Data Affiliates V4.0 API Partner Specification

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

Document Info

Back To Contents

Contact Information

Back To Contents

Company Address	Business Contacts	Technical Contacts
All Web Leads, Inc. 7300 FM 2222 Bldg 2 Ste 100 Austin, TX 78730 Office: 888-522-7355	Affiliate Management Team affiliates@allwebleads.com Office: 512-222-4477	AWL Operations Team Alias operations@allwebleads.com

Revision History

Back To Contents

Date	Rev.	Comments
3/21/2011	4.0	New version of document specific to V4.0 API, based on original V3.0 API document [JDR]
4/07/2011	4.1	Updated staging test application information. Added clarification that the actual web service endpoints do not support HTTPS GET protocol. [JDR]
4/13/2011	4.2	Updated for new Health Insurance and Home Insurance support, as well as Undersold support [JDR]
4/25/2011	4.3	Moved document to the AWL extranet wiki [JDR]
5/02/2011	4.4	Fixed incorrect documentation for Price Presentation staging and production URLs [JDR]
5/12/2011	4.5	Updated "Staging Environment Test Application" section to remove note about lead service URL not being defaulted, as it now is. [JDR]
5/17/2011	4.6	Updated spec with latest insurance provider carriers, split list between Undersold supported and current insurance supported carriers. [JDR]
7/26/2011	4.7	Updated spec and schemas to include new Leg Ping and Post Lead APIs.
10/6/2011	4.8	Updated features section to include information about the Lead Source Id parameter.
3/5/2012	4.9	Updated spec/schema to add support for LeadIdToken parameter.
4/23/2012	4.10	Updated schema and added description for "NotSoldExclude" distribution directives.
8/2/2012	4.11	Added AWL/LeadiD Getting Started Guide information to enhance LeadIdToken parameter information.
7/15/2013	4.12	Added LeadMetaData node to the schema which has the following elements in it: IpAddress, LeadBornOnDateTimeUtc, and UserAgent.
8/1/2013	4.13	Updated the Insurance Carrier lists.
8/5/2013	4.14	Schema was updated to add new optional parameters. Parameters include: marital status for health, home, renters, and life lead types, gender for home and renters lead types, first name and last name for health lead dependents, and "is military" indicator for health lead type.
3/13/2014	4.15	Lead samples were added under the Lead Types section.

5/27/2014	4.16	Added full list of possible API ping/post responses.
8/26/2014	4.17	Added optional "Motorcycle" element to Vehicle node in AutoInsuranceRequest.
11/12/2014	4.18	Added sample raw ping/post request/responses for Auto and Health leads.
1/22/2015	4.19	Updated lead schema. Removed duplicate Marital element in Health, LTC, and Disability types. Added QualifyingLifeEvent element in Health type.
11/24/2015	4.20	Updated the Current Provider Carrier List in order to remove carrier strings containing ampersands. The ampersands have been changed to "and" instead.
4/25/2016	4.21	Updated the Undersold Carrier List and added several new carriers
5/11/2017	4.22	Added support for Inbound Warm Transfer Call Affiliates. Updated LeadXMLSchema.xsd and PricePresentationResult.xsd.
1/31/2018	4.23	Added language about omitting the ContactInfo node in the ping XML and omitting the ProductionEnvironment element from ping and post XML.
2/22/2018	4.24	Updated LeadXMLSchema.xsd
3/29/2019	4.25	Added Nonbinary gender option to GenderType.
9/26/2019	4.26	Updated documentation about the usage of the ProductionEnvironment flag.

Introduction

Back To Contents

This document details the information necessary for Data Affiliate partners to begin selling leads to All Web Leads, Inc. (referred to hereafter as AWL) via the AWL Version 4.0 Data Affiliate web service API.

Data Affiliates are defined as those business partners that offer complete Lead Data for purchase to AWL by means of a real time web service provided by AWL.

The AWL Version 4.0 API replaces the previous Version 3.0 API. All partners are encouraged to integrate with the Version 4.0 API, as the 3.0 API will slowly be phased out and discontinued early in 2012.

The Version 4.0 API is intended to be an evolutionary step from the non-SOAP capabilities of the 3.0 API. The primary difference between the V4.0 API and the V3.0 API is that the SOAP-based web services are no longer support, and the V4.0 API relies on entirely new Lead definitions (new lead XML Schema, included along with this specification document).

Supporting XML Schema Files

Back To Contents

- AWL Insurance Lead Type XML Schema
- Send Lead Exclusive API Response XML Schema
- Price Presentation Ping/Post Lead API Response XML Schema
- AWL Insurance Carrier List - Supported Undersold Carriers
- AWL Insurance Carrier List - Supported Consumer Carriers

Supported Features

Back To Contents

Lead Types

Back To Contents

Currently, the Version 4.0 API only supports the following insurance lead types:

(Click the link to see a sample lead for the given lead type)

- [Auto](#)
- [Life](#)
- [Health](#)
- [Homeowners](#)
- [Renters](#)
- [Annuity](#)
- [Business Property/Casualty](#)
- [Business Benefits](#)

Lead Subtypes

In addition to the above lead types, a more detailed categorization, or "Lead Subtype", is also supported. Lead Subtypes are commonly used to define payout and daily lead limit information associated with each Data Affiliate account. Note that not all Lead Types have associated Lead Subtypes. Lead Subtypes are not explicitly represented in the AWL Lead XML Schema, but are instead evaluated at runtime as business rules.

Auto

- Premium Auto
  - Require 1 Driver 22 or Older
  - Require all drivers currently insured
  - Require all drivers have valid licenses /no license suspension (meaning ACTIVE and TEMPORARY is OK)
  - Require NO SR-22
  - Require NO DUI/DWI
  - Require One or Less Tickets
  - Require NO Major Violations (meaning no DUI, and no careless driving)
  - Require No At Fault Accidents
    - Note that if the user selects that an accident is not at-fault, but then selects "Your Vehicle Hit Other Vehicle", we treat that accident as an at-fault accident
- Standard Auto
  - Require 1 Driver 18 or Older
  - Require all drivers currently insured
  - Require all drivers have valid licenses/no license suspension
  - Require NO SR-22
  - Require NO DUI/DWI
  - Require Three or Less Tickets
  - Require One or Less At Fault Accidents
- Substandard Auto
  - Require 1 Driver 18 or Older
  - Anything not meeting Premium or Standard risk
- Juvenile Auto
  - All drivers under age 18

Life

- Premium Life
  - Applicant age 35 - 60
- Standard Life
  - Applicant age 25 - 34
- Substandard Life
  - Others

Health

- Standard Health
  - 1 applicant age 64 or younger
  - No Pregnancy
  - BMI Less than or equal to 35
  - No Major health conditions
  - No Diabetes
  - No Kidney disease
- Substandard Health
  - 1 applicant age 64 or younger
  - Anything not meeting Standard Health criteria
- Senior Health
  - All applicants age 65 or older

Exclusive and Undersold (Shared) Leads

The Version 4.0 API currently allows AWL to purchase/accept exclusive leads from its Data Affiliate partners, as well as Undersold (shared) leads that provide distribution directives indicating how the lead has already been sold by the offering Data Affiliate partner.

