



USPS Web Tools™
Application Programming Interface
Development Guide

Version 4.7 (10/30/2023)

To Our Customers

In the email from the registration process, you received a user ID and password that will allow you to utilize the Web Tools APIs. Any additional documentation or contact with you will be made through the contact person and email address indicated on the registration form.

If you require technical support, contact the USPS Internet Customer Care Center (ICCC). This office is staffed as follows:

- Monday through Friday from 8:00 a.m. to 8:30 p.m. Eastern Time
- Saturdays from 8:00 a.m. to 6:00 p.m. Eastern Time

Email: <https://Email.usps.com/>

Telephone: 1-800-ASK-USPS® (1-800-275-8777)

USPS Customer Commitment

The United States Postal Service fully understands the importance of providing information and service anytime day or night to your Internet and e-commerce customers. For that reason, the USPS is committed to providing 24 x 7 service from our API servers, 365 days a year.

Thank you for helping the U.S. Postal Service provide new Internet services to our shipping customers.

USPS APIs Functional Team
U.S. Postal Service
475 L'Enfant Plaza, SW
Washington, DC 20260-2464

Registered Trademarks

Priority Mail, Priority Mail Flat Rate, Priority Mail International, Priority Mail Regional Rate, Global Express Mail, Global Express Guaranteed, Global Priority Mail, Parcel Post, Parcel Select, First-Class Mail, USPS, USPS Web Tools, and ZIP + 4 are registered trademarks of the U.S. Postal Service.

Priority Mail Express, Priority Mail Express 1-Day, Priority Mail Express 2-Day, Priority Mail Express 3-Day, Priority Mail Express DPO, Priority Mail Express International, Priority Mail Express Intl, Priority Mail Express Military, Priority Mail Express Offshore, Priority Mail 1-Day, Priority Mail 2-Day, Priority Mail 3-Day, Priority Mail DPO, Priority Mail Intl, Priority Mail Military, Priority Mail Offshore, Signature Confirmation, Standard Post, USPS Tracking, ZIP, and ZIP Code are trademarks of the U.S. Postal Service.

Microsoft and Visual Basic are registered trademarks of Microsoft Corporation.

Adobe Acrobat and Adobe Reader are trademarks of Adobe Systems Incorporated.

DUNS is a registered trademark of Dun & Bradstreet.

© Copyright 2015 United States Postal Service

Table of Contents

Introduction	3
User ID Restrictions	3
USPS Corporate Branding Guidelines	4
Preferred Reference	4
Alternative Reference.....	4
Registered Trademarks.....	5
Trademark Ownership and Use	5
Getting Started	6
Administrative Steps	6
Step 1: Register	6
Step 2: Address API Permissions	6
Step 3: eVS Label API Permissions	7
Step 4: Return Label APIs.....	8
Step 5: Tracking API Permissions.....	9
Step 6: Facility API Permissions	10
Step 7: Run XML and Integrate.....	10
Administrative Forms.....	11
Technical Instructions	12
Client Protocols	12
XML Overview	12
Software Development.....	12
Error Handling.....	13
Testing	14
Certify APIs.....	14
Web Tools API Examples.....	14
Testing Tips	23
Updates to APIs	24
References	25
CODE Example.....	26

Introduction

The USPS Web Tools Application Program Interfaces (APIs) allow developers of web-based and shrink-wrapped applications access to the online services of the United States Postal Service (USPS). They provide easy access to shipping information and services for your customers. By integrating these APIs into your web site, your customers can utilize the functions provided by the USPS without ever leaving your web site. Once the APIs are integrated into your website, the USPS Shipping API Server communicates over HTTPS using XML (Extensible Markup Language).

Implementing these APIs requires experienced programmers who are familiar with Internet and web site development tools and techniques.

This document provides guidance and step-by-step instructions for installing the USPS APIs, as well as fulfilling various administrative requirements. The administrative requirements vary between different APIs (e.g., submitting samples of labels for some APIs, signing a licensing agreement for certain software, etc.), and this document provides guidance to navigate through the process for the API you are implementing. *It is imperative that developers read this manual first, as it provides necessary information and procedures prior to use.*

There is a *Web Tools User's Guide* for each API available at <http://www.usps.com/webtools/>. These user guides provide information of the XML transactions to the USPS Shipping API server.

User ID Restrictions

The user ID that you have received is for you or your company to use in accordance with the Terms and Conditions of Use to which you agreed during the registration process. *This user ID are not to be shared with others outside your organization, nor are they to be packaged, distributed, or sold to any other person or entity.* Please refer to the Terms and Conditions of Use Agreement for additional restrictions on the use of your user ID, as well as this document and the APIs contained herein.

Warning: If the U.S. Postal Service discovers use of the same user ID from more than one web site, all users will be subject to immediate loss of access to the USPS server and termination of the licenses granted under the Terms and Conditions of Use.

The documentation and sample code contained in the *Web Tools User Guide* series may be reused and/or distributed to your customers or affiliates to generate awareness, encourage Web Tool use, or provide ease-of-use. However, it is your responsibility to ensure that your customers do not use your user ID. Direct your users to www.usps.com/webtools/ so that they can register, agree to the Terms and Conditions of Use agreement, and receive their own unique user ID.

Note to Software Distributors: The User ID restrictions discussed above are intended for e-tailers that use the USPS Web Tools exclusively within their own web sites. If you plan to distribute software with the USPS Web Tools embedded, you must refer to the "Software Developers" Terms and Conditions of Use" available at <http://www.usps.com/webtools/>.

For more information regarding the USPS Web Tools APIs user ID policy, or for questions regarding the distribution of documentation, please send an email to the Internet Customer Care Center (ICCC) at <https://Email.usps.com/>.

Notice on Web Tools Account Duration: A Web Tools user ID will be deactivated if it has not been used within the last 18 months. If your Web Tools credentials have been deactivated due to inactivity, a new account will need to be created. Reference the Web Tools registration page <https://registration.shippingapis.com> to create new Web Tools account.

USPS Corporate Branding Guidelines

The U.S. Postal Service requests that it is referenced and acknowledged as the source of information for all U.S. Postal Service data that has been acquired through the Internet and/or from other sources. However, this is not mandatory. The following guidelines should be followed for those that want to authenticate and/or validate the data displayed from the U.S. Postal Service.

Preferred Reference

Use one of the following when the USPS is the only referenced source:

- “Information provided by www.usps.com”
- **or**
- Use the official USPS corporate logo or USPS product-specific logos.

