Sprint Web Services User Guide

Address Management Service

Version: 1.1

Date: 07/21/2011

Table of Contents

Audience

This document is designed to help Web Services consumers and application developers integrate their applications with Sprint Web Services. It details the functional and technical aspects of the Web Service interface, including the technologies utilized, the security approach, error handling procedures, and available support options.

Revision History

Date	Version	Description	Author
06/22/11	1.0	Initial Version	Fernando C Filho
07/21/11	1.1	Added production WSDL URL	Fernando C Filho

Contact Information

Source	Description	Email
Web Services Team	Functional/Integration/User Guide Issues	SprintWebServicesIntegrationTeam@sprint.com
Prod Support Team	Production Support Issues	eaipsall@sprint.com

1 Service Overview

The Address Management Service include the following operations:

Operation	Description
ValidateAddress	Returns an address validation
QueryCityState	Returns a list of the cities associated with a US Postal Service zip code.

2 Web Service Operation Details

The Address Management Service contains the following operations:

- ValidateAddress
- QueryCityState

NOTE: Operations with enumerated types are case-sensitive and in some cases restricted. Values must match exactly as documented in the WSDL.

The following sections contain the details of each of these operations, including documentation on every input and output parameter.

They also contain sample SOAP request and response messages. Note that they are filled with test data that is unlikely to be available after publication of this document. Note also that these samples are unsigned (for readability), and actual messages sent to and returned from Sprint's Web Services Gateway should contain the Web Services Security element (`<wsse:Security>`) inside the SOAP Header (`<soapenv:Header>`).

2.1 ValidateAddress

This Service provides calling applications with the ability to request additional validation options like validating the delivery point address validation, to check if the address is deliverable address or not, also to request matching address for an input address if the input address cannot be found with exact or close match.

This API supports USA based Address Validation only (including Peurto Rico and excluding military Addresses). This API doesn't support foreign and USA Military address validation (because CODE1 that we have doesn't support these validations).

If input address cannot be validated by CODE1 validation (i.e. address is not found in Address DB) then input address will be just formatted with the country format specified in input and in response addressNotFoundInCode1Ind will be set to true, to indicate address is not validated by CODE1.

If calling application requests for matching address by setting multipleMatchInd to true, then FE can optionally specify the number of matching addresses to be returned, which can be from 1 to 10. If multipleMatchInd is not set to true, and input Address cannot be validated by CODE1 then input address will be parsed and returned. (follows above scenario). If multipleMatchInd is set to true then the specified number of matches will be returned, and in response multipleMatchesFoundInd will be set to true for each matching address record in response.

2.1.1 Input Parameters

The input request to ValidateAddress API consists of two parts, the WSMessageHeader V2 and the request data structure

trackingMessageHeader

Name	Details	Descriptions
consumerId	Required	The ID of the corporate entity or business unit from which this message originated (e.g. Corporation code or Portal applId).
applicationId	Required	The ID of the application within the corporate entity from which this message originated.
messageId	Required	A unique identifier for this message, to enable tracking, reporting, and correlation.
conversationId	Optional	Each message is part of a conversation between two or more systems in a given session. The Conversation ID is a UID, which can be used to group all messages exchanged within a session. This is field is primarily used for tracking, reporting, root-cause-analysis, and non-repudiation.
timeToLive	Required	Specifies how long in seconds this message is relevant after the message timestamp. The calling application can time-out after 'timeToLive' seconds.
messageDateTimeStamp	Required	The dateTimeStamp of when the message was created. This can be used for tracking and reporting and also by the consuming application to determine if the message is stale.