Distribution Directives

The included distribution directives within undersold leads need to meet certain criteria or these leads will be rejected. The value in the CarrierName node must be in our list (AWL Insurance Carrier List - Supported Undersold Carriers). If the name is not in this list, "Independent" should be used for the CarrierName. In this case, a LicenseNumber must be included.

Account Lead Limits

The AWL system allows AWL account managers to specify limits (e.g. daily lead limits) on inbound lead volume. If a partner attempts to send leads into the AWL system and the partner account has already exceeded a configured lead limit cap, the lead will be rejected.

With the AWL price presentation ping-post APIs, lead limits are checked during the ping, and also on a subsequent post. This can result in the AWL system accepting and returning a non-zero price for a ping, but then rejecting the subsequent post if the partner lead limit has been hit after the successful ping but prior to the subsequent post. This should be a rare occurrence, and should only occur temporarily at the partner's lead limit boundary point. The frequency with which this will happen is also highly dependent upon the partner's inbound rate of pings and posts, and the concurrency of those transactions into the AWL system.

Duplicate Detection

The Version 4.0 API will reject leads that are determined to be a "duplicate" within our system. When a lead is determined to be a duplicate, it is rejected with a discrete error code.

Our system identifies duplicates by comparing an incoming lead's primary application contact information with leads already in our system. Specifically, we compare based on email address, daytime phone number, and street address, and only compare against leads of the same lead type (Health, Auto, etc.), and only perform this comparison on leads created within our system within the past 30 days.

Note that the AWL system performs duplicate detection on all incoming leads where personally identifying information is provided. For inbound lead pings where personally identifying information is not provided, it is possible for the AWL system to return a non-zero ping price, but then determine that the lead is a duplicate on a subsequent inbound post of the same lead.

Note that the AWL system makes every effort to accurately differentiate between duplicate leads from the same partner (which are not really duplicates, but rather re-tries), and duplicate leads from different partners, which are true duplicates. For more information, please refer to the section entitled "Idempotence and Duplicate Detection" later in this document.

Lead Source Id

The Version 4.0 API requires partners to pass a "Lead Source Id" in the XML. The "Lead Source Id" field is for partners to pass some sort of string identifier that identifies the source from which the partner received the lead. This field helps the process of source optimization.

The "Lead Source Id" field is a string with a maximum length of 20. Below is an example affiliateInfo node where the "Lead Source Id" is included:

```xml
<AffiliateInfo>
    <Username>User</Username>
    <Password>Pass</Password>
    <TrackingCampaign>MyCampaign</TrackingCampaign>
    <LeadSourceID>12345</LeadSourceID>
</AffiliateInfo>
```

See Lead Format Schema for more details.

LeadId Token

We require that the "LeadIdToken" be included in the Lead XML. This parameter is a string type of maximum length 36. This is a token created by LeadId. com (Jornaya) for lead validation purposes.

For more information, please refer to our LeadiD/All Web Leads Getting Started Guide

Sample AffiliateInfo xml snipplet:

```
<AffiliateInfo>
    <Username>User</Username>
    <Password>Pass</Password>
    <TrackingCampaign>MyCampaign</TrackingCampaign>
    <LeadSourceID>12345</LeadSourceID>
    <LeadIdToken>PassedToken</LeadIdToken>
</AffiliateInfo>
```

TrustedForm Token

Back To Contents

We support an optional parameter called "TrustedFormToken". This parameter is a string of maximum length 40. This is a token created by Active Prospect for validation purposes.

Sample AffiliateInfo xml snippet:

```
<AffiliateInfo>
    <Username>User</Username>
    <Password>Pass</Password>
    <TrackingCampaign>MyCampaign</TrackingCampaign>
    <LeadSourceID>12345</LeadSourceID>
    <TrustedFormToken>PassedToken</TrustedFormToken>
</AffiliateInfo>
```

LeadMetaData

We support a few optional fields under the optional "LeadMetaData" node. These fields are defined below:

LeadBornOnDateTimeUtc

This field represents the DateTime (in UTC) that the consumer entered his or her information to create the lead (not necessarily the time it was created in your system, but when the lead was actually "born").

IpAddress

This field represents the ip address where the lead originated (the consumer's ip address). **If you do not have this data, please omit this field.**

UserAgent

This field represents the user agent string of the browser that was used to originally create the lead (the consumer's user agent string). **If you do not have this data, please omit this field.**

Vetting Questions Passed

This field is only used by Inbound Warm Transfer Call Affiliates. If the consumer has answered all verification questions successfully, set field to true. Otherwise set field to false. If you are not an Inbound Warm Transfer Call Affiliate, please omit this field.

Example

```
<LeadMetaData>
  <LeadBornOnDateTimeUtc>2015-07-15T13:23:34Z</LeadBornOnDateTimeUtc>
  <IpAddress>97.112.54.102</IpAddress>
  <UserAgent>Mozilla/5.0 (Windows NT 6.1; WOW64; rv:22.0) Gecko/20100101 Firefox/22.0</UserAgent>
  <VettingQuestionsPassed>true</VettingQuestionsPassed>
</LeadMetaData>
```

Special Distribution Directives

Description

All Web Leads now supports the concept of a "Not Sold" distribution directive and requires partners to use this in special cases. This special type of distribution directive allows a partner to indicate an exclusion of an insurance provider and/or agent license number without affecting the lead's distribution cap or count.

Example:

```
<DistributionDirectives>
    <Cap>4</Cap>
    <DistributionCount>1</DistributionCount>
    <Excludes>
      <Exclude>
        <CarrierName>Independent</CarrierName>
        <LicenseNumber>4491903731590</LicenseNumber>
      </Exclude>
      <NotSoldExclude>
          <CarrierName>State Farm</CarrierName>
      </NotSoldExclude>
    </Excludes>
  </DistributionDirectives>
```

In this example, the directives indicate that the lead has been sold once to an independent agent and that there are 4 more legs available to sell (All Web Leads supports up to 5 leg sales). The directives also indicate that this lead cannot be sold to any entity with the primary carrier of "State Farm" by the "NotSoldExclude" directive.

