Real-time WebService Processing

Matthew Harshbarger
2017/12/12 13:20

Table of Contents

Contents

Show

   Back To Main Page

Overview

The Address Validation service provides basic delivery point validation (DPV) for addresses deliverable by USPS, including U.S. states, territories and APO/FPO. It does not currently provide multiple responses or suggest alternate addresses.

   In addition to DPV, there are lookup methods for cross-referencing states, counties, cities and zip codes.
WSDL file - machine-readable file describing the functions and data types used by the service. The service is currently implemented as rpc-literal for compatibility with PHP.

   ****

   Looping calls to simulate bulk/batch processing is not allowed as it may cause issues for you and others using this service. Please do not attempt.

   ****

   ~~The development instance of the service is available at http://dev.iowa.gov/addrval/ValidationService.asmx~~

   The test instance of the service is available at http://addrval-test.iowa.gov/ValidationService.asmx

   The production instance of the service is available at http://addrval.iowa.gov/ValidationService.asmx

   '
'NOTE: The Postal Service accepts mailing pieces that use a format known as "dual addresses", where an address contains both a street delivery address and a box delivery address. Dual addresses are supposed to appear on separate lines. The ZIP code used should be for the delivery point, which is the lower line. (Boxes and streets often use different five-digit ZIP codes.) For example,

   XYZ CORP
100 GRANADA DR
BOX 500 <-- delivered to this address
APTOS CA 95001 <-- ZIP for PO BOX

   XYZ CORP
BOX 500
100 GRANADA DR <-- delivered to this address
APTOS CA 95003 <-- ZIP for GRANADA DR

Getting Setup

To implement the Address Validation service in your application you will need to provide a non-expiring Active Directory account and provide an eDAS code for billing.

   To request a non-expiring Active Directory account, follow these directions:

1. Email the Service Desk (just copy and paste in Outlook and it should find it) and CC Darwin Ten Haken
2. In the subject type Non-expiring active directory account for address validation
3. In the body of the email indicate who is requesting the account, what department you are requesting the account for, the application it will be used for, and the password you want setup for the account.
4. When you receive confirmation from the Active Directory team that the account is setup, forward that email to Matthew Harshbarger. Include the eDAS code you want the Address Validation Service charges applied to.
5. After the eDAS code is confirmed with I/3, we will add the account to the application and you can begin processing.

Check the Address Validation Service Availability

To check the Address Validation Service Availability visit this page: ~~http://dev.iowa.gov/addrval/ValidationService.asmx~~ http://addrval-test.iowa.gov/ValidationService.asmx

Validating Addresses Via a Web Browser

Visit ~~http://dev.iowa.gov/addrval/ValidateAddress.aspx~~ http://addrval-test.iowa.gov/ValidationService.asmx if you want to validate a single address with the Address Valdiation Service via a browser.

~~VAlidating Addresses Using the SOA~~

~~When you are trying to connect to or get the WSDL file from the development instance, please use the following URL: http://165.206.47.55:7017/addrval/ValidationService.asmx. You will need to use complete address.~~

~~Before you start development, you will need to request a network conduit to the above URL.~~

~~You may not be able to use a browser to test much functionality, since the web service requires POSTed data that includes authentication data. The SoapUI is commonly used for development and testing against web services (http://www.soapui.org/).~~

Service Methods

SanityCheck

This method does not look up any data and is included for operational monitoring and troubleshooting if needed. Performs the same functions as the check.aspx page, which checks all of the dependencies for the web service and reports and configuration or networking issues that are found.

Parameters: none

Returns: (boolean) TRUE if all tests pass, otherwise an exception is thrown with a description of the first error that was found.

ListPostalCodesForCounty

Provides a list of the valid postal codes for the specified combination of State and County Code.

Parameters:

- state (string): The two-character common abbreviation for the U.S. state or territory (e.g., IA, NY, PR).
- countycode (string): The three-digit IFIPS code for the county within the specified state. For Iowa, this is the commonly-used county code times two, minus one. Example: IA IFIPS county 153 = Iowa common county 77 (Polk).

Returns: (string[]) A list of strings, corresponding to the five-digit ZIP codes that are valid for the specified state and county.

Note: This does not mean that the specified ZIP code falls entirely within the given county, only that some addresses in the county use that ZIP.

ListCountiesForState

Provides a list of the counties within the specified state.

Parameters:

- state (string): The two-character common abbreviation for the U.S. state or territory (e.g., IA, NY, PR).

Returns: (KeyValuePair[]) A list of KeyValuePair structures, using the IFIPS code for the key and the county name for the value. For Iowa, this is the commonly-used county code times two, minus one. Example: IA IFIPS county 153 = Iowa common county 77 (Polk).

ListCitiesForPostalCode

Provides a list of locality data for a given postal code. If the preferred flag is true, then only preferred delivery places will be returned.

Parameters:

- postalcode (string): A five-digit ZIP code.

Returns: (LocalityType[]) A list of LocalityType structures that are deliverable in the specified ZIP.

ValidateAddress

Searches the DPV database and returns the closest matching address to the one submitted. When a match is found, the address returned uses the USPS standards. For example, if a user inputs **100 Second Street**, the address returned would be **100 2nd St.** In the case where no good match is found, the "Error Message" attribute will contain an explanation.

Parameters:

- Auth (AuthData): The object representing the ENTAA UserID, Password, and Token. This can be a department or user, but they will need to be setup in the Address Valdiation ENTAA application with the corresponding eDAS code.