Digital copies of USPS corporate trademarks/logos are available through the U.S. Postal Service, Public Policy and Communications Department, Washington, D.C. You can request the USPS corporate logo and/or product-specific logos by e-mailing ilogo@email.usps.gov. Requests will be responded to by e-mail within 10 days. We will review your web site, and if appropriate, provide the logo for usage in accordance with these guidelines and the license grant contained in the Terms and Conditions of Use for Internet Shipping Application Program Interfaces (APIs). If your web page is not available over the Internet, please provide a screen shot of the page where the logo will reside.

When requesting logo(s) you must provide the following information:

- company name
- URL and page where logo will reside
- type of business
- how and where the logo will be used
- contact name
- telephone number
- e-mail address
- desired graphic format, e.g., GIF, TIF, JPEG, etc.
- logo desired:
 - ___ USPS Corporate Eagle Logo
 - ___ Express Mail
 - ___ Priority Mail
 - ___ other (describe)

Alternative Reference

Use one of the following when the USPS is listed with other shipping carriers or web sites:

- United States Postal Service
- U.S. Postal Service
- U.S.P.S. (use period after each initial)

The above alternatives are listed in the order of U.S. Postal Service preference.

Examples:

“U.S. Postal Service delivery standard is two days.”

“U.S.P.S. Priority Mail rate is \$3.95.”

Registered Trademarks

The USPS trademarks listed on page i, as well as any logos requested from USPS Public Policy and Communications Department, should not be altered or abbreviated. You can request product-specific logos by emailing ilogo@email.usps.gov.

Trademark Ownership and Use

USPS trademarks are trademarks owned solely and exclusively by USPS and may be used only in the form, manner and with appropriate legends prescribed by USPS. All advertising and other uses of USPS trademarks must include a legend indicating that USPS trademarks are the property of USPS and that they are being used under license from USPS, together with any other legends or marking that may be required by law. Nothing contained in this document shall be deemed to convey any title or ownership interest to any user except for the nonexclusive rights granted under the Terms and Conditions of Use for Internet Shipping Application Program Interfaces and this document. All use of USPS Trademarks shall inure to the benefit of USPS.

Getting Started

Administrative Steps

For each step described below, it is shown which of the APIs listed requires the action(s) in the step. Many of the steps apply to all APIs.

Step 1: Register

Applicable APIs: All

To use the USPS Web Tools APIs, you must be a registered user. If you have not registered, go to <http://www.usps.com/webtools/> and follow the instructions to register for the APIs.

Upon completion of the registration process, your user ID will be sent via email to the address specified in the registration. Please refer to the Restrictions on page 3 for an important notice regarding the use of your user ID.

Your user ID will be immediately granted access use of the following APIs in the production server:

- Price Calculators
- Address Information
- Service Standards and Commitments
- Package Pickup

Step 2: Address API Permissions

Applicable APIs: Address Verification, City State Lookup, Zip Code Lookup

The Web Tools Address APIs are intended to be used on a transactional basis—i.e., no batch processing or database cleansing—and in conjunction with USPS mailing and shipping. Upon agreement to these terms and conditions customers will be granted access automatically when registering for Web Tools.

Step 3: eVS Label API Permissions

Applicable APIs: eVS, eVSCertify, eVSCancel, eVSExpressMailIntl, eVSExpressMailIntlCertify, eVSPriorityMailIntl, eVSPriorityMailIntlCertify, eVSFirstClassMailIntl, eVSFirstClassMailIntlCertify, eVSCancel

The only option for creating outbound shipping labels through our APIs requires eVS setup/enrollment. The Electronic Verification System (eVS®) is all about doing parcel business with the Postal Service using electronic technology. This manifesting system allows parcel mailers to document and pay postage by transmitting electronic manifest files to the eVS database, which is part of the PostalOne!® <https://gateway.usps.com/eAdmin/view/signin> system. In general, eVS:

- Requires 50 pieces or 50 pounds per mailing
- Requires a permit imprint
- Requires payment via ACH debit daily (no other forms of payment)
- Handles origin entered mail (no destination entry or presort)
- Requires enrollment and new Mailer IDs (MIDs) and permits
- Supports domestic/international/APO/FPO/DPO/PTFAS locations
- Supports Priority Mail Cubic (Contact your Technical Integration Specialist for further details)

For eVS registration, please visit <https://www.usps.com/postalone/evs.htm> and see <https://postalpro.usps.com/pub205> for the Electronic Verification System (eVS®) Business and Technical Guide. Contact eVS@usps.gov for support. If that will not work for you, then you can follow up with sales@usps.gov (or your local Postmaster or USPS Sales Manager) for additional solutions outside of the USPS Web Tools APIs.

Step 4: Return Label APIs

Applicable APIs: USPSReturnsLabel, USPSReturnsLabelCertify

The Web Tools USPS Returns Label API enables customers to receive USPS Returns service labels which are processed using the new automated returns process via Package Platform. USPS Returns service account holders will pay postage and fees through an Enterprise Payment System (EPS) account so that items can be returned by their customers (at no charge to their customers) using merchant provided USPS Returns service labels. The API allows integrators to request USPS Returns service labels for items that can be mailed using Ground Advantage Return Service, Priority Mail Return Service, and Priority Mail Express Return Service. See <https://www.usps.com/business/return-services.htm> for additional USPS Returns service details.

Initial onboarding will include:

- Package Platform enrollment
- USPS Returns enrollment
- EPS account enrollment for electronic funds transfer for payment of USPS Returns service postage
- USPS Web Tools API registration resulting in generation of Web Tools user ID.
- Mailer ID (MID) enrollment
- Customer Registration ID (CRID) enrollment
- Configuration of Destination Delivery Address (where applicable)

The USPS Returns Label API will support the following configuration options determined during initial customer onboarding:

- **Non-manifesting (SSF) Option:** Supports creation of returns label to a single delivery destination address configured per Web Tools User ID with the option of Signature Confirmation extra service. Unique ZIP4 for returns supported. Note: The ability to generate returns labels to multiple delivery destinations requires separate Web Tools User IDs for each destination address.
- **Manifesting (SSF) Option:** Supports creation of returns label to a configured delivery destination address with options for Signature Confirmation and Insurance extra services. Unique ZIP4 for returns supported. A Shipping Services File will be submitted by Web Tools for returns labels on behalf of the integrator.
- **Manifesting (SSF) eFulfillment for Returns Option:** Supports creation of returns label for eFulfillment returns customers to a delivery destination address collected in the API request with options for Signature Confirmation and Insurance extra services. A Shipping Services File will be submitted by Web Tools for returns labels on behalf of the integrator. Note: The USPSReturnsLabel API will validate destination addresses provided by eFulfillment customers for a ZIP+4 or an error will return that includes an invalid ZIP+4.

