Data Affiliate Version 4.0 API Quick Start Guide

> ⓘ **Terms and Conditions**
>
> Please read our Terms and Conditions

Document Information

Back To Contents

This document contains the necessary information required to integrate with All Web Leads' data affiliate platform. For a more detailed description, please reference our full integration specifications found here: Data Affiliates V4.0 API Partner Specification

Getting Started

The All Web Leads API requires an XML payload to be sent via an HTTP POST in a parameter named "XmlString". The XML must conform to our schema. Follow this link to generate sample leads that conform to our schema.

The API requires a username and password for authentication. This will be provided to you by an All Web Leads engineer or by an account manager and must be included in the XML payload (lead data).

All Web Leads also uses an IP address whitelisting system for inbound requests. It is required that your IP addresses are whitelisted before given access to the All Web Leads API.

Supporting XML Schema Files

Back To Contents

- AWL Insurance Lead Type XML Schema
  - Describes the schema for the lead XML that is used across all APIs.

- Send Lead Exclusive API Response XML Schema
  - Describes the schema for the XML response sent back by the fixed pricing/send lead exclusive API.

- Price Presentation Ping/Post Lead API Response XML Schema
  - Describes the schema for the XML response sent back by the ping/post API (used in both ping and post phases).

Staging Endpoint URLs

- Fixed Pricing
  - https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/SendLeadExclusive
- Ping Post
  - Ping
    - https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPingLead

- Post
  - https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPostLead
- Legs Ping Post
  - Ping
    - https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPingLead
  - Post
    - https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPostLead

Production Endpoint URLs

- Fixed Pricing
  - https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/SendLeadExclusive
- Ping Post
  - Ping
    - https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPingLead
  - Post
    - https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPostLead
- Legs Ping Post
  - Ping
    - https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPingLead
  - Post
    - https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPostLead

Carrier Lists

All Web Leads requires mapping to our provided insurance carrier names. The mappings for both undersold and currently insured carriers are found below:

- [AWL Insurance Carrier List - Supported Undersold Carriers](#)

- [AWL Insurance Carrier List - Supported Consumer Carriers](#)

Sample Leads

A web-based test application is available for testing purposes. This simple web application allows partners to generate randomized test leads and post them through the V4.0 API. Alternatively, partners can copy-and-paste their own lead XML into the app for testing purposes as well.

https://dastaging-lm.dev.allwebleads.com/leads/DataAffiliateTestApp.aspx

Approval Testing

[Back To Contents](#)

As part of the onboarding process for new Data Affiliates, we require that sufficient testing be done on the dastaging server before we'll move forward with accepting leads in production. This section explains the specific testing that we require before we'll turn the account on in production.

- The /InsuranceRequest/LeadMetaData/LeadBornOnDateTimeUtc is the date/time that the consumer submitted the lead. This is required and must be represented in [Coordinated Universal Time (UTC)](#). For example, if the server that received the lead is in Central Standard Time (CST) (ie. 14:32:34), then during Daylight Savings Time (DST), UTC is 5 hours ahead (ie. 19:32:34). During the winter when DST is not observed, UTC is 6 hours ahead (ie. 20:32:34). Therefore it is best not to hard code the time difference since it changes with DST.
  - Note: The date/time value needs to be converted ahead of time since a date offset is not supported by our endpoint and thus will not work.
  - Here's a sample LeadBornOnDateTimeUtc using the UTC format. Note the "Z" at the end. This is important since it indicates that the date/time is represented in UTC.

    ```
    <LeadBornOnDateTimeUtc>2015-07-15T13:23:34Z</LeadBornOnDateTimeUtc>
    ```

- The /InsuranceRequest/LeadMetaData/IpAddress is the IP address of the consumer that submitted the lead. While testing, putting a fake 192.168.* or 127.0.0.1 IP address may result in the lead getting rejected with no explanation in the ping/post request since these IPs are blacklisted in our system.

- The /InsuranceRequest/AffiliateInfo/ProductionEnvironment element is for internal testing only and should not be included in the lead XML.

- All optional elements that do not have a value should be omitted. Any element that has the minOccurs="0" in the [Lead XML Schema](#) is an optional element. For example, <LeadSourceID/> or <Dependents/> should be omitted from your XML when there is no value or there are no additional dependents.

- Test leads must be submitted with the CurrentInsurance/CurrentlyInsured element for both the true and false scenarios. When false, the CurrentPolicy node should be omitted. When true, the CurrentPolicy node is required and the Carrier string must be mapped to our [list of supported Carrier strings](#). If the mapping is not done properly, it will result in the lead getting rejected.