Important Notes

- NotSoldExclude directives should not count towards DistributionCount or Cap (meaning no leg has been sold to this entity).
- NotSoldExclude directives indicate that a leg cannot be sold to the specified entity.
- Exclude directives indicate that the leg cannot be sold to a specified entity AND that a leg of the lead has been sold.

NotSoldExclude Example Usage

1. Affiliate attempted to sell a lead to a State Farm agent but that captive agent's CRM rejected the lead. State Farm does not want another attempt to purchase that lead.
2. Affiliate tries to sell the lead to All Web Leads.
3. Using the NotSoldExclude directve, the affiliate indicates that they did not sell the lead to a State Farm captive agent but State Farm does not want another attempt to purchase.

Technical Overview

Back To Contents

Protocol

Back To Contents

The AWL API is available as a simple HTTP POST API, exposed over SSL. This API can be consumed by almost any HTTP client technology.

Security

Back To Contents

Transport Security

Back To Contents

AWL requires that all Data Affiliates utilize industry-standard SSL/TLS over HTTP (HTTPS) as a transport mechanism. HTTPS provides authentication, message confidentiality, and message integrity at the transport level.

Authentication

Back To Contents

All Data Affiliates will be assigned a unique affiliate identifier, an affiliate username, and a password. The identifier is used internally to uniquely identify the partner, while the affiliate username and password must be provided in each API call in order to allow AWL to ensure the source of the incoming lead.

Firewall Restriction

Back To Contents

Access to the AWL Data Affiliates API is firewall-restricted. Data Affiliate partners must provide an IP address or an IP address range in order for AWL to white-list access. Note that both staging and production IP addresses will be required.

Quality of Service

Back To Contents

Reliability and Availability

Back To Contents

AWL makes every effort to ensure that the API is available 24 hours a day, 7 days per week so that services requests from Data Affiliate partners to AWL may be transmitted successfully and accurately.

Realistically, some level of downtime will be required to allow for routine maintenance. A regular maintenance window occurs on Sundays between 12:00am and 2:00am. Any necessary downtime outside of that window will be communicated ahead-of-time to the appropriate parties, and every effort will be made to provide adequate notice.

When the AWL API is unavailable due to routine maintenance, any API calls will experience a standard HTTP response of 503 (Service Unavailable).

For any problems encountered by partners that do not appear to be associated with routine maintenance, the appropriate AWL technical personnel can be contacted via our "operations" email list: operations@allwebleads.com.

Idempotence and Duplicate Detection

Back To Contents

The term idempotence describes the property of operations in computer science which yield the same result after an operation is applied multiple times.

In the case of the AWL Data Affiliates web service, the web service ensures idempotence by providing the same result in the event that a transaction contains duplicate data received from a previously successful transaction.

For instance, it is possible that a partner sends a lead which is successfully stored in the AWL system, but before the partner system receives a success acknowledgement. This can occur if the underlying connection is closed. Ideally, the AWL system would detect the closed connection and therefore understand that it cannot return a success acknowledgement to the partner client, however it is understood that this is not possible in all scenarios. In such a situation, AWL requires the partner to attempt to retry the transaction by posting the exact same lead data a second time. The AWL system will return a success code in this case, because the lead was previously successfully received, so it should return an identical result.

Data Affiliate partners should make every effort to refrain from posting the exact same lead data multiple times; however, we understand that the inherent nature of distributed interoperable systems renders any guarantee of duplicate posts impossible.

Note that the AWL system makes every effort to accurately differentiate between duplicate leads from the same partner (which are not really duplicates, but rather re-tries), and duplicate leads from different partners, which are true duplicates.

Timeout and Retries

Back To Contents

AWL expects Data Affiliate partners to ensure reliability within their systems by utilizing transaction timeouts and a retry mechanism for failed or timed out transactions. In the event that the partner experiences an unacceptable latency during any given transaction with the AWL API, the partner should timeout that transaction, and, after a short delay, attempt to retry the exact same transaction. The AWL system is designed to expect partners to retry failed transactions.

If for any reason a failed (but not timed-out) transaction attempt should not be retried by the partner, the AWL system will return the standard HTTP status code of 503 to indicate that the web service is temporarily unavailable, which will inform the AWL systems that a retry is not appropriate at the current moment.

Performance

Back To Contents

Web Service Latency

Back To Contents

AWL records and closely monitors the latency, in seconds, of every post to its system. AWL makes every attempt to return either a success or failure result in the shortest amount of time possible.

AWL single-lead post latencies average about 3 to 10 seconds. In a worst-case scenario, latencies may be as high as 60 seconds. Obviously, external factors are always at play on the open internet and latency values are expected to fluctuate; however, we realize it is important that average latency remains within a reasonable time range.

AWL realizes that low web service posting latency is critical to ensure timely processing and routing of the leads, and benefits all involved partners (consumer, Data Affiliate, and AWL).

Simultaneous Requests

To ensure the most expedient acceptance and routing of leads, the AWL systems accept leads in a multi-threaded (simultaneous) fashion, either from multiple Data Affiliate partners or from one partner.

The AWL system supports receiving multiple leads in a manner that is independent and isolated for each individual lead post.

If for any reason a failed (but not timed-out) transaction attempt should not be retried, the AWL system will return the standard HTTP status code of 503 to indicate that the web service is temporarily unavailable, which will inform the AWL systems that a retry is not appropriate at the current moment.

Result Code and Acknowledgements

The AWL API supports returning a rich set of result codes to communicate as much information as possible in the event of a failure or rejection. Please see the detailed API specification information later in this document for specific information related to the supported set of result codes.

API Versioning

AWL understands that partners invest time and effort into integrating with the AWL API. For this reason, AWL intends to maintain API backward compatibility as much as possible. From time to time, major version releases may break backwards compatibility with existing API consumers.

When it is necessary to break backwards compatibility with a new API version release, AWL intends to fully support the previous major version of the API for a period of up to 6 months, with the intent of providing adequate time for existing partners to migrate their client integrations.

At this time, AWL supports the previous Version 3.0 API and the new Version 4.0 API.

Web Service API Details

The Version 4.0 API also supports a simple API mechanism consisting of an HTTP POST over HTTPS. This version of the API is supported by any standard HTTP client technology.

Data is passed to the HTTP POST API simply as UTF-8 encoded text, organized into ampersand-delimited key/value pairs representing a query-string-like syntax. The Content-Type in the header of the HTTP POST should be set to "application/x-www-form-urlencoded".

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 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).

The "Send Lead Exclusive" API consists of the following URL:

https://<*environment-base-url*>/SendLeadExclusive

For example, in the AWL staging environment:

https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/SendLeadExclusive

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>

API Response

The Send Lead Exclusive API generally will return a standard HTTP 200 response code to indicate a successful operation. Note, however, that a 200 response does not indicate that AWL has accepted the lead. The Data Affiliate partner must inspect the return value of the API to determine if AWL has elected to accept or reject the lead.