Step 5: Tracking API Permissions

Applicable APIs: TrackV2, PTSEmail, PTSPod, PTSTPod, PTSRre

The Track and Confirm Web Tools API provides tracking status and delivery information for USPS packages. The data returned by the Package Tracking Web Tools API is intended for display only. Submit your Package Tracking access request at <https://usps.force.com/emailus/s/web-tools-inquiry> and provide your Web Tools User ID. Select 'Tracking APIs' and 'Access for Tracking APIs' and paste the following information inside the Additional Information text box:

- Web Tools User ID:
- Mailer ID (MID): more information below*
- Company Name:
- Company Website:
- Requester First and Last Name:
- Requester Email:
- Requester Phone Number:
- Mailing Address:
- Mailing City:
- Mailing State:
- Mailing Zip Code:
- Web Tools Registration Date:
- API Access Requested: Package Tracking (API=TrackV2)
- Anticipated API volume: (note if daily, weekly, monthly, or annually)
- Shipping done with USPS: Please describe.
- Any additional information:

***Mailer ID Information:** A mailer identification number (MID) is a 6 or 9-digit number assigned to a customer through the USPS Business Customer Gateway (BCG). Please refer to the following links for help.

<https://gateway.usps.com/eAdmin/view/knowledge?securityId=MID>

<https://postalpro.usps.com/mailing/mailler-id>

Please note set up of your MID must be completed prior to granting access. Please contact NCSC-Delivery Confirmation (Email: Delivery.confirmation@usps.gov; Phone: 1-877-264-9693, Option 1).

Four service APIs are offered in conjunction with the Package Tracking API: Track and Confirm by Email (PTSEmail), Proof of Delivery (PTSPod), Track Proof of Delivery (PTSTPod), and Return Receipt Electronic (PTSRre). If requesting access to these other APIs, please follow the above steps and be sure to identify your anticipated API request volume, mailer ID and how you will be utilizing the API.

The Track and Confirm by Email API allows the customer to submit their email address to be notified of current or future tracking activity. Proof of Delivery is a letter that includes the recipient's name and a copy of their signature. The Proof of Delivery API allows the customer to request proof of delivery notification via email. The Track Proof of Delivery API allows the customer to request proof of delivery notification via email. The Return Receipt Electronic API allows the customer to request a copy of the proof of delivery record via email. Track Proof of Delivery is a letter that includes the recipient's name and a copy of their signature.

Step 6: Facility API Permissions

Facility APIs: POLocatorV2Ext, GetLockerInfo

The Web Tools Post Office Locator and Get Locker Info APIs require additional API permissions. Integrators should contact [USPS Internet Customer Care Center](#) and follow the below instructions to submit a request for Web Tools API access.

1. Select “USPS.com” then “Web Tools (APIs)” – reference [Email Us | Web Tools Inquiry \(usps.com\)](#).
2. In the “Account Information”, specify your Web Tools USERID.
3. In the “Issue Information”, specify “API Authorization and Testing” and “Authorization error messages”.
4. In the “Additional Information” section include the following:
 - API Access being requested: Specify “Web Tools GetLockerInfo API” or “USPS Parcel Locker Lookup API” or “POLocatorV2Ext API” or “Post Office Locator API”
 - Indicate expected API volume
 - Indicate use-case (i.e., how API will be used)
 - Provide Mailer ID (MID) if you have one

Step 7: Run XML and Integrate.

Applicable APIs: All

At this point, you have completed all required initiation steps and are now ready to send data to Web Tools servers and begin the full API service. The Web Tools User's Guide for the API you are using will have XML schema information.

For expanded functional testing support of the Web Tools APIs in the testing/staging environment, users should contact USPS Internet Customer Care Center at <https://emailus.usps.com/s/web-tools-inquiry> and select ‘API Authorization and Testing’ and then ‘Staging/Testing Access’.

Staging/Testing endpoint: <https://stg-secure.shippingapis.com/shippingapi.dll>

Production endpoint: <https://secure.shippingapis.com/shippingapi.dll>

Administrative Forms

The following forms are provided in this section:

1. Licensing Agreement for the API Connector Code

Print the Licensing Agreement and email the completed agreement to <https://Emailus.usps.com/>.

2. USPS API Printer Certification Submission Form

Print and complete the Printer Certification Submission form and mail, along with the ten barcoded labels, to:

National Customer Support Center
Attn: Barcode Certification - Web Tools (API)
United States Postal Service
225 N Humphreys Blvd Ste 501
Memphis, TN 38188-1001

You can either print the form from this PDF file and fill out with a pen or copy the form to a Microsoft Word file and fill out electronically before printing and signing.

<p>Important: When printing PDF files with barcodes, be sure that the "Fit to Page" option in the print dialogue box of Adobe Acrobat Reader is unchecked.</p>

Technical Instructions

Client Protocols

HTTPS protocols are used for exchange of data. Requests for Web Tools APIs are sent to a secure server group using a URL of the form:

```
https://secure.shippingaps.com/ShippingAPI.dll?  
API=APINAME&XML=<APINAMERequest USERID="Your account" PASSWORD="Your account  
password">  
<tag>data here</tag><tag1>data</tag1></APINAMERequest>
```

Label API users must include the PASSWORD field with the USERID in the XML request otherwise an error will return. Current integrators are strongly encouraged to provide this information for all APIs in advance of future changes to avoid any service disruptions or integration issues.

Coding to utilize TCP/IP sockets can also provide a basis for a solution though it should be considered only for extreme processing conditions.

Other options are Java objects and Microsoft ActiveX objects that are documented and robust enough for production use. For Java, please reference the Apache Jakarta Project at <https://jakarta.apache.org/>. For Microsoft, reference material for Windows HTTP Services can be located at <https://learn.microsoft.com/>.