All front-ends must send a consistent value in MessageHeader structure in the API request message. This is the only way to identify which application is calling the API. This structure will be used for analysis, reporting and troubleshooting purposes. MessageHeader will start to be validated in future releases.

validateAddress

Name	Details	Descriptions
addressInfo	Required	This elements holds the elements of an address that need to be validated
additionalValidationOptions	Optional	This structure contains the additional validation options available for calling application to specify.

addressInfo

Name	Details	Descriptions
addressLine1	Required	First line of the address. This can include the RuralRouting number and POBox number
addressLine2	Optional	Second line of address if any
city	Required	City name
state	Required	US State code
zipcode	Required	5 digit zipcode
zip4Code	Optional	4 digit zipcode extension
country	Optional	3 letter country code or country name. Default USA.
companyName	Optional	Name of company or firm
usUrbanName	Optional	US Urbanization name, mainly for Peurto Rico addresses

additionalValidationOptions

Name	Details	Descriptions
multipleMatchInd	Optional	Indicates whether multiple matching addresses should be returned or not.
maxMatchingAddressesToBeReturned	Optional	Number of matching addresses to be returned.
deliveryPointValidationInd	Optional	If true CODE1validation will check if this Address can be delivered or not. If this flag is set to true, then the corresponding deliveryPointValidationCode and deliveryPointValidationNote in response will be returned as applicable. Note: Currently this is not supported, and

		should not be set in request, an error will be returned if used.

2.1.2 Output Parameters

Name	Values	Details	Descriptions
Faultcode	Fault code, e.g. “Client.804	20	String
Faultstring	Fault reason text e.g., “Server 804: System Exception”.	100	String
Faultactor	Shall have the MessageNumber returned from Amdocs for an exception	10	String
Detail	Contains the error detail text.	n/a	Structure

validateAddressResponse

Name	Details	Descriptions
validatedAddressList	Required	This structure returns the validated and parsed address. If multipleMatchInd is set to true and no exact matching address is found, then this structure may contain more than one validatedAddressInfo structures.

valiatedAddressList

Name	Details	Descriptions
validatedAddressInfo	Required	This structure contains the formatted address and additional address information.

validatedAddressInfo

Name	Details	Descriptions
addressType	Optional	This field indicates what type of address the input Address is. Ex: S – Street Type P – PO Box R – Rural Route
addressLine1	Required	First line of the address. This can include the RuralRouting number and POBox

		number
addressLine2	Optional	Second line of address if any
apartmentType	Optional	Apartment Designator (Ex: APT, STE)
apartmentNumber	Optional	Extracted and validated Apartment number
pobNumber	Optional	Post office box number. If the address is PO BOX.
ruralNumber	Optional	This holds the mail box number on a rural route. If addressType is R.
ruralRoute	Optional	The rural road or highway identifier.
houseNumber	Optional	Extracted and validated US House number. Can be used to map to number (street number) field in Amdocs request, if addressType is 'S'.
streetPrefix	Optional	The type of street when the street type appears before the base street name. For example, AVENUE: 12 AVENUE B
streetName	Optional	Extracted and validated US Street name
streetSuffix	Optional	Extracted Street name suffix. E.g., St., Lane, blvd., etc.
leadingDirection	Optional	Street directional that precedes the street name. For example, the N in 138 N Main Street.
trailingDirection	Optional	Street directional that follows the street name. For example, the N in 456 Washington N.
city	Required	City name
state	Required	US State code
zipcode	Required	5 digit zipcode
zip4Code	Optional	4 digit zipcode extension
country	Optional	3 letter country code. Default USA.
companyName	Optional	Extracted and validated name of company or firm
usUrbanName	Optional	Extracted and validated US Urbanization name, mainly for Peurto Rico addresses
addressNotFoundInCode1Ind	Optional	Indicates whether CODE1 is able validate the address or not (i.e. whether address is found in CODE1 Address DB or not). If Address is not found in DB, then input

		address will be formatted and returned.
multipleMatchesFoundInd	Optional	Indicates if multiple matches are found or not for input address, if multipleMatchInd in request is set to true.
additionalAddressInfo	Optional	Structure containing the additional information about the input address.