When an HTTP 200 response is returned, the return result consists of a UTF-8 encoded XML string. The XML returned conforms to the "LeadsUploadResult.xsd" XML schema document included along with this specification.

This schema contains a single XML type called "LeadsUploadResultType".

- Result
  - Values:
    - "OK"
      - Indicates that the request was successfully processed
      - Only indicates that the lead was accepted/purchased in the event that a non-zero "Payout" value is returned
    - "ERROR"
      - Indicates that an error occurred during processing
- ErrorType
  - Provided when the "Result" is ERROR to provide additional information
  - Values:
    - "XML Validation"
    - "Data Validation"
    - "Login"
    - "Duplicate"
    - "Internal Server Error"
- ErrorDescription
  - Provided when the "Result" is "ERROR" to provide additional information
  - Values consist of a descriptive human-readable string. A few examples:
    - "Invalid Username/Password"
    - "XML Parse/Validation Failure"
    - "Re-post of identical lead from same partner"
    - "Unknown error occurred processing lead"
    - Etc.
- Payout
  - Provided when the "Result" is "OK", and the AWL system has accepted/purchased the incoming lead
  - Represents the expected/estimated payout associated with the acceptance of an incoming lead by the AWL system
- LeadID
  - Provided when the "Result" is "OK", and the "Payout" is greater than $0.00
  - The Lead ID within the AWL system, useful for diagnostic and tracking purposes
  - This value should be logged/recorded in each partner's system
- DataFieldValidation
  - Provided when the "Result" is "ERROR". Zero or more "DataFieldValidation" nodes may be included indicate which field or fields triggered an error response.
  - DataField
    - Identifies the data field (element) within the Lead XML that triggered a validation error
    - Values:
      - nameFirst
      - nameLast
      - nameMiddle
      - namePrefix
      - nameSuffix
      - addressStreet
      - addressStreet2
      - city
      - state
      - zipCode
      - phoneDay
      - phoneEvening
      - email
      - currentlyInsured
      - insuranceProvider
      - lengthOfCoverage
      - coverageExpiration

- DataFieldErrorType
  - Identifies the type of validation error that occurred
  - Values:
    - Fail
    - Warning
- DataFieldErrorMessage
  - Provides additional information related to the validation error that occurred
  - May be empty when no additional information is available

Price Presentation Ping Lead API

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 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.

The "Price Presentation Ping Lead" API consists of the following URL:

https://<*environment-base-url*>/PricePresentationPingLead

For example, in the AWL staging environment:

https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPingLead

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)
  - The ContactInfo node should be omitted in the ping XML

Example HTTP POST Data Payload

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

XmlString=<raw lead xml>

API Response

The Price Presentation Ping Lead API generally will return a standard HTTP 200 response code to indicate a successful operation. Note, however, that a 200 response does not indicate that AWL has placed a bid on the lead. The Data Affiliate partner must inspect the return value of the API to determine if AWL has indeed placed a bid on the lead.

When an HTTP 200 response is returned, the return result consists of a UTF-8 encoded XML string. The XML returned conforms to the "PricePresentationResult.xsd" XML schema document included along with this specification.

This schema contains a single XML type called "PricePresentationResultType".

- Result
  - Values:
    - "OK"
      - Indicates that the request was successfully processed
      - Only indicates that the lead was bid upon in the event that a non-zero "Payout" and a valid "LeadID" value is returned
    - "ERROR"
      - Indicates that an error occurred during processing
- ErrorType
  - Provided when the "Result" is "ERROR" to provide additional information
  - Values:
    - "XML Validation"

- "Data Validation"
- "Login"
- "Duplicate"
- "Internal Server Error"
- ErrorDescription
  - Provided when the "Result" is "ERROR" to provide additional information
  - Values consist of a descriptive human-readable string. A few examples:
    - "Invalid Username/Password"
    - "XML Parse/Validation Failure"
    - "Re-post of identical lead from same partner"
    - "Unknown error occurred processing lead"
    - Etc.
- Payout
  - Provided when the "Result" is "OK", and the AWL system has successfully placed a bid on the incoming lead
  - Represents the expected/estimated payout associated with a potential subsequent post of the same lead
- LeadID
  - Provided when the "Result" is "OK", and the "Payout" is greater than $0.00
  - The Lead ID value is not a "real" lead identifier within the AWL system, but instead is actually a ping session identifier that must be used in a subsequent "Price Presentation Post Lead" API call if the partner chooses to offer the full lead to AWL for sale at the bid-upon price point
  - This value should be logged/recorded in each partner's system
- DataFieldValidation
  - Provided when the "Result" is "ERROR". Zero or more "DataFieldValidation" nodes may be included indicate which field or fields triggered an error response.
  - DataField
    - Identifies the data field (element) within the Lead XML that triggered a validation error
    - Values:
      - nameFirst
      - nameLast
      - nameMiddle
      - namePrefix
      - nameSuffix
      - addressStreet
      - addressStreet2
      - city
      - state
      - zipCode
      - phoneDay
      - phoneEvening
      - email
      - currentlyInsured
      - insuranceProvider
      - lengthOfCoverage
      - coverageExpiration
  - DataFieldErrorType
    - Identifies the type of validation error that occurred
    - Values:
      - Fail
      - Warning
  - DataFieldErrorMessage
    - Provides additional information related to the validation error that occurred
    - May be empty when no additional information is available

Price Presentation Post Lead API

Back To Contents

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).

The "Price Presentation Post Lead" API consists of the following URL:

https://<*environment-base-url*>/PricePresentationPostLead

For example, in the AWL staging environment:

https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPostLead

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)
  - With the exception of the ContactInfo node and the FirstName/LastName elements (Auto leads only), the XML string must be equivalent to XML provided in previous successful call to the "Price Presentation Ping Lead" API

Example HTTP POST Data Payload

Back To Contents

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

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

API Response

Back To Contents

List of possible responses

The Price Presentation Post Lead API generally will return a standard HTTP 200 response code to indicate a successful operation. Note, however, that a 200 response does not indicate that AWL has successfully purchased the lead. The Data Affiliate partner must inspect the return value of the API to determine if AWL has accepted/purchased the posted lead.

When an HTTP 200 response is returned, the return result consists of a UTF-8 encoded XML string. The XML returned conforms to the "PricePresentationResult.xsd" XML schema document included along with this specification.

This schema contains a single XML type called "PricePresentationResultType".