XML Overview

The exchanges presented in this document are presented in XML, since that is how the actual data will be structured and transmitted, and it is also a convenient method to document this interface. XML uses a hierarchical (tree) element structure. Each element consists of a start tag of the form <Name>, and an end tag of the form </Name>, between which can be data and other elements. <Name/> is shorthand for <Name></Name>, an element with no data. Attributes such as USERID and PASSWORD can be included with the start tag. *All data and attribute values in this document are for illustration purposes and are to be replaced by the actual values.* Developers must use the order and case for tag names of the sample code contained in this document. However, the Tabs and Carriage Returns in the XML structures are for readability only; there is no need for white space in the actual transmissions.

The XML API offers an interface that enables both request(s) and responses to be fully structured. As shown in the following example, XML's set of self-defining tags allows multiple packages to be tracked with a single request. The ID field is used to match a particular entity in the request with the corresponding entity in the response.

For the latest information on XML from Microsoft and other leading vendors, browse:

- <http://www.w3.org/XML>
- <http://www.xml.com>

Software Development

An API request is the start of a transaction that concludes when the response is returned. Every request is stateless, and no cookies or URL rewrites are used during the exchange. The easiest model to describe is that a request behaves like an HTML page with a <FORM> submission. An example of this looks as such:

```
<HTML>  
<HEAD></HEAD>  
<BODY>  
<FORM method="POST" NAME="MYFORM"  
ACTION="http://uspsshippinagonlineserver/receiver.dll">
```

```
<INPUT TYPE="TEXT" NAME="API" SIZE="25">
<INPUT TYPE="TEXT" NAME="XML" SIZE="8000">
<INPUT TYPE="SUBMIT" NAME="GO" VALUE="GO">
</FORM>
</BODY>
</HTML>
```

The web servers will accept either GET* or POST http requests.

The return content is XML structured data. Refer to the Apache XML web site at <http://xml.apache.org/> for further information on character encodings and characteristics. A return will have this form (formatted for readability):

```
<?XML version="1.0" ?>
<APINAMEResponse>
<tag>return data</tag><tag1>additional</tag1>
</APINAMEResponse>
```

All request and response tags are case sensitive and misspelled tags will generate errors. In addition, order is important because the coordination of some requests require specific tags in a specific order.

Label requests return the label in the requested format encoded as base64 text within the XML. Microsoft XML Core Services can be used to decode the text stream into a binary representation of the requested image. Currently decoding will generate a TIFF (Tagged Information File Format with CCITT group4 compression) or a PDF (<http://www.adobe.com/>).

***Note:** GET http requests have length restrictions, whereas POST http requests do not. Please take this under consideration when determining the request-response method that you choose.

Error Handling

When an error condition exists, a specific XML return is generated. The following example shows the tags that are returned:

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Number>-2147217951</Number>
  <Source>EMI_Respond :EMI:clsEMI.ValidateParameters:
    clsEMI.ProcessRequest;SOLServerIntl.EMI_Respond</Source>
  <Description>Missing value for To Phone number.</Description>
  <HelpFile></HelpFile>
  <HelpContext>1000440</HelpContext>
</Error>
```

For APIs that can handle multiple transactions, the error conditions for requests for multiple responses to be returned together are handled at the response level. For example: a request for rates for two packages. If the addresses are non-existent, an "Error document" is returned to the user. On the other hand, if the address for the first package is acceptable but not the second, the response document contains the information for the first address, but under the XML tag for the second address there is an error tag.

Errors that are further down in the hierarchy also follow the above format.

There is not a compendium of error messages or states that is available. The <Description> tag has the identified problem and should be used to diagnosis the situation. Problems can arise even after thoroughly testing an application because of changes to USPS rates, policies or procedures may be implemented periodically may generate errors that didn't occur previously. **If a request to the Internet Customer Care Center (ICCC) is made for help with diagnosing a problem, please have both the response XML and the request XML available.**

Testing

There is no capacity for load testing in the USPS Web Tools infrastructure. Any account performing capacity/stress testing without prior authorization may be terminated.

A minimal testing function is available for client applications. Some APIs have canned tests that use an explicit XML request and allow for the testing of general programming techniques. As these tests are explicit, they are identified as available only for selected APIs. Processing returns, errors, and parsing techniques can be programmatically done using these canned test scenarios.

There are categories of APIs that provide repeatable responses such as the domestic and international rates APIs or address verification. These APIs are the easiest to use because of the small number of tags. These APIs are included in the default permissions.

If there are questions relating testing options, contact USPS Internet Customer Care Center at <https://emailus.usps.com/s/web-tools-inquiry>.

Certify APIs

APIs that return printed labels or barcodes become available after applying for advanced permissions from the USPS APIs Functional Team. To test label accuracy, a label certification API is available for each label type. These certify APIs can also be used for limited testing because "live" labels are not generated.

Any Web Tools API that contains "Certify" is for testing purposes and will not generate usable labels or tracking numbers. A printed Certify labels will have a "SAMPLE" watermark. To generate a real label and tracking number, you must remove "Certify" from your Web Tools API requests. Integrators should note that our Certify APIs are intended for limited functional verification of API requests and responses. Any secondary processes, such as the creation of a SCAN Form or Shipping Services File, is not supported by Certify APIs.

Web Tools API Examples

The XML request and response samples below illustrate API inputs and outputs for the purpose of illustrating the function of each API.

Replace the USERID and PASSWORD values with the information received from the registration process.

Address Info

Sample Request #1 – Verify

This API collects, "cleanses" an address, and completes the ZIP Code to validate and standardize an address before shipping with USPS.

```
https://secure.shippingapis.com/ShippingAPI.dll?API=Verify
&XML=<AddressValidateRequest USERID="xxxxxxx" PASSWORD="">
  <Address ID="0">
    <Address1></Address1>
    <Address2>6406 Ivy Lane</Address2>
    <City>Greenbelt</City>
    <State>MD</State>
    <Zip5></Zip5>
    <Zip4></Zip4>
  </Address>
</AddressValidateRequest>
```

```
<AddressValidateResponse>
  <Address ID="0">
    <Address2>6406 IVY LN</Address2>
    <City>GREENBELT</City>
    <State>MD</State>
    <Zip5>20770</Zip5>
    <Zip4>1435</Zip4>
    <ReturnText>Default address: The address you entered was found but
more information is needed (such as an apartment, suite, or box number) to
match to a specific address.</ReturnText>
  </Address>
</AddressValidateResponse>
```

Sample Request #2 – City State Lookup

This API is used to find the City and State associated with a ZIP Code.