additionalAddressInfo

Name	Details	Descriptions
ruralRouteHighwayContractCode	Optional	Validated and extractedRural Route/Highway Contract indicator HC - The address is a Highway Contract address RR - The address is a Rural Route address
privateMailbox	Optional	Extracted Private mailbox info from input.
privateMailboxType	Optional	Validated and extracted type of private mailbox. Possible values include: • Standard • Non-Standard
deliveryPointValidationIndicator	Optional	Indicates the results of Delivery Point Validation (DPV) processing. • Y—DPV confirmed • N—Address not deliverable • S—Primary number was validated but secondary number could not be confirmed • D—Primary number was validated but secondary number was missing from input • M—The address matches multiple valid delivery points • U—Unconfirmed (because the address did not code at the ZIP + 4® level • V—The address caused a false-positive

		violation. For more information
deliveryPointValidationNote	Optional	AA—Input address matched to the ZIP + 4® file. • A1—Input address not matched to the ZIP + 4® file. • BB—Input address matched to DPV (all components). • CC—Input address primary number matched to DPV but secondary number not match (present but not valid). • N1—Input address primary number matched to DPV but high rise address missing secondary number. • M1—Input address primary number missing. • M3—Input address primary number invalid. • P1—Input address missing PO, RR or HC Box number. • RR—Input address matched to CMRA. • R1—Input address matched to CMRA but secondary number not present.
confidence	Optional	The level of confidence assigned to the address being returned. Range is from zero (0) to 100; zero indicates failure, 100 indicates a very high level of confidence that the match results are correct. For multiple matches, the confidence level is 0. For details about how this number is calculated
recordType	Optional	Indicates the record type: • GeneralDelivery • HighRise • FirmRecord • Normal

		• PostOfficeBox • RRHighwayContract • Geographic (non USPS TIGER match) • Auxiliary (match to an auxiliary file)
matchScore	Optional	MatchScore provides an indication of the degree to which the output address is correct. It is significantly different from Confidence in that Confidence is indicates how much the input address changed to obtain a match, whereas the meaning of Match Score varies between U.S. and non-U.S. addresses. For U.S. addresses, MatchScore is a one-digit score on a scale of 0 to 9 that reflects the closeness of the street-name match (after transformations by ValidateAddress, if any). Zero indicates an exact match and 9 indicates the least likely match. If no match was found, this field is blank.

2.1.3 Sample Request Message

```
<soapenv:Envelope
   xmlns:soapenv = "http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:v2 = "http://integration.sprint.com/common/header/WSMessageHeader/v2"
   xmlns:val = "http://integration.sprint.com/interfaces/validateAddress/v1/validateAddress.xsd">
   <soapenv:Header>
      <v2:wsMessageHeader>
         <v2:trackingMessageHeader>
            <v2:applicationId>7UL</v2:applicationId>
            <v2:applicationUserId>7UL</v2:applicationUserId>
            <v2:consumerId>7UL</v2:consumerId>
            <v2:messageId>1001</v2:messageId>
            <v2:conversationId>1001</v2:conversationId>
            <v2:timeToLive>123</v2:timeToLive>
            <v2:messageDateTimeStamp>2011-04-04T00:16:15.587-05:00</v2:messageDateTimeStamp>
         </v2:trackingMessageHeader>
      </v2:wsMessageHeader>
   </soapenv:Header>
   <soapenv:Body>
      <val:validateAddress>
         <val:addressInfo>
```

```
            <val:addressLine1>HERNDON</val:addressLine1>
            <val:city>HERNDON</val:city>
            <val:state>VA</val:state>
            <val:zipCode>20170</val:zipCode>
         </val:addressInfo>
         <val:additionalValidationOptions>
            <val:multipleMatchInd>true</val:multipleMatchInd>
            <val:maxMatchingAddressesToBeReturned>1</val:maxMatchingAddressesToBeReturned>
         </val:additionalValidationOptions>
      </val:validateAddress>
   </soapenv:Body>
</soapenv:Envelope>
```