- Result
  - Values:
    - "OK"
      - Indicates that the request was successfully processed
      - Only indicates that the lead was successfully accepted/purchased in the event that the "Payout" value is non-zero
    - "ERROR"
      - Indicates that an error occurred during processing
- ErrorType
  - Provided when the "Result" is ERROR to provide additional information
  - Values:
    - "XML Validation"
    - "Data Validation"
    - "Login"
    - "Duplicate"
    - "Internal Server Error"
- ErrorDescription
  - Provided when the "Result" is ERROR to provide additional information
  - Values consist of a descriptive human-readable string. A few examples:
    - "Invalid Username/Password"
    - "XML Parse/Validation Failure"
    - "Re-post of identical lead from same partner"
    - "Unknown error occurred processing lead"
    - Etc.
- Payout
  - Provided when the "Result" is "OK", and the AWL system has successfully accepted/purchased incoming lead
  - Represents the expected/estimated payout, and will always match the bid price from the previous ping
- LeadID
  - Provided when the "Result" is "OK", and the "Payout" is greater than $0.00
  - The Lead ID within the AWL system, useful for diagnostic and tracking purposes
  - This value should be logged/recorded in each partner's system
- DataFieldValidation
  - Provided when the "Result" is "ERROR". Zero or more "DataFieldValidation" nodes may be included indicate which field or fields triggered an error response.
  - DataField
    - Identifies the data field (element) within the Lead XML that triggered a validation error
    - Values:

- nameFirst
- nameLast
- nameMiddle
- namePrefix
- nameSuffix
- addressStreet
- addressStreet2
- city
- state
- zipCode
- phoneDay
- phoneEvening
- email
- currentlyInsured
- insuranceProvider
- lengthOfCoverage
- coverageExpiration
- DataFieldErrorType
  - Identifies the type of validation error that occurred
  - Values:
    - Fail
    - Warning
- DataFieldErrorMessage
  - Provides additional information related to the validation error that occurred
  - May be empty when no additional information is available

Price Presentation Ping Inbound Warm Transfer API

Back To Contents

The "Price Presentation Ping Inbound Warm Transfer" API is used to send the data associated with warm transfer to the AWL system to receive a bid on the warm transfer (this is often referred to as "price presentation ping-post" in the industry).

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

The warm transfer data must be provided in its entirety, although all personally identifiable information for the primary applicant is considered optional. The AWL system evaluates the provided warm transfer 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 warm transfer.

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 Inbound Warm Transfer" API call to offer the warm transfer for sale at the previously bid price point. Note that a warm transfer bid price and its ping session identifier is valid only for up to five minutes.

The "Price Presentation Ping Inbound Warm Transfer" API consists of the following URL:

https://<*environment-base-url*>/PricePresentationPingLead

For example, in the AWL staging environment:

https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPingLead

A single post parameter is supported, which allows the passing of warm transfer 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

Back To Contents

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

XmlString=<raw lead xml>

API Response

Back To Contents

List of possible responses

The Price Presentation Ping Inbound Warm Transfer API generally will return a standard HTTP 200 response code to indicate a successful operation. Note, however, that a 200 response does not indicate that AWL has placed a bid on the warm transfer. The Data Affiliate partner must inspect the return value of the API to determine if AWL has indeed placed a bid on the warm transfer.

When an HTTP 200 response is returned, the return result consists of a UTF-8 encoded XML string. The XML returned conforms to the "PricePresentationResult.xsd" XML schema document included along with this specification.

This schema contains a single XML type called "PricePresentationResultType".

- Result
    - Values:
        - "OK"
            - Indicates that the request was successfully processed
            - Only indicates that the lead was bid upon in the event that a non-zero "Payout" and a valid "LeadID" value is returned
        - "ERROR"
            - Indicates that an error occurred during processing
- ErrorType
    - Provided when the "Result" is "ERROR" to provide additional information
    - Values:
        - "XML Validation"
        - "Data Validation"
        - "Login"
        - "Duplicate"
        - "Internal Server Error"
- ErrorDescription
    - Provided when the "Result" is "ERROR" to provide additional information
    - Values consist of a descriptive human-readable string. A few examples:
        - "Invalid Username/Password"
        - "XML Parse/Validation Failure"
        - "Re-post of identical lead from same partner"
        - "Unknown error occurred processing lead"
        - Etc.
- Payout
    - Provided when the "Result" is "OK", and the AWL system has successfully placed a bid on the incoming warm transfer
    - Represents the expected/estimated payout associated with a potential subsequent post of the same warm transfer
- LeadID
    - Provided when the "Result" is "OK", and the "Payout" is greater than $0.00
    - The Lead ID value is not a "real" warm transfer identifier within the AWL system, but instead is actually a ping session identifier that must be used in a subsequent "Price Presentation Post Inbound Warm Transfer" API call if the partner chooses to offer the full warm transfer data to AWL for sale at the bid-upon price point
    - This value should be logged/recorded in each partner's system
- DataFieldValidation
    - Provided when the "Result" is "ERROR". Zero or more "DataFieldValidation" nodes may be included indicate which field or fields triggered an error response.
    - DataField
        - Identifies the data field (element) within the Lead XML that triggered a validation error
        - Values:
            - nameFirst
            - nameLast
            - nameMiddle
            - namePrefix
            - nameSuffix
            - addressStreet
            - addressStreet2
            - city
            - state
            - zipCode
            - phoneDay
            - phoneEvening
            - email
            - currentlyInsured
            - insuranceProvider
            - lengthOfCoverage
            - coverageExpiration
    - DataFieldErrorType
        - Identifies the type of validation error that occurred
        - Values:
            - Fail
            - Warning
    - DataFieldErrorMessage
        - Provides additional information related to the validation error that occurred
        - May be empty when no additional information is available
- AllowTransfer
    - Boolean field that indicates whether or not the warm transfer has been accepted

Price Presentation Post Inbound Warm Transfer API

The "Price Presentation Post Inbound Warm Transfer" API is used to offer warm transfers to sell after those warm transfers have successfully been bid upon by the AWL system via a previous call to the "Price Presentation Ping Inbound Warm Transfer" API.

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

The warm transfer 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 Inbound Warm Transfer" API. This "LeadID" value represents a session identifier that associates the post with the ping. Note that a warm transfer 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 warm transfer data via this API, the personal identifying information will be validated. If the identifying information cannot be validated, the warm transfer will rejected. Additionally, the personal identifying information will be used to determine if the warm transfer data is considered a duplicate within the AWL system. If the warm transfer data is a duplicate, it will be rejected (please refer to the "Duplicate Detection" section earlier in this document for additional details).