```
https://secure.shippingapis.com/ShippingAPI.dll?API=CityStateLookup
&XML=<CityStateLookupRequest USERID="xxxxxxx" PASSWORD="">
  <ZipCode ID="0">
    <Zip5>90210</Zip5>
  </ZipCode>
</CityStateLookupRequest>

<CityStateLookupResponse>
  <ZipCode ID="0">
    <Zip5>90210</Zip5>
    <City>BEVERLY HILLS</City>
    <State>CA</State>
  </ZipCode>
</CityStateLookupResponse>
```

Sample Request #3 – City State Lookup

This API will return associated City and State for given ZIP Codes.

```
https://secure.shippingapis.com/ShippingAPI.dll?API=CityStateLookup
&XML=<CityStateLookupRequest USERID="xxxxxxx" PASSWORD="">
  <ZipCode ID="1">
    <Zip5>20770</Zip5>
  </ZipCode>
  <ZipCode ID="3">
    <Zip5>06371</Zip5>
  </ZipCode>
  <ZipCode ID="5">
    <Zip5>90210</Zip5>
  </ZipCode>
</CityStateLookupRequest>

<CityStateLookupResponse>
  <ZipCode ID="1">
    <Zip5>20770</Zip5>
    <City>GREENBELT</City>
    <State>MD</State>
  </ZipCode>
  <ZipCode ID="3">
    <Zip5>06371</Zip5>
    <City>OLD LYME</City>
    <State>CT</State>
  </ZipCode>
  <ZipCode ID="5">
    <Zip5>90210</Zip5>
    <City>BEVERLY HILLS</City>
    <State>CA</State>
  </ZipCode>
</CityStateLookupResponse>
```

```
</ZipCode>
<ZipCode ID="5">
  <Zip5>90210</Zip5>
  <City>BEVERLY HILLS</City>
  <State>CA</State>
</ZipCode>
</CityStateLookupResponse>
```

Package Tracking

Sample Request #1

This request returns the package tracking status for a given barcode. Note: access to this API requires additional approval.

```
https://secure.shippingapis.com/ShippingAPI.dll?API=TrackV2
&XML=<TrackRequest USERID="xxxxxxx" PASSWORD="">
  <TrackID ID="EJXXXXXXXXXXUS"></TrackID>
</TrackRequest>

<TrackResponse>
  <TrackInfo ID="EJXXXXXXXXXXUS">
    <TrackSummary>Your item was delivered at 8:10 am on June 1 in
Wilmington DE 19801.</TrackSummary>
    <TrackDetail>May 30 11:07 am NOTICE LEFT WILMINGTON DE
19801.</TrackDetail>
    <TrackDetail>May 30 10:08 am ARRIVAL AT UNIT WILMINGTON DE
19850.</TrackDetail>
    <TrackDetail>May 29 9:55 am ACCEPT OR PICKUP EDGEWATER NJ
07020.</TrackDetail>
  </TrackInfo>
</TrackResponse>
```

Domestic Price Calculator – RateV4 API

Sample Request #1

```
https://secure.shippingapis.com/ShippingAPI.dll?API=RateV4
&XML=<RateV4Request USERID="xxxxxxx" PASSWORD="">
  <Revision>2</Revision>
  <Package ID="1ST">
    <Service>GROUND ADVANTAGE</Service>
    <ZipOrigination>30677</ZipOrigination>
    <ZipDestination>46750</ZipDestination>
    <Pounds>0</Pounds>
    <Ounces>10</Ounces>
    <Container>VARIABLE</Container>
    <Width></Width>
    <Length></Length>
    <Height></Height>
    <Machinable></Machinable>
    <DropOffTime></DropOffTime>
    <ShipDate>2023-08-08</ShipDate>
    <SortationLevel></SortationLevel>
    <DestinationEntryFacilityType></DestinationEntryFacilityType>
    <ReturnFees>true</ReturnFees>
  </Package>
</RateV4Request>
```

```
<RateV4Response>
  <Package ID="1ST">
    <ZipOrigination>30677</ZipOrigination>
    <ZipDestination>46750</ZipDestination>
    <Pounds>0</Pounds>
    <Ounces>10</Ounces>
    <Container>VARIABLE</Container>
    <Zone>4</Zone>
    <Postage CLASSID="1058">
      <MailService>USPS Ground
Advantage<sup>&#8482;</sup></MailService>
      <Rate>6.35</Rate>
      <SpecialServices>
        <SpecialService>
          <ServiceID>119</ServiceID>
          <ServiceName>Adult Signature Required</ServiceName>
          <Available>true</Available>
          <Price>9.05</Price>
        </SpecialService>
        <SpecialService>
          <ServiceID>120</ServiceID>
          <ServiceName>Adult Signature Restricted
Delivery</ServiceName>
          <Available>true</Available>
          <Price>9.35</Price>
        </SpecialService>
        <SpecialService>
          <ServiceID>171</ServiceID>
          <ServiceName>Certified Mail<sup>&#174;</sup></ServiceName>
Adult Signature Required</ServiceName>
          <Available>true</Available>
          <Price>11.45</Price>
        </SpecialService>
        <SpecialService>
          <ServiceID>172</ServiceID>
          <ServiceName>Certified Mail<sup>&#174;</sup></ServiceName>
Adult Signature Restricted Delivery</ServiceName>
          <Available>true</Available>
          <Price>11.45</Price>
        </SpecialService>
        <SpecialService>
          <ServiceID>103</ServiceID>
          <ServiceName>Collect on Delivery</ServiceName>
          <Available>true</Available>
          <Price>10.60</Price>
          <DeclaredValueRequired>true</DeclaredValueRequired>
          <DueSenderRequired>false</DueSenderRequired>
        </SpecialService>
        <SpecialService>
          <ServiceID>175</ServiceID>
          <ServiceName>Collect on Delivery Restricted
Delivery</ServiceName>
          <Available>true</Available>
          <Price>17.50</Price>
          <DeclaredValueRequired>true</DeclaredValueRequired>
          <DueSenderRequired>false</DueSenderRequired>
        </SpecialService>
      </SpecialServices>
    </Postage CLASSID="1058">
  </Package ID="1ST">
</RateV4Response>
```