2.1.4 Sample Response Message

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENC = "http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
   xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance">
   <SOAP-ENV:Header>
      <m:wsMessageHeader xmlns:m =
"http://integration.sprint.com/common/header/WSMessageHeader/v2">
         <m:trackingMessageHeader>
            <m:applicationId>7UL</m:applicationId>
            <m:applicationUserId>7UL</m:applicationUserId>
            <m:consumerId>7UL</m:consumerId>
            <m:messageId>1001</m:messageId>
            <m:conversationId>1001</m:conversationId>
            <m:timeToLive>123</m:timeToLive>
            <m:replyCompletionCode>0</m:replyCompletionCode>
            <m:messageDateTimeStamp>2011-06-22T13:26:49-05:00</m:messageDateTimeStamp>
         </m:trackingMessageHeader>
      </m:wsMessageHeader>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body wsu:Id = "Body-9448be8e-e7d3-48dd-8275-019091103d81" xmlns:wsu =
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <tns:validateAddressResponse xmlns:tns =
"http://integration.sprint.com/interfaces/validateAddress/v1/validateAddress.xsd">
         <tns:validatedAddressList>
            <tns:validatedAddressInfo>
               <tns:addressLine1>Herndon</tns:addressLine1>
               <tns:streetName>Herndon</tns:streetName>
               <tns:city>Herndon</tns:city>
               <tns:state>VA</tns:state>
               <tns:zipCode>20170</tns:zipCode>
               <tns:country>United States Of America</tns:country>
               <tns:addressNotFoundInCode1Ind>true</tns:addressNotFoundInCode1Ind>
               <tns:additionalAddressInfo>
                  <tns:confidence>0</tns:confidence>
               </tns:additionalAddressInfo>
            </tns:validatedAddressInfo>
         </tns:validatedAddressList>
```

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

2.2 QueryCityState

This interface provides the capability to retrieve a list of the cities associated with a US Postal Service zip code.

2.2.1 Input Parameters

The input request to QueryHierarchyInfo API consists of two parts, the WSMessageHeader V2 and the request data structure

trackingMessageHeader

Name	Details	Descriptions
consumerId	Required	The ID of the corporate entity or business unit from which this message originated (e.g. Corporation code or Portal applId).
applicationId	Required	The ID of the application within the corporate entity from which this message originated.
messageId	Required	A unique identifier for this message, to enable tracking, reporting, and correlation.
conversationId	Optional	Each message is part of a conversation between two or more systems in a given session. The Conversation ID is a UID, which can be used to group all messages exchanged within a session. This is field is primarily used for tracking, reporting, root-cause-analysis, and non-repudiation.
timeToLive	Required	Specifies how long in seconds this message is relevant after the message timestamp. The calling application can time-out after 'timeToLive' seconds.
messageDateTimeStamp	Required	The dateTimeStamp of when the message was created. This can be used for tracking and reporting and also by the consuming application to determine if the message is stale.

All front-ends must send a consistent value in MessageHeader structure in the API request message. This is the only way to identify which application is calling the API. This structure will be used for analysis, reporting and troubleshooting purposes. MessageHeader will start to be validated in future releases.

queryCityState

Name	Details	Descriptions
zipCode	Required	Zip Code
zip4Code	Optional	Zip 4 Code
maximumResults	Optional	Maximum Results controls the total number of city rows that can be returned in the reply. If the number of cities associated with a zipcode is greater than the value of maximumResults, the number of cities in the response will equal the value of maximumResults. If maximumResults is not supplied, it will be defaulted to 10.

2.2.2 Output Parameters

Name	Values	Details	Descriptions
Faultcode	Fault code, e.g. "Client.804	20	String
Faultstring	Fault reason text e.g., "Server 804: System Exception".	100	String
Faultactor	Shall have the MessageNumber returned from Amdocs for an exception	10	String
Detail	Contains the error detail text.	n/a	Structure

queryCityStateResponse

Name	Details	Descriptions
cityStateList	Optional	Contains from 0 to 999 cityState Structures

cityState

Name	Details	Descriptions
city	Required	City
state	Required	The 2 character USPS state abbreviation.

2.2.3 Sample Request Message