- When submitting Auto leads, the Driver and Vehicle ID elements should begin at one instead of zero.

Fixed Pricing aka Send Lead Exclusive API

The "Send Lead Exclusive" API is used to send leads to AWL for consideration for purchase. The lead must be provided in its entirety, including all personally identifiable information **(The "ContactInfo" XML node must be present and complete)**. The AWL system makes a real-time determination of whether or not to accept/purchase the lead based on a number of factors (see your account manager for details regarding your account configuration).

A single post parameter is supported, which allows the passing of lead data in the AWL lead XML format:

- **XmlString** (String)
    - An XML string containing the lead object. The XML string must conform to the AWL Lead XML V1.5 Schema (included with this documentation)
    - XML string must be in UTF-8 format and must NOT include a Unicode byte order mark (BOM)

Example HTTP POST Data Payload

The following represents an example HTTP POST data payload for this API:

XmlString=<raw lead xml>

Ping Post (Ping Phase)

The "Price Presentation Ping Lead" API is used to send leads to the AWL system to receive a bid on the lead (this is often referred to as "price presentation ping-post" in the industry).

This API supports both Exclusive and Undersold (shared) lead buying. A lead presented with zero distribution directives is considered an Exclusive lead. A lead presented with one or more distribution directives is considered an Undersold lead.

The lead must be provided in its entirety, although all personally identifiable information for the primary applicant is considered optional **(The "ContactInfo" XML node can be left out)**. The AWL system evaluates the provided lead data based on a number of factors (see your account manager for details regarding your account configuration), and returns a price estimate constituting a bid on the lead.

When a non-zero price is returned, a "LeadID" value is returned which represents a ping session identifier. The ping session identifier may be used in a follow-up "Price Presentation Post Lead" API call to offer the lead for sale at the previously bid price point. Note that a lead bid price and its ping session identifier is valid only for up to five minutes.

A single post parameter is supported, which allows the passing of lead data in the AWL lead XML format:

- **XmlString** (String)
    - An XML string containing the lead object. The XML string must conform to the AWL Lead XML V1.5 Schema (included with this documentation)
    - XML string must be in UTF-8 format and must NOT include a Unicode byte order mark (BOM)

Example HTTP POST Data Payload

The following represents an example post data payload for this API:

XmlString=<raw lead xml>

Ping Post (Post Phase)

The "Price Presentation Post Lead" API is used to offer leads to sale after those leads have successfully been bid upon by the AWL system via a previous call to the "Price Presentation Ping Lead" API.

This API supports both Exclusive and Undersold (shared) lead buying. A lead presented with zero distribution directives is considered an Exclusive lead. A lead presented with one or more distribution directives is considered an Undersold lead.

The lead must be provided in its entirety, including all personally identifiable information.

This API call must provide the "LeadID" value that was previously returned in a successful call to the "Price Presentation Ping Lead" API. This "LeadID" value represents a session identifier that associates the post with the ping. Note that a lead bid price and its ping session identifier is valid only for up to five minutes. If a call to this API uses a ping session identifier that is more than five (5) minutes old, that call will be rejected.

With the exception of the ping-time optional personally identifying information, the Lead XML provided in this post API must be identical to the XML presented earlier in the associated ping API. If the XML differs, the post will be rejected.

Prior to accepting the lead via this API, the personal identifying information will be validated. If the identifying information cannot be validated, the lead will rejected. Additionally, the personal identifying information will be used to determine if the lead is considered a duplicate within the AWL system. If the lead is a duplicate, it will be rejected (please refer to the "Duplicate Detection" section earlier in this document for additional details).

A single post parameter is supported, which allows the passing of lead data in the AWL lead XML format:

- **LeadID** (String)
  - A string containing the "LeadID" value returned by a previous successful call to the "Price Presentation Ping Lead" API that uniquely identifies the ping session and is used to associate this API call with the price bid returned by the ping
- **XmlString** (String)
  - An XML string containing the lead object. The XML string must conform to the AWL Lead XML V1.5 Schema (included with this documentation)
  - XML string must be in UTF-8 format and must NOT include a Unicode byte order mark (BOM)
  - XML string must be equivalent to XML provided in previous successful call to the "Price Presentation Ping Lead" API

Example HTTP POST Data Payload

The following represents an example post data payload for this API:

LeadID=<ping session identifier>&XmlString=<raw lead xml>

Undersold

The Ping Post API does support the inclusion of distribution directives. When passing in a carrier name, you are required to map your carrier name to one of our supported carriers. If you cannot map a carrier or the agent is independent, please pass in "Independent" as the carrier name with an accompanying license number (this can be found in the example below). The distribution count cannot exceed four or we will reject the lead. Currently the number of distribution directives listed MUST match the value in "DistributionCount".