```
<SpecialService>
  <ServiceID>100</ServiceID>
  <ServiceName>Insurance</ServiceName>
  <Available>true</Available>
  <Price>0.00</Price>
  <DeclaredValueRequired>true</DeclaredValueRequired>
  <DueSenderRequired>false</DueSenderRequired>
</SpecialService>
<SpecialService>
  <ServiceID>177</ServiceID>
  <ServiceName>Insurance Restricted Delivery</ServiceName>
  <Available>true</Available>
  <Price>19.05</Price>
  <DeclaredValueRequired>true</DeclaredValueRequired>
  <DueSenderRequired>false</DueSenderRequired>
</SpecialService>
<SpecialService>
  <ServiceID>109</ServiceID>
  <ServiceName>Registered
Mail<sup>&#8482;&lt;/sup></ServiceName>
  <Available>true</Available>
  <Price>16.80</Price>
  <DeclaredValueRequired>true</DeclaredValueRequired>
  <DueSenderRequired>false</DueSenderRequired>
</SpecialService>
<SpecialService>
  <ServiceID>176</ServiceID>
  <ServiceName>Registered
Mail<sup>&#8482;&lt;/sup> Restricted Delivery</ServiceName>
  <Available>true</Available>
  <Price>23.70</Price>
  <DeclaredValueRequired>true</DeclaredValueRequired>
  <DueSenderRequired>false</DueSenderRequired>
</SpecialService>
<SpecialService>
  <ServiceID>181</ServiceID>
  <ServiceName>Scan Retention</ServiceName>
  <Available>true</Available>
  <Price>0.99</Price>
</SpecialService>
<SpecialService>
  <ServiceID>183</ServiceID>
  <ServiceName>USPS Label Delivery Service</ServiceName>
  <Available>true</Available>
  <Price>1.25</Price>
</SpecialService>
<SpecialService>
  <ServiceID>108</ServiceID>
  <ServiceName>Signature
Confirmation<sup>&#8482;&lt;/sup></ServiceName>
  <Available>true</Available>
  <Price>4.05</Price>
</SpecialService>
<SpecialService>
  <ServiceID>173</ServiceID>
  <ServiceName>Signature
Confirmation<sup>&#8482;&lt;/sup> Restricted Delivery</ServiceName>
```

```
        <Available>true</Available>
        <Price>10.95</Price>
    </SpecialService>
    <SpecialService>
        <ServiceID>156</ServiceID>
        <ServiceName>Signature
Confirmation<sup>&#8482;</sup> Electronic</ServiceName>
        <Available>true</Available>
        <Price>3.40</Price>
    </SpecialService>
    <SpecialService>
        <ServiceID>174</ServiceID>
        <ServiceName>Signature
Confirmation<sup>&#8482;</sup> Electronic Restricted
Delivery</ServiceName>
        <Available>true</Available>
        <Price>10.30</Price>
    </SpecialService>
    <SpecialService>
        <ServiceID>106</ServiceID>
        <ServiceName>USPS
Tracking<sup>&#174;</sup></ServiceName>
        <Available>true</Available>
        <Price>0.00</Price>
    </SpecialService>
    <SpecialService>
        <ServiceID>155</ServiceID>
        <ServiceName>USPS Tracking<sup>&#174;</sup>
Electronic</ServiceName>
        <Available>true</Available>
        <Price>0.00</Price>
    </SpecialService>
</SpecialServices>
</Postage>
</Package>
```

</RateV4Response>Sample Request #2

```
https://secure.shippingapis.com/ShippingAPI.dll?API=RateV4
&XML=<RateV4Request USERID="xxxxxxx" PASSWORD="">
    <Revision>2</Revision>
    <Package ID="ALL">
        <Service>ONLINE</Service>
        <ZipOrigination>20024</ZipOrigination>
        <ZipDestination>36732</ZipDestination>
        <Pounds>0</Pounds>
        <Ounces>10</Ounces>
        <Container>VARIABLE</Container>
        <Width></Width>
        <Length></Length>
        <Height></Height>
        <Girth></Girth>
        <Machinable>true</Machinable>
        <ReturnLocations>false</ReturnLocations>
        <ReturnServiceInfo>false</ReturnServiceInfo>
        <DropOffTime></DropOffTime>
        <ShipDate>2023-08-08</ShipDate>
```

```
<SortationLevel></SortationLevel>
<DestinationEntryFacilityType>DDU</DestinationEntryFacilityType>
<ReturnFees>>false</ReturnFees>
<ReturnHSCodeReqd>Y</ReturnHSCodeReqd>
</Package>
</RateV4Request>

<RateV4Response>
  <Package ID="ALL">
    <ZipOrigination>20024</ZipOrigination>
    <ZipDestination>36732</ZipDestination>
    <Pounds>0</Pounds>
    <Ounces>10</Ounces>
    <Machinable>TRUE</Machinable>
    <Zone>5</Zone>
    <Postage CLASSID="4079">
      <MailService>Parcel Select Destination Entry Machinable DDU
HAZMAT</MailService>
      <Rate>0.00</Rate>
      <CommercialRate>3.78</CommercialRate>
      <HSCodeReqd>False</HSCodeReqd>
    </Postage>
    <Postage CLASSID="179">
      <MailService>Parcel Select Destination Entry Machinable
DDU</MailService>
      <Rate>0.00</Rate>
      <CommercialRate>3.78</CommercialRate>
      <HSCodeReqd>False</HSCodeReqd>
    </Postage>
    <Postage CLASSID="88">
      <MailService>USPS Connect Local DDU</MailService>
      <Rate>0.00</Rate>
      <CommercialRate>3.95</CommercialRate>
      <HSCodeReqd>False</HSCodeReqd>
    </Postage>
    <Postage CLASSID="89">
      <MailService>USPS Connect Local Flat Rate Bag - Small
DDU</MailService>
      <Rate>0.00</Rate>
      <CommercialRate>4.15</CommercialRate>
      <HSCodeReqd>False</HSCodeReqd>
    </Postage>
    <Postage CLASSID="90">
      <MailService>USPS Connect Local Flat Rate Bag - Large
DDU</MailService>
      <Rate>0.00</Rate>
      <CommercialRate>4.95</CommercialRate>
      <HSCodeReqd>False</HSCodeReqd>
    </Postage>
    <Postage CLASSID="91">
      <MailService>USPS Connect Local Flat Rate Box DDU</MailService>
      <Rate>0.00</Rate>
      <CommercialRate>4.95</CommercialRate>
      <HSCodeReqd>False</HSCodeReqd>
    </Postage>
    <DestinationEntryFacilityType>DDU</DestinationEntryFacilityType>
  </Package>
```

</RateV4Response></RateV4Response>

Service Commitments – SDC Get Locations API

Sample Request #1

The API return is an example of a complex return where there are nested tags.

https://secure.shippingapis.com/ShippingAPI.dll?API=SDCGetLocations
&XML=