```
<soapenv:Envelope
```

```
  xmlns:soapenv = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v2 = "http://integration.sprint.com/common/header/WSMessageHeader/v2"
  xmlns:quer = "http://integration.sprint.com/interfaces/queryCityState/v1/queryCityState.xsd">
  <soapenv:Header>
    <v2:wsMessageHeader>
      <v2:trackingMessageHeader>
        <v2:applicationId>7UL</v2:applicationId>
        <v2:applicationUserId>7UL</v2:applicationUserId>
        <v2:consumerId>7UL</v2:consumerId>
        <v2:messageId>1001</v2:messageId>
        <v2:conversationId>1001</v2:conversationId>
        <v2:timeToLive>123</v2:timeToLive>
        <v2:messageDateTimeStamp>2011-04-04T00:16:15.587-05:00</v2:messageDateTimeStamp>
      </v2:trackingMessageHeader>
    </v2:wsMessageHeader>
  </soapenv:Header>
  <soapenv:Body>
    <quer:queryCityState>
      <quer:zipCode>90068</quer:zipCode>
    </quer:queryCityState>
  </soapenv:Body>
</soapenv:Envelope>
```

2.2.4 Sample Response Message

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENC = "http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <m:wsMessageHeader xmlns:m =
"http://integration.sprint.com/common/header/WSMessageHeader/v2">
      <m:trackingMessageHeader>
        <m:applicationId>7UL</m:applicationId>
        <m:applicationUserId>7UL</m:applicationUserId>
        <m:consumerId>7UL</m:consumerId>
        <m:messageId>1001</m:messageId>
        <m:conversationId>1001</m:conversationId>
        <m:timeToLive>123</m:timeToLive>
        <m:replyCompletionCode>0</m:replyCompletionCode>
        <m:messageDateTimeStamp>2011-06-22T13:27:36-05:00</m:messageDateTimeStamp>
      </m:trackingMessageHeader>
    </m:wsMessageHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body wsu:Id = "Body-edd718e4-3978-4a2e-a00b-7a5518caa81f" xmlns:wsu =
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <quer1:queryCityStateResponse xmlns:quer =
"http://integration.sprint.com/integration/interfaces/queryCityStateBtV1/v1/queryCityStateBtV1"
xmlns:quer1 = "http://integration.sprint.com/interfaces/queryCityState/v1/queryCityState.xsd">
      <quer1:cityStateList>
        <quer1:city>HOLLYWOOD</quer1:city>
        <quer1:state>CA</quer1:state>
      </querl:cityStateList>
      <querl:cityStateList>
        <querl:city>LOS ANGELES</querl:city>
        <querl:state>CA</querl:state>
      </querl:cityStateList>
    </querl:queryCityStateResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.3 Error Handling

Sprint’s approach to error handling is to maximize transparency for all exceptional conditions and minimize the occurrence of “under the radar” errors, thus encouraging improved data quality and rapid error resolution. In support of this philosophy, all exceptional or out-of-band responses, including error responses for Sprint’s Web Services will be returned as SOAP Faults conforming to the SOAP 1.1 specification.

SOAP 1.1 allows for customization of the details element to carry implementation-specific fault information. The XML schema ErrorDetails.xsd defines the customization used by Sprint and is also presented in this document.

SOAP 1.1 defines a required fault code element that is used to communicate the codified error. The spec also defines four major fault code categories that can be extended by applications using a dot-delimited format. These are major fault code categories are:

- VersionMismatch
- MustUnderstand
- Client
- Server

The WS-I Basic Profile 1.1 restricts this extension capability by disallowing use of the dot-delimited notation for the faults in the soapenv namespace (http://schemas.xmlsoap.org/soap/envelope/). The Basic Profile does, however, allow for the use of dot-delimited faultcodes provided that the fault codes are NOT in the soapenv namespace.

3 Error Information

There are a number of SOAP Faults commonly returned from Sprint’s Web Services Gateway. These are generally configuration issues and usually easily resolved.

Code	Message	Description	Action
env:Client	Rejected by Policy	Rejected by Policy (from client)	Ensure that the request Soap message is signed with the certificate that was used during the beta integration and that any

			additional authentication fields (if required) are populated correctly. If yes, then contact your EWS contact.
General	Internal error	Internal errors may be due to various reasons such as a schema validation error or due to an outage.	Ensure that the correct service URL is being utilized. If so contact your EWS contact.

3.1 Service-Specific Errors

This interface can generate standard SOAP exceptions as defined by the SOAP specification. In addition, the service itself can also generate certain exceptions that will be communicated via the SOAP Fault construct. The fault string/Text is dependent on the exact error encountered.

Code	**Message**	**Description**
Client.701	Data not found	Used only in cases where the data specifically requested by the front end is not found. If some *other* data is not found while performing an ancillary query as part of the API logic, It will throw Server.704 or simply return successfully without returning the elements associated with the query.
Server.702	Database error	Used when an error (e.g., an ORA-* exception) from a database access is received.
Server.703	Multiple records found	Used when multiple records are found when only one is expected.
Server.704	Application processing error	Used for application processing errors as well as most backend exceptions.
Client.705	Input validation error	Includes XSD schema errors as well as code-checked validations.
Server.706	Output validation error	Includes response XSD schema errors. Response could be generated by the backend or internal system
Client.707	Security validation error	Authentication or Authorization validation has failed
Server.803	Internal database error	Reserved for database connectivity / timeout issues.
Server.804	Communication error	Reserved for backend system connectivity / timeout issues.
Server.805	Service unavailable;	The transaction could not be completed due to high

	experiencing high load	volume experienced by the receiving service.

4 WSDL Information

The following table lists the environments and the corresponding URL location of this of Web Service

Environment	Description
Production	https://webservicesgateway.sprint.com:444/services/wireless/account/AddressManagementService/v1?wsdl
Test RTB1	https://webservicesgatewaytest.sprint.com:444/rtb1/services/wireless/account/AddressManagementService/v1?wsdl

5 Definitions and Acronyms

Term	Definition
XML	eXtensible Markup Language
SOAP	Simple Object Access Protocol
WSDL	Web Service Description Language
XSD	Xml Schema Definition
UDDI	Universal Description, Discovery and Integration
WSM	Web Services Management Platform
EWS	Enterprise Web Services Team
Client	The users of Sprint's Web Services, either Business Customers or Partners and Resellers

6 References

This guide assumes that the reader is conversant with XML, WSDL, SOAP and HTTP specifications. It further assumes a basic knowledge of either Java or Microsoft's.Net web service technologies. The following links provide useful background knowledge of the Web Service standards:

Term	Definition
Apache Axis 1.x, 2.x	http://ws.apache.org/axis/
HTTP	http://www.w3.org/Protocols/rfc2616/rfc2616.htm
SOAP 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
WSDL 1.1	http://www.w3.org/TR/2001/NOTE-wsdl-20010315
Web Services Security 1.0	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
XML 1.1	http://www.w3.org/TR/xml11/
XML Schema 1.1	http://www.w3.org/XML/Schema

XML Schema Definition	http://www.w3.org/2001/XMLSchema.xsd

7 Feedback

Please email to SprintWebServicesIntegrationTeam@sprint.com for feedback. We are trying to make our online user guides as complete, error free, and easy to read as possible. You can help by giving us your feedback. Thank you!