The "Price Presentation Post Inbound Warm Transfer" API consists of the following URL:

https://<*environment-base-url*>/PricePresentationPostLead

For example, in the AWL staging environment:

https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPostLead

A single post parameter is supported, which allows the passing of warm transfer 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 Inbound Warm Transfer" 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)
    - With the exception of the ContactInfo node and the FirstName/LastName elements (Auto leads only), the XML string must be equivalent to XML provided in previous successful call to the "Price Presentation Ping Inbound Warm Transfer" 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>

API Response

The Price Presentation Post Inbound Warm Transfer API generally will return a standard HTTP 200 response code to indicate a successful operation. Note, however, that a 200 response does not indicate that AWL has successfully purchased the warm transfer. The Data Affiliate partner must inspect the return value of the API to determine if AWL has accepted/purchased the posted warm transfer.

When an HTTP 200 response is returned, the return result consists of a UTF-8 encoded XML string. The XML returned conforms to the "PricePresentationResult.xsd" XML schema document included along with this specification.

This schema contains a single XML type called "PricePresentationResultType".

- Result
    - Values:
        - "OK"
            - Indicates that the request was successfully processed
            - Only indicates that the warm transfer was successfully accepted/purchased in the event that the "Payout" value is non-zero
        - "ERROR"
            - Indicates that an error occurred during processing
- ErrorType
    - Provided when the "Result" is "ERROR" to provide additional information
    - Values:
        - "XML Validation"
        - "Data Validation"
        - "Login"
        - "Duplicate"
        - "Internal Server Error"
- ErrorDescription

- Provided when the "Result" is "ERROR" to provide additional information
- Values consist of a descriptive human-readable string. A few examples:
  - "Invalid Username/Password"
  - "XML Parse/Validation Failure"
  - "Re-post of identical lead from same partner"
  - "Unknown error occurred processing lead"
  - Etc.
- Payout
  - Provided when the "Result" is "OK", and the AWL system has successfully accepted/purchased incoming warm transfer
  - Represents the expected/estimated payout, and will always match the bid price from the previous ping
- LeadID
  - Provided when the "Result" is "OK", and the "Payout" is greater than $0.00
  - The Lead ID within the AWL system, useful for diagnostic and tracking purposes
  - This value should be logged/recorded in each partner's system
- DataFieldValidation
  - Provided when the "Result" is "ERROR". Zero or more "DataFieldValidation" nodes may be included indicate which field or fields triggered an error response.
  - DataField
    - Identifies the data field (element) within the Lead XML that triggered a validation error
    - Values:
      - nameFirst
      - nameLast
      - nameMiddle
      - namePrefix
      - nameSuffix
      - addressStreet
      - addressStreet2
      - city
      - state
      - zipCode
      - phoneDay
      - phoneEvening
      - email
      - currentlyInsured
      - insuranceProvider
      - lengthOfCoverage
      - coverageExpiration
  - DataFieldErrorType
    - Identifies the type of validation error that occurred
    - Values:
      - Fail
      - Warning
  - DataFieldErrorMessage
    - Provides additional information related to the validation error that occurred
    - May be empty when no additional information is available

Legs Ping Lead API

Back To Contents

The "Legs Ping Lead" API is used to send leads to the AWL system to receive a bid on certain legs that we are interested in purchasing. Note that, within the context of shared leads, we refer to each individual potential shared sale of a lead as a **leg.**

The ping process is identical to that of the "Price Presentation Ping Lead" API except the endpoint is different and the response has additional data added to it. In a successful response, the ping response will include the legs (up to 5) that AWL is interested in purchasing. Each leg returned in the ping will have an associated payout with it.

The partner posting can then determine which legs that they want to sell to AWL and they can indicate this in the following Legs Post (see Legs Post Lead API description). When legs with associated payouts are returned, a "LeadID" value is returned which represents a ping session identifier. The ping session identifier may be used in a follow-up "Legs Post Lead" API call to offer selected legs for sale at the previous bid price points. Note that leg bid prices and their ping session identifier are valid only for up to two minutes.

The "Legs Ping Lead" API consists of the following URL:

https://<*environment-base-url*>/LegsPingLead

For example, in the AWL staging environment:

https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPingLead

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)
  - The ContactInfo node should be omitted in the ping XML

Relevant schemas:

Sample data:

Example HTTP POST Data Payload

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

XmlString=<raw lead xml>

API Response

The Legs Ping Lead API generally will return a standard HTTP 200 response code to indicate a successful operation. Note, however, that a 200 response does not indicate that AWL has placed bids on legs. The Data Affiliate partner must inspect the return value of the API to determine if AWL has indeed placed bids on legs.

When an HTTP 200 response is returned, the return result consists of a UTF-8 encoded XML string. The XML returned conforms to the "PricePresentationResult.xsd" XML schema document included along with this specification.

This schema contains a single XML type called "PricePresentationResultType".

- Result
    - Values:
        - "OK"
            - Indicates that the request was successfully processed
            - Only indicates that the lead was bid upon in the event that a non-zero "Payout" and a valid "LeadID" value is returned
        - "ERROR"
            - Indicates that an error occurred during processing
- ErrorType
    - Provided when the "Result" is "ERROR" to provide additional information
    - Values:
        - "XML Validation"
        - "Data Validation"
        - "Login"
        - "Duplicate"
        - "Internal Server Error"
- ErrorDescription
    - Provided when the "Result" is "ERROR" to provide additional information
    - Values consist of a descriptive human-readable string. A few examples:
        - "Invalid Username/Password"
        - "XML Parse/Validation Failure"
        - "Re-post of identical lead from same partner"
        - "Unknown error occurred processing lead"
        - Etc.
- Payout
    - Provided when the "Result" is "OK", and the AWL system has successfully placed a bid on a leg(s).
    - Represents the expected/estimated payout associated with a potential subsequent post of the same lead. This payout price is a sum of all the leg payouts and is what the payout would be if the partner chose to sell all the legs that were presented in the ping response.
- LeadID
    - Provided when the "Result" is "OK", and the "Payout" is greater than $0.00 (meaning there are legs we want to purchase).
    - The Lead ID value is not a "real" lead identifier within the AWL system, but instead is actually a ping session identifier that must be used in a subsequent "Legs Post Lead" API call if the partner chooses to offer legs at their corresponding payout prices.
    - This value should be logged/recorded in each partner's system
- DataFieldValidation
    - Provided when the "Result" is "ERROR". Zero or more "DataFieldValidation" nodes may be included indicate which field or fields triggered an error response.
    - DataField
        - Identifies the data field (element) within the Lead XML that triggered a validation error
        - Values:
            - nameFirst
            - nameLast
            - nameMiddle
            - namePrefix
            - nameSuffix
            - addressStreet
            - addressStreet2
            - city
            - state