Partial XML sample with distribution directives:

```
<InsuranceRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001
/XMLSchema">
  <AffiliateInfo>
    <Username>Username</Username>
    <Password>Password</Password>
    <TrackingCampaign>MyCampaign</TrackingCampaign>
    <LeadSourceID>InHouse</LeadSourceID>
  </AffiliateInfo>
  <DistributionDirectives>
    <Cap>1</Cap>
    <DistributionCount>4</DistributionCount>
    <Excludes>
      <Exclude>
        <CarrierName>Independent</CarrierName>
        <LicenseNumber>7893106634983</LicenseNumber>
      </Exclude>
      <Exclude>
        <CarrierName>Farm Bureau/Farm Family/Rural</CarrierName>
      </Exclude>
      <Exclude>
        <CarrierName>Kaiser Permanente</CarrierName>
      </Exclude>
      <Exclude>
        <CarrierName>US Health Group</CarrierName>
      </Exclude>
    </Excludes>
  </DistributionDirectives>
```

Response Structure

The response from the All Web Leads' web service will be an XML structure. The XML indicates whether or not the post was a success or a failure. The structure of the response is described in our schemas below:

- Fixed Price
    - Send Lead Exclusive API Response XML Schema
    - Full description found here
- Ping Post
    - Price Presentation Ping/Post Lead API Response XML Schema
    - Ping Phase full description found here
    - Post Phase full description found here

Detailed Response Messages

Our staging environment supports more detailed responses for the purpose of expediting the integration process. For detailed responses, please include the HTTP POST parameter "DetailedResponse" with a value of "1".

Example: XmlString=<XmlDocument>&DetailedResponse=1

Sample Responses

- Ping Post Success
    - A result of "OK" with an included payout indicates that the lead passed validation checks and we wish to purchase the lead for the returned amount.

```
<?xml version="1.0" encoding="utf-8"?>
<PricePresentationResult xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001
/XMLSchema" xmlns="https://www.insuranceleads.com/partner/PricePresentationResult.xsd">
  <Result>OK</Result>
  <LeadID>1234</LeadID>
  <Payout>3.94</Payout>
</PricePresentationResult>
```

- Ping Post Failure

```
<?xml version="1.0" encoding="utf-8"?>
<PricePresentationResult xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001
/XMLSchema" xmlns="https://www.insuranceleads.com/partner/PricePresentationResult.xsd">
  <Result>ERROR</Result>
  <ErrorType>XML Validation</ErrorType>
  <ErrorDescription>XML Parse/Validation Failure</ErrorDescription>
  <Payout>0</Payout>
</PricePresentationResult>
```

- Fixed Pricing Success

```
<?xml version="1.0" encoding="utf-8"?>
<LeadsUploadResult xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001
/XMLSchema" xmlns="https://www.insuranceleads.com/partner/LeadsUploadResult.xsd">
  <Result>OK</Result>
  <LeadID>12345</LeadID>
  <Payout>1.95</Payout>
</LeadsUploadResult>
```

- Fixed Price Failure With Data Validation Errors

```xml
<?xml version="1.0" encoding="utf-8"?>
<LeadsUploadResult xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001
/XMLSchema" xmlns="https://www.insuranceleads.com/partner/LeadsUploadResult.xsd">
  <Result>ERROR</Result>
  <ErrorType>Data Validation</ErrorType>
  <ErrorDescription>XML Parse/Validation Failure</ErrorDescription>
  <Payout>0</Payout>
  <DataFieldValidation>
    <DataField>nameFirst</DataField>
    <DataFieldErrorType>Fail</DataFieldErrorType>
  </DataFieldValidation>
</LeadsUploadResult>
```

Frequently Asked Questions

Q: What is the name of the HTTP POST parameter that should contain the XML payload (the lead data)?
A: XmlString

Q: In the response, I am seeing the following error description: "Lead was rejected." What does this mean?
A: This means that either your account was not configured to receive this lead or there were no customers in our system to purchase this lead.

Q: How do I tell if a field is required?
A: This is indicated in our schema . Elements with minOccurs="0" are not required.