ValidateAddressEx

A variation of ValidateAddress(ParsedAddressType addr) that uses scalar fields instead of the "standard" data structure. Useful for troubleshooting client issues or REST-style interaction with the service (using HTTP GET).

Parameters:

- UserID (string): The ENTAA UserID of the department or user that is making the call to the service.
- Password (string): The password of the UserID.
- Token (string): The ENTAA authentication token of a logged in user. If the individual user is passing their token instead of passing a department login, they must be setup in Enterprise A & A and have a privilege associated with their eDAS code.

Data Structures

KeyValuePair

A utility structure for holding lists of values and their associated (internal) keys.

Field Name	Data Type (Max Length)	Description
Key	String (50)	Unique identifier for the associated value, within this list.
Value	String (100)	The user-friendly display value.

LocalityType

The set of fields that uniquely identify a town or city.

Field Name	Data Type (Max Length)	Description
City	String (30)	Name of the town or city.
State	String (2)	Abbreviation for the U.S. state, territory or Canadian province.
PostalCode	String (10)	ZIP, ZIP+4 or Canadian postal code. For ZIP+4, this value will include the dash.
CountyCode	String (3)	The IFIPS code for the county within the specified state. For Iowa, this is the commonly-used county code times two, minus one. Example: IA IFIPS county 153 = Iowa common county 77 (Polk).
CountyName	String (40)	Common name for the county.

ParsedAddressType

Standardized fields representing a USPS-deliverable address.

Field Name	Data Type (Max Length)	Description
CompanyName	String (30)	Business name (if a business).
StreetAddress	String (64)	Primary line, usually including bldg number, street name, pre- and post-directionals.
StreetAddress2	String (64)	Secondary line, used only for P.O. Boxes or other special delivery types.
Locality	LocalityType	The locality within which the address is located.
Attributes	KeyValuePair[]	A flexible list of extended information about the deliverable address.

AuthData

A utility structure for passing ENTAA authentication object to the service.

Field Name	Data Type (Max Length)	Description
UserID	String (255)	ENTAA UserID of the department account or the individual that has been setup in ENTAA to authenticate.
Password	String (255)	Password of the specified user.
AAToken	String (255)	The ENTAA Token of a user's session that is already authenticated in ENTAA. The user must be setup in ENTAA for the AVS app.

Attribute Definitions

In addition to the standard parsed fields, the ValidateAddress methods return a flexible set of key/value pairs. The following keys are currently returned and provide the values identified for each:

BDI - Business Delivery Indicator. "Y" if the address is a business, otherwise it is a residence.

Carrier route - The postal carrier route code.

~~**Checksum** - A hashed identifier for the specific deliverable address. Can be stored or compared as an abbreviated form of the address.~~ DEPRECATED IN MAF 11/30/2017

Congressional district - The U.S. Congressional district for the address. Numbers are only unique *within* a given state.

Delivery point code - Indicates the type of delivery point:

- **a** - High-rise building
- **b** - A firm or office
- **e** - Post Office Box
- **g** - Alias for another location
- **r** - Rural Route
- **s** - General Delivery

DPV confirm - "Y" if the input record is a valid mailing address listed in the DPV database.

Position 1 = DPV Confirmation Indicator

The DPV Confirmation Indicator is the primary method used by the USPS to determine whether an address was considered deliverable or undeliverable.

Field contains the results of the call to the DPV Confirmation Hash Table: dph.hsa

Y = Address was DPV confirmed for both primary and (if present) secondary numbers.

D = Address was DPV confirmed for the primary number only, and Secondary number information was missing.

S = Address was DPV confirmed for the primary number only, and Secondary number information was present but unconfirmed.

N = Both Primary and (if present) Secondary number information failed to DPV Confirm.

Blank = Address not presented to hash table.

DPV footnotes - a string of two-character standard Postal Service™ codes indicating the quality of the match to the DPV database:

- **BB –** Matched to DPV (all components).
- **CC –** Primary number matched to DPV, but secondary number not matched (present but invalid).
- **N1 –** Primary number matched to DPV, but high-rise address missing secondary number.
- **M1 –** Primary number missing.
- **M3 –** Non-postal Primary number invalid.
- **P1 –** Input Address RR or HC Box number Missing.
- **P3 –** Input Address PO, RR, or HC Box number Invalid.
- **F1 –** Input Address Matched to a Military Address.
- **G1 –** Input Address Matched to a General Delivery Address.
- **U1 –** Input Address Matched to a Unique ZIP Code.

```
A commercial mail-receiving agency (CMRA) is a private business that acts
as the mail-receiving agent for specific clients by providing a delivery
```

address and other services. If the address matches to a CMRA location one of the following footnotes will appear:

- **RR –** Matched to CMRA.

```
* R1 - Matched to CMRA but Secondary Number not Present.
```

Error message - Various messages about limitations or problems found in the validation of a given address.

Private Mailbox - the PMB prefix and PMB number, separated by a blank (e.g., "PMB 011").

RDI - Residential Delivery Indicator. "Y" if the address is a residence, otherwise it is a business.

Geo from lat - The latitude of the start of the address
Geo from long - The longitude of the start of the address
Geo to lat - The latitude of the end of the address
Geo to long - The longitude of the end of the address

The geocoding information provided has a start point and an end point. One address starts at the end of the previous address and continues until the start of the next address. Using the start and end coordinates you can calculate the "frontage" length of the address. Here is a link that searches for calculating the distance (http://www.google.com/search?hl=en&q=calculate+distance+latitude+longitude&aq=9&oq=calculate+distance)