- zipCode
- phoneDay
- phoneEvening
- email
- currentlyInsured
- insuranceProvider
- lengthOfCoverage
- coverageExpiration
- DataFieldErrorType
  - Identifies the type of validation error that occurred
  - Values:
    - Fail
    - Warning
- DataFieldErrorMessage
  - Provides additional information related to the validation error that occurred
  - May be empty when no additional information is available
- Legs
  - Represents the beginning of a collection of legs.
  - Leg
    - Represents an individual leg that we are wishing to purchase.
    - CarrierName
      - The carrier name of a captive agent that corresponds to a leg we are wishing to purchase. If a CarrierName is present, no LiceseNumber will be present.
      - If this leg is given up (sold to AWL), then we will only route this leg to an agent that is captive with that CarrierName and we expect that the partner selling the leg will not sell the lead to a captive agent with the same carrier.
    - Payout
      - The price AWL will payout for a given leg.
    - LegID
      - A unique leg identifier that it is used in the corresponding post to indicate which leg(s) that is being sold.
    - LegGUID
      - Another unique identifier that is used in the corresponding post to indicate which leg(s) that is being sold.
    - LicenseNumber
      - The license number of an agent that corresponds to a leg we are wishing to purchase. If a license number is present, the associated CarrierName will be "Independent".
      - If this leg is given up (sold to AWL), then we will only route this leg to the indicated license number and we expect that the partner selling the leg will not sell the lead to an agent with the same license number.

Sample Leg Ping response:

```
<?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>31</LeadID>
    <Payout>25.36</Payout>
    <Legs>
        <Leg>
            <LegID>9</LegID>
            <LegGUID>1ec301b331474c55bbb7517ee7055d5d</LegGUID>
            <Payout>6.96</Payout>
            <CarrierName>Independent</CarrierName>
            <LicenseNumber>173007</LicenseNumber>
        </Leg>
        <Leg>
            <LegID>10</LegID>
            <LegGUID>a216dd235c274b56a50c573028ace3d7</LegGUID>
            <Payout>5.2</Payout>
            <CarrierName>Health Choice One</CarrierName>
        </Leg>
        <Leg>
            <LegID>11</LegID>
            <LegGUID>744587c381f247a4b6151cffd627d96c</LegGUID>
            <Payout>4.8</Payout>
            <CarrierName>American Insurance</CarrierName>
        </Leg>
        <Leg>
            <LegID>12</LegID>
            <LegGUID>389f84c26ee44183a1b4edfd4eb3adb2</LegGUID>
            <Payout>4.8</Payout>
            <CarrierName>US Health Group</CarrierName>
        </Leg>
        <Leg>
            <LegID>13</LegID>
            <LegGUID>34880766679b4a6d8deddee9c51b3110</LegGUID>
            <Payout>3.6</Payout>
            <CarrierName>Independent</CarrierName>
            <LicenseNumber>162232</LicenseNumber>
        </Leg>
    </Legs>
</PricePresentationResult>
```

Legs Post Lead API

[Back To Contents](#)

The "Legs Post Lead" API is used to offer legs for a given lead for sale after those legs have successfully been bid upon by the AWL system via a previous call to the "Legs Ping Lead" API.

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 "Legs Ping Lead" API. This "LeadID" value represents a session identifier that associates the post with the ping. Note that a lead and its associated leg prices and its ping session identifier are valid only for up to two minutes. If a call to this API uses a ping session identifier that is more than two (2) minutes old, that call will be rejected.

The post Lead Xml will also need to indicate leg offers that have been accepted. This is done by including in the lead Post Xml the LegID and LegGUIDs of the accepted leg offers that were returned in the previous call to the "Legs Ping Lead" API.

With the exception of the ping-time optional personally identifying information and accepted leg offers indicated in the Lead post Xml, 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).

The "Legs Post Lead" API consists of the following URL:

https://<*environment-base-url*>/LegsPostLead

For example, in the AWL staging environment:

Two post parameters are supported, which allow the passing of lead data in the AWL lead XML format and indicating what the corresponding lead is:

- **LeadID** (String)
    - A string containing the "LeadID" value returned by a previous successful call to the "Legs Ping Lead" API that uniquely identifies the ping session and is used to associate this API call with the leg price bids 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)
    - With the exception of the ContactInfo node and the FirstName/LastName elements (Auto leads only), the XML string must be equivalent to XML provided in previous successful call to the "Legs Ping Lead" API.

Relevant schemas:

Lead Format Schema
API Response Schema

Sample data:

Sample payload data and responses

Example HTTP POST Data Payload

Back To Contents

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

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

API Response

Back To Contents

List of possible responses

The Legs Post Lead API generally will return a standard HTTP 200 response code to indicate a successful operation. Note, however, that a 200 response does not indicate that AWL has successfully purchased the lead and its associated legs. The Data Affiliate partner must inspect the return value of the API to determine if AWL has accepted/purchased the posted lead and its associated legs.

When an HTTP 200 response is returned, the return result consists of a UTF-8 encoded XML string. The XML returned conforms to the "PricePresentationResult.xsd" XML schema document included along with this specification.

This schema contains a single XML type called "PricePresentationResultType".

- Result
    - Values:
        - "OK"
            - Indicates that the request was successfully processed
            - Only indicates that the lead was successfully accepted/purchased in the event that the "Payout" value is non-zero
        - "ERROR"
            - Indicates that an error occurred during processing
- ErrorType
    - Provided when the "Result" is "ERROR" to provide additional information
    - Values:
        - "XML Validation"
        - "Data Validation"
        - "Login"
        - "Duplicate"
        - "Internal Server Error"
- ErrorDescription
    - Provided when the "Result" is "ERROR" to provide additional information
    - Values consist of a descriptive human-readable string. A few examples:
        - "Invalid Username/Password"
        - "XML Parse/Validation Failure"
        - "Re-post of identical lead from same partner"
        - "Unknown error occurred processing lead"
        - Etc.
- Payout
    - Provided when the "Result" is "OK", and the AWL system has successfully accepted/purchased incoming lead and the offered legs
    - Represents the expected/estimated payout of all the legs that have been sold to AWL
- LeadID
    - Provided when the "Result" is "OK", and the "Payout" is greater than $0.00
    - The Lead ID within the AWL system, useful for diagnostic and tracking purposes
    - This value should be logged/recorded in each partner's system
- DataFieldValidation

- Provided when the "Result" is "ERROR". Zero or more "DataFieldValidation" nodes may be included indicate which field or fields triggered an error response.
- DataField
    - Identifies the data field (element) within the Lead XML that triggered a validation error
    - Values:
        - nameFirst
        - nameLast
        - nameMiddle
        - namePrefix
        - nameSuffix
        - addressStreet
        - addressStreet2
        - city
        - state
        - zipCode
        - phoneDay
        - phoneEvening
        - email
        - currentlyInsured
        - insuranceProvider
        - lengthOfCoverage
        - coverageExpiration
- DataFieldErrorType
    - Identifies the type of validation error that occurred
    - Values:
        - Fail
        - Warning
- DataFieldErrorMessage
    - Provides additional information related to the validation error that occurred
    - May be empty when no additional information is available

Sample response:

<?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>18487630</LeadID><Payout>10.36</Payout></PricePresentationResult>

Staging Environment

AWL provides a dedicated staging environment that partners may use to begin testing their integration with the Data Affiliate API.