Q: We do no collect data for a given field but it is required. What should we pass?
A: AWL does not provide required values. If you do not collect all the fields possible in our schema for a particular lead type, you should provide the full list of fields you do collect to your affiliate manager, and they will work with you on which fields must be added, and which can be defaulted or omitted.

Q: What is the LeadSourceId field for in the XML?
A: Please follow this link

Q: What does an error type of "Data Validation" mean?
A: We validate contact information and other select fields. When one of these fields fails validation, we will return this error type.

Q: Can you provide me with a list of your accepted car make and models?
A: We use the Polk database (https://www.ihs.com/btp/polk.html) to fill in our auto make and model data. Other databases can work but will not match exactly. Alternatively, you can also pass a partially obscured VIN. We cannot provide the Polk data to you since this would breach the license agreement with Polk.

Q: My company does not use the Universal leadid.com token, what should I do with that field?
A: If you do not use the Universal leadid.com token, please omit that field from the XML. Note that if you do not pass this token, you must pass the create date (LeadBornOnDateTimeUtc) and IP address of the consumer to be considered TCPA compliant within our system.

Q: I am a call center, what should I pass for the consumer's IP address?
A: Please pass the IP address of your call center.

Q: How do I know if my IP address has been whitelisted?
A: Try hitting any of the endpoint URLs in a browser. The message "Method not allowed" is expected behavior and indicates that your IP address has been whitelisted. If you get back an HTTP connection error then it has not been whitelisted.

Q: What are the acceptable values for "PrimaryUse" in an auto lead?
A: All acceptable values are listed in our schema via an enumeration or described element type.

Q: Should I pass the ProductionEnvironment element as true or false when I am testing?
A: This element is only used for our internal testing and should be omitted from the XML that you are sending.

Q: I am getting a XML/Data validation error, but my XML validates against your schema. What am I doing wrong?
A: If you are performing a post after a ping, please verify that your post phase XML (not including contact information) is identical with that from the ping phase. Having an expiration date in the past or other bogus values that are checked beyond XML validation can trigger this error as well.

Sample Client Code Snippets

Back To Contents

The following simplistic sample client code is provided for example purposes only. AWL does not make claims as to its quality of fitness for a specific purpose.

C# .NET Client Code Snippet

```csharp
using System;
using System.IO;
using System.Net;
using System.Text;

public static String HttpPost()
{
    string url = "https://<server-name>/leads/4.0/LeadServiceHttpPost.svc/SendLeadExclusive";
    string xmlString = "<your-xml-here>";
    NetworkCredential credential = null
    int timeoutSeconds = 30;
    string contentType = "application/x-www-form-urlencoded";

    try
    {
        string postparams = String.Format("XmlString={0}", xmlString);

        HttpWebRequest myReq = (HttpWebRequest) WebRequest.Create(url);

        myReq.Method = "POST";
        myReq.Credentials = credential;
        myReq.ContentType = contentType;

        myReq.Timeout = timeoutSeconds * 1000; // convert to milliseconds
        StreamWriter writer = new StreamWriter(myReq.GetRequestStream());
        writer.Write(postparams);
        writer.Close();

        WebResponse webResponse = myReq.GetResponse();
        Stream receiveStream = webResponse.GetResponseStream();
        Encoding encode = System.Text.Encoding.GetEncoding("utf-8");

        // Pipe the stream to a higher level stream reader
        // with the required encoding format.
        StreamReader readStream = new StreamReader(receiveStream, encode);

        string responseInfo = readStream.ReadToEnd();
        readStream.Close();
        webResponse.Close();

        return responseInfo;
    }
    catch (WebException)
    {
        throw;
    }
}
```

PHP Client Code Snippet

```php
<?php
$endpoint = "https://<server-name>/leads/4.0/LeadServiceHttpPost.svc/SendLeadExclusive";
$content_type = "Content-Type: application/x-www-form-urlencoded";
$lead_xml = "<your-xml-string-here>";

$postValue = urlencode($lead_xml);
$ch = curl_init($endpoint);
curl_setopt($ch, CURLOPT_POST, "XmlString={$postValue}");
curl_setopt($ch, CURLOPT_POSTFIELDS, "XmlString={$postValue}");
curl_setopt($ch, CURLOPT_HTTPHEADER, array($content_type));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

try
{
    $output = curl_exec($ch);
    $success = curl_errno($ch) == 0;
    curl_close($ch);
}
catch (Exception $e)
{
}
?>
```