 4 in a live production environment.

Integration and FAQ

Integrating Phone Exchange into your application should be easy and straightforward. If you are using a common platform, Service Objects may already have sample code built that you can use:

<http://www.serviceobjects.com/support/sample-code/category/17->

However, if you are using a common platform that does not already have sample code, you can ask Service Objects to build you an example. Email support@serviceobjects.com for more details.

Can Phone Exchange give me information for Canadian phone numbers?

Yes. Phone Exchange can verify the telco exchange of any telephone number combination in the North American Numbering Plan.

Phone Exchange doesn’t find any data for my phone number!

You may have mistyped your phone number. A valid phone number will always return exchange information, whether it is wireless or landline, connected or disconnected. If you are certain that the number is valid and that it is part of the North American Numbering Plan, then please let us know at support@serviceobjects.com.

The Sample Code is Giving Strange Errors or is Crashing!

Most likely, the sample code cannot connect to ServiceObjects. Many environments will not allow you to connect out on port 80, or will clip out XML data from these requests/responses.

The easiest way to check for this is to open a browser on the machine running the sample code. In your browser, navigate to:

<http://trial.serviceobjects.com/pe/PhoneExchange.aspx>

Then try to run one of the operations with your trial key. If you get a browser error, or get no data back, then the sample code isn't able to connect, either. Contact your systems administrator to resolve why you are not able to connect to ServiceObjects.

Conclusion

Service Objects is proud to offer you a free trial of DOTS Phone Exchange.

Sign up today for a free trial at:

<http://www.serviceobjects.com/products/phone/phone-exchange>

Other technical questions or concerns can be directed to support@serviceobjects.com.

If you are interested in purchasing DOTS Phone Exchange, please contact sales@serviceobjects.com.

We welcome your feedback! Please do not hesitate to let us know what you think of our web services, documentation, or customer support.

Service Objects, Inc.
Insight on Demand

133 E. De la Guerra St., #10
Santa Barbara, CA 93101
Tel: 805.963.1700
Fax: 805.963.9179
www.serviceobjects.com

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.