DOTS Name Validation

Introduction

DOTS Name Validation is an XML web service that provides information about a person's name. With DOTS Name Validation users can validate names, verify name accuracy, fix unordered names and return gender information.

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/nw/NameWizard.asmx

The location of the WSDL, or Web Service Definition Language document, is here (This is also accessible via the "Service Definition" link on the web service page.)
http://trial.serviceobjects.com/nw/NameWizard.asmx?WSDL

> **Important Note!**
> SOAP is done via POST, only with special XML markup in the post-body.

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 – methods that do different work and return different output. Examining the link above, you will notice several of these operations available, which are described in detail later on.

GetNameInfo – Uses the provided name to return information such as gender, name origin, and validity results.
GetNameInfo_V2 - Uses the provided name to return information such as gender, name origin, validity results, and possible related names.
GetOrderedName - Given two names, this service determines which the likely first name is, and which the likely last name is.

Request Types

DOTS Name Validation 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/nw/NameWizard.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 Name Validation.

GetNameInfo

Uses the provided name to return information such as gender, name origin, and validity results.

GetNameInfo Inputs

Name	Type	Description
Name	String	The name to validate.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com.

GetNameInfo Outputs

If no errors are encountered a NameInfo 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
FirstName	String	The parsed first name.
FirstNameCorrected	String	The corrected first name.
MiddleName	String	The parsed middle name.
LastName	String	The parsed last name.
Prefix	String	The parsed out prefix.
Suffix	String	The parsed out suffix.
Gender	String	The gender returned from the supplied name input.
FirstNameOrigin	String	The base origin of the first name e.g. Old-English.
FirstNameFound	Boolean	Boolean value returned from first name exists.
LastNameFound	Boolean	Boolean value returned from last name exists.
PossibleVulgarity	Boolean	Boolean value returned from possible vulgar match.
PossibleCelebrity	Boolean	Boolean value returned from possible celebrity match.
PossibleBogus	Boolean	Boolean value returned from possible bogus match.

Status	String	Output of results from name validation operation.

GetNameInfo_V2

Uses the provided name to return information such as gender, name origin, and validity results. GetNameInfo_V2 also returns proper cased names and possible related names.

GetNameInfo_V2 Inputs

Name	Type	Description
Name	String	The name to validate.
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com.

GetNameInfo_V2 Outputs

If no errors are encountered a NameInfo 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
FirstName	String	The parsed first name.
MiddleName	String	The parsed middle name.
LastName	String	The parsed last name.
FirstNameAlt	String	The parsed first name alt.
MiddleNameAlt	String	The parsed middle name alt.
LastNameAlt	String	The parsed last name alt.
Prefix	String	The parsed out prefix.
Suffix	String	The parsed out suffix.
Gender	String	The gender returned from the supplied name input.
GenderAlt	String	The possible alternate gender returned.
FirstNameOrigin	String	The base origin of the first name e.g. Old-English.
FirstNameFound	Boolean	Boolean value returned from first name exists.
LastNameFound	Boolean	Boolean value returned from last name exists.
FirstNameAltFound	Boolean	Boolean value returned from first name alt found.
LastNameAltFound	Boolean	Boolean value returned from last name alt found.
RelatedNames	String	Possible related names to first name.
PossibleVulgarity	Boolean	Boolean value returned from possible vulgar match.
PossibleCelebrity	Boolean	Boolean value returned from possible celebrity match.
PossibleBogus	Boolean	Boolean value returned from possible bogus match.
PossibleGarbage	Boolean	Boolean value returned from possible garbage name.
Status	String	Output of results from name validation operation.

GetOrderedName

Given two names, this service determines which the likely first name is, and which the likely last name is.

GetOrderedName Inputs

Name	Type	Description
Name	String	The names to order (e.g Last Name First Name)
LicenseKey	String	Your license key to use the service. Sign up for a free trial key at www.serviceobjects.com.

GetOrderedName Outputs

If no errors are encountered a OrderedInfo 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
FirstName	String	The ordered first name.
LastName	String	The ordered last name.
FirstNameFound	String	Value if first name found.
LastNameFound	String	Value if last name found.

Errors

Error codes in DOTS Name Validation are the same for all operations. They are as follows:

Error Number 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 (Name, LicenseKey, etc.) 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: <GetNameInfo xmlns="http://www.serviceobjects.com/">

Error Number 2 - Various descriptions:

This error code appears when various errors occur, but are of the expected nature. Oftentimes, maligned or incomplete input will cause an error 2.
The following is a list of the possible Error Descriptions that may accompany an Error Code 2:

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"
- "Name is required"

Error Number 3/4/5 - Various descriptions:

An error code 3, 4, or 5 is a fatal error and it means something serious has gone wrong. You will never see one of these error codes in a live production environment.

Conclusion

Service Objects is pleased to offer you a free trial of DOTS Name Validation.

Sign up today for a free trial at:
http://www.serviceobjects.com/products/product-directory?wsid=25

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

If you are interested in purchasing DOTS Name Validation, 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.
www.serviceobjects.com