```
<SDCGetLocationsRequest USERID="xxxxxxx" PASSWORD="">
  <MailClass>1</MailClass>
  <OriginZIP>63123</OriginZIP>
  <DestinationZIP>30621</DestinationZIP>
  <AcceptDate>2023-03-14</AcceptDate>
  <AcceptTime>0900</AcceptTime>
  <NonEMDetail>True</NonEMDetail>
  <NonEMOriginType>1</NonEMOriginType>
  <NonEMDestType>1</NonEMDestType>
</SDCGetLocationsRequest>

<SDCGetLocationsResponse>
  <Release>2.0</Release>
  <CallerID>4</CallerID>
  <SourceID>004</SourceID>
  <MailClass>1</MailClass>
  <OriginZIP>63123</OriginZIP>
  <OriginCity>SAINT LOUIS</OriginCity>
  <OriginState>MO</OriginState>
  <DestZIP>30621</DestZIP>
  <DestCity>BISHOP</DestCity>
  <DestState>GA</DestState>
  <AcceptDate>2023-03-14</AcceptDate>
  <AcceptTime>0900</AcceptTime>
  <NonExpeditedOriginType>1</NonExpeditedOriginType>
  <Expedited>
    <EAD>2023-03-14</EAD>
    <Commitment>
      <MailClass>1</MailClass>
      <CommitmentName>2-Day</CommitmentName>
      <CommitmentTime>1800</CommitmentTime>
      <CommitmentSeq>A0218</CommitmentSeq>
      <Location>
        <SDD>2023-03-16</SDD>
        <COT>1730</COT>
        <FacType>POST OFFICE</FacType>
        <Street>55 GRASSO PLZ</Street>
        <City>SAINT LOUIS</City>
        <State>MO</State>
        <ZIP>63123</ZIP>
        <IsGuaranteed>1</IsGuaranteed>
      </Location>
    </Commitment>
  </Expedited>
</SDCGetLocationsResponse>
```

Package Pickup**Sample Request #1 – Package Pickup Availability**

This API verifies if an address is eligible to schedule a carrier pickup.

```
https://secure.shippingapis.com/ShippingAPI.dll?API=CarrierPickupAvailability
&XML=<CarrierPickupAvailabilityRequest USERID="xxxxxxx" PASSWORD="">
  <FirmName>ABC Corp.</FirmName>
  <SuiteOrApt>STE 101</SuiteOrApt>
  <Address2>1390 Market Street</Address2>
  <Urbanization></Urbanization>
  <City>SAN FRANCISCO</City>
  <State>CA</State>
  <ZIP5>94102</ZIP5>
  <ZIP4>5403</ZIP4>
</CarrierPickupAvailabilityRequest>

<CarrierPickupAvailabilityResponse>
  <FirmName>ABC CORP.</FirmName>
  <SuiteOrApt>STE 101</SuiteOrApt>
  <Address2>1390 MARKET ST</Address2>
  <Urbanization/>
  <City>SAN FRANCISCO</City>
  <State>CA</State>
  <ZIP5>94102</ZIP5>
  <ZIP4>5403</ZIP4>
  <DayOfWeek>Wednesday</DayOfWeek>
  <Date>02/08/2023</Date>
  <CarrierRoute>C013</CarrierRoute>
</CarrierPickupAvailabilityResponse>
```

Sample Request #2 – Package Pickup Schedule

This API allows integrators to schedule a package pickup.

```
https://secure.shippingapis.com/ShippingAPI.dll?API=CarrierPickupSchedule
&XML=
<CarrierPickupScheduleRequest USERID="xxxxxxx" PASSWORD="">
  <FirstName>Jon</FirstName>
  <LastName>Smith</LastName>
  <FirmName>Essential Hub</FirmName>
  <SuiteOrApt></SuiteOrApt>
  <Address2>8835 Greaves Ln</Address2>
  <Urbanization/>
  <City>Schertz</City>
  <State>TX</State>
  <ZIP5>78154</ZIP5>
  <ZIP4>6220</ZIP4>
  <Phone>770-555-4555</Phone>
  <Extension>201</Extension>
  <Package>
    <ServiceType>PriorityMail</ServiceType>
    <Count>7</Count>
  </Package>
  <EstimatedWeight>50</EstimatedWeight>
  <PackageLocation>Other</PackageLocation>
  <SpecialInstructions>FRONT DESK LATE AFTERNOON
  PICKUP</SpecialInstructions>
```

```
<EmailAddress>myemail@example.com</EmailAddress>
</CarrierPickupScheduleRequest>