The staging environment is completely isolated from our production environment, and therefore there are no restrictions on the lead data that can be posted into the system.

Note, however, that our duplicate detection rules (detailed earlier in the section "Duplicate Detection"), continue to apply within the staging environment. This means that each test lead that is successfully posted into the staging environment should have a unique email address, daytime phone number, and street address.

Note that the AWL staging environment currently resides on hardware infrastructure that is not as powerful as that used by the AWL production environment. Additionally, some tasks performed in the staging environment are "faked", and therefore will result in different latencies in staging versus production. Essentially, performance within the staging environment should not be considered indicative of the production environment.

If latencies are a concern, additional testing may be performed against the production environment using a restricted set of lead data to ensure that test leads are not inadvertently mixed with live production leads. Note that even when sending test leads into the production environment, certain internal operations can vary with respect to the operations performed on a real live lead. For more information on testing within the production environment, please refer to the "Production Environment" section later in this document.

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

Staging Environment Test Application

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

Staging Environment URLs

The following URLs should be used for the HTTPS POST into the AWL system.

Note that these web service endpoints do not support HTTPS GET, and therefore will provide no useful output if entered into a web browser. Please use the staging environment test application mentioned earlier for attempting to experiment with the system ahead of your own development.

Send Lead Exclusive API:
https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/SendLeadExclusive

Price Presentation Ping API:
https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPingLead

Price Presentation Post API:
https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPostLead

Price Presentation Legs Ping API:
https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPingLead

Price Presentation Legs Post API:
https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPostLead

Production Environment

The AWL production environment is used to receive live leads from our Data Affiliate partners.

Production Environment URLs

The following URLs should be used for the HTTPS POST into the AWL system.

Note that these web service endpoints do not support HTTPS GET, and therefore will provide no useful output if entered into a web browser. Please use the staging environment test application mentioned earlier for attempting to experiment with the system ahead of your own development.

Send Lead Exclusive API:
https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/SendLeadExclusive

Price Presentation Ping API:
https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPingLead

Price Presentation Post API:
https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPostLead

Price Presentation Legs Ping API:
https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPingLead

Price Presentation Legs Post API:
https://ws.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/LegsPostLead

Production Environment Flag

All testing should be conducted within the AWL dedicated staging environment. The ProductionEnvironment flag should be completely omitted from the Lead XML in both the staging and production.

Quick Start Guide

If you would like to see a summarized version to get started, please check out the Quick Start Guide.

Sample Ping/Post Raw Request/Response

Auto Ping Raw Request

```
POST https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPingLead HTTP
/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 4.0.30319.1)
Host: dastaging-lm.dev.allwebleads.com
Content-Length: 3497
Expect: 100-continue

XmlString=<?xml version="1.0" encoding="utf-8"?><InsuranceRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><AffiliateInfo><Username>12696<
/Username><Password>MyPassword</Password><TrackingCampaign>MyTrackingCampaign<
/TrackingCampaign><LeadSourceID>SourceID</LeadSourceID><LeadIdToken>8731d15b-793c-4ea0-b7e2-35b35a66f063<
/LeadIdToken></AffiliateInfo><IsMobile>false</IsMobile><LeadMetaData><LeadBornOnDateTimeUtc>2014-11-11T03:31:
21Z</LeadBornOnDateTimeUtc><IpAddress>204.205.123.125</IpAddress><UserAgent>UserAgentTextGoesHere</UserAgent><
/LeadMetaData><ZipCode>34747</ZipCode><AutoInsurance><Vehicles><Vehicle><Id>1</Id><Vin>*****************<
/Vin><Year>2008</Year><Make>VOLVO</Make><Model>S60 2.5T AWD</Model><SubModel>_</SubModel><Owner>Leased<
/Owner><Garage>No Cover</Garage><PrimaryUse>Commute Work</PrimaryUse><OneWay>90</OneWay><AnnualMileage>3000<
/AnnualMileage><Salvaged>true</Salvaged><Comprehensive>No Coverage</Comprehensive><Collision>50<
/Collision><Motorcycle>false</Motorcycle></Vehicle><Vehicle><Id>2</Id><Vin>*****************</Vin><Year>2008<
/Year><Make>VOLVO</Make><Model>S60 2.5T AWD</Model><SubModel>_</SubModel><Owner>Owned</Owner><Garage>No Cover<
/Garage><PrimaryUse>Business</PrimaryUse><OneWay>29</OneWay><AnnualMileage>5001</AnnualMileage><Salvaged>true<
/Salvaged><Comprehensive>250</Comprehensive><Collision>500</Collision><Motorcycle>false</Motorcycle></Vehicle><
/Vehicles><Drivers><Driver><Id>1</Id><FirstName>Scarlett</FirstName><LastName>Johnson</LastName><Gender>Male<
/Gender><Marital>Separated</Marital><DOB>1953-07-15</DOB><Relation>Self</Relation><DLicenseState>TX<
/DLicenseState><DLicenseStatus>Active</DLicenseStatus><LicensedAge>18</LicensedAge><VehicleId>1<
/VehicleId><Filing>false</Filing><Education>High School Diploma</Education><Occupation>Pharmaceutical/Biotech<
/Occupation><GoodStudent>true</GoodStudent><Suspension>false</Suspension></Driver><Driver><Id>2<
/Id><FirstName>Grace</FirstName><LastName>Johnson</LastName><Gender>Male</Gender><Marital>Single<
/Marital><DOB>1943-11-24</DOB><Relation>Parent</Relation><DLicenseState>TX<
/DLicenseState><DLicenseStatus>Active</DLicenseStatus><LicensedAge>17</LicensedAge><VehicleId>1<
/VehicleId><Filing>false</Filing><Education>Other</Education><Occupation>Advertising/Public Relations<
/Occupation><GoodStudent>false</GoodStudent><Suspension>false</Suspension></Driver><
/Drivers><Incidents><Ticket><DriverId>1</DriverId><Date>2012-07-01</Date><Description>Speeding</Description><
/Ticket><Ticket><DriverId>2