<CarrierPickupScheduleResponse>
  <FirstName>JON</FirstName>
  <LastName>SMITH</LastName>
  <FirmName>ESSENTIAL HUB</FirmName>
  <SuiteOrApt/>
  <Address2>8835 GREAVES LN</Address2>
  <Urbanization/>
  <City>SCHERTZ</City>
  <State>TX</State>
  <ZIP5>78154</ZIP5>
  <ZIP4>6220</ZIP4>
  <Phone>770-555-4555</Phone>
  <Extension>201</Extension>
  <Package>
    <ServiceType>PriorityMail</ServiceType>
    <Count>7</Count>
  </Package>
  <EstimatedWeight>50</EstimatedWeight>
  <PackageLocation>Other</PackageLocation>
  <SpecialInstructions>FRONT DESK LATE AFTERNOON
PICKUP</SpecialInstructions>
  <ConfirmationNumber>WTCXXXXXXX</ConfirmationNumber>
  <DayOfWeek>Wednesday</DayOfWeek>
  <Date>03/15/2023</Date>
  <CarrierRoute>R005</CarrierRoute>
  <EmailAddress>myemail@example.com</EmailAddress>
</CarrierPickupScheduleResponse>
```

Testing Tips

- ISO-8859-1 encoding is the expected character set for the requests. Make sure that special characters embedded in any xml tags (such as < or >) are properly escaped (& becomes &).
- If you are calling our APIs with certain methods (i.e., GET method), some characters must be URL encoded. This would be in addition to being XML escaped (& becomes %26amp;).
- Check for proxy servers that may be blocking access.
- If SSL connections are failing, check if the software is accepting the latest security certificate.
- Most APIs complete after a few seconds, but some APIs require the compilation of a lot of interrelated data and may require significantly more time. Use a network trace utility to determine transport time and provide a cushion for timeouts.
- Use a browser to verify request/response interaction if the client application does not provide some form of debug tracking.
- Use the Domestic Mail Manual (DMM) and International Mail Manual (IMM) as a reference for package dimensions, weights, and restrictions for the many types of service. The Postal Explorer web site at <http://pe.usps.gov/> has online versions of these manuals.
- Keep registration contact information accurate. Email notices involving major updates are sent using the email address and person listed as contacts.
- There are no sanctioned online forums, newsgroups, or blogs for Web Tools APIs. The Internet Customer Care Center (ICCC) has current information as to the operational status of the Web

Tools API service and can answer questions about registration and operation status and receive problem reports.

- If you are receiving errors in the SCAN, eVSCancel, or eVSICancel APIs make sure original labels were not created using a Certify label API.
- Ensure API integration is occurring within the same USPS environment. For example, labels created in the testing environment cannot be cancelled in the production environment and vice versa.

Updates to APIs

Periodically, the USPS Web Tools program has major updates based upon Postal Price Changes. This update may necessitate the introduction of new APIs, discontinuance of APIs due to mail service changes, and addition of optional tags to existing APIs. Notice will be sent via email to the registered email address for all users.

To subscribe to the USPS Web Tools distribution list and ensure you receive all future release notes, navigate to our microsite, and click 'Subscribe Now' under the 'Get Email Updates' section.

Address information for verification is updated weekly. Postal rates, restrictions, prohibitions, and other information that effect international mailings are updated as soon as they become available in the IMM and DMM.

References

Web Tools homepage and registration site

<https://www.usps.com/webtools/>

<https://www.usps.com/business/web-tools-apis/documentation-updates.htm>

<https://registration.shippingapis.com>

US Postal Service USPS FAQ home page and Web Tools information page

<https://faq.usps.com/s/>

<https://faq.usps.com/s/article/USPS-Web-Tools-The-Basics>

US Postal Service Postal Explorer site and reference materials for mailing services and prices

<https://pe.usps.com>

<https://pe.usps.com/text/dmm300/notice123.htm>

<https://pe.usps.com/DMM300>

<https://pe.usps.com/text/imm/welcome.htm>

US Postal Service Postal Pro site and reference materials for Electronic Verification System (eVS®)

<https://postalpro.usps.com>

<https://postalpro.usps.com/pub199>

<https://postalpro.usps.com/pub205>

Business Solutions and USPS Return Services

<https://www.usps.com/business/>

<https://www.usps.com/business/return-services.htm>

USPS® Business Customer Gateway

<https://gateway.usps.com/eAdmin/view/signin>

<https://faq.usps.com/s/article/What-is-the-Business-Customer-Gateway-Website>

<https://postalpro.usps.com/bcg/guide>

Online Postal Calculators and Postal Bulletin

<https://postcalc.usps.com/>

<https://about.usps.com/resources/postal-bulletin.htm>

USPS Internet Customer Care Center and Web Tools Inquiry shortcut

<https://emailus.usps.com/s/>

<https://emailus.usps.com/s/web-tools-inquiry>

XML reference for use and SDK type tools for parsing.

<https://www.w3.org/XML>

<https://www.xml.com/>

Open-Source reference for all types of technologies relevant to the use of Web Tools APIs.

<https://jakarta.apache.org/>

CODE Example

The following code example is done as an HTML page that works under Microsoft's Internet Explorer. Copy the page to the clipboard and then paste into notepad or a text editor and save to a file. The browser issues warnings about ActiveX Objects and may require some adjustments to get the page to execute. The JavaScript code assumes that there are no firewalls or proxy servers between the client and the destination server that must be used to complete the communication path.

```
<html><head>
<title>Example courtesy of United States Postal Service</title>
</head><body>
<script type="text/javascript">
var objSrvHTTP = null;
var Version;
var HTTPVersions = new Array( "7.0","6.0","5.0","4.0","3.0" );
for( Version = 0; Version < HTTPVersions.length; Version++ ) {
    try {
        objSrvHTTP = new ActiveXObject("Msxml2.ServerXMLHTTP." +
HTTPVersions[Version]);
        window.status = "Using ServerXMLHTTP." + HTTPVersions[Version];
        break;
    } catch(e) {}
    objSrvHTTP = null;
}
if( objSrvHTTP == null ) { alert( "No HTTP object available" );}
else {
    objSrvHTTP.setOption( 2, 13056 ); // ignore certificate errors
    objSrvHTTP.setProxy(1); // ignore proxy servers (see proxycfg utility)
    objSrvHTTP.setTimeouts( 10000, 10000, 10000, 10000 ); //10 sec timeout
}
function getAPIResponse( style ) {
    var URL = (style) ? APIForm.XMLContent.value :

((APIForm.secure.checked)?"https://":"http://")+APIForm.APIServer.value+
    APIForm.tURI.value+"?API="+APIForm.APIName.value+"&XML="+
    APIForm.XMLContent.value;
// alert( URL );
    try {objSrvHTTP.open ("GET", URL, false); objSrvHTTP.send ();
        XMLResponse.innerHTML = (objSrvHTTP.status == 200) ?
            objSrvHTTP.responseText : "HTTP Error " +
objSrvHTTP.status;}
    catch(e) { XMLResponse.innerHTML = "Error condition " + e.description;
        objSrvHTTP = new
ActiveXObject("Msxml2.ServerXMLHTTP."+HTTPVersions[Version]);}
}
</script>
<form name="APIForm" method="get">
<label for="APIName">API Name</label>
<input type="text" name="APIName" size="25" style="margin:0 51px 0 10px">
<button onClick="getAPIResponse(false);">Build URL</button><br>
<label for="APIServer">API Server</label>
<input type="text" size="30" name="APIServer" style="margin:0 13px 0 7px;">
<input name="secure" type="checkbox">
<label for="secure"> use https</label><br>
<label for="tURI">URI path</label>
```

```
<input name="tURI" type="text" value="/ShippingAPITest.dll" style="margin:0
21px 0 20px;" size="30">
<button onClick="getAPIResponse(true);">Text is URL</button><br>
<textarea id="XMLContent" name="XMLContent" cols="80" rows="5"></textarea>
</form>
<hr>
<div style="width:100%"><span id="XMLResponse"></span></div>
</body></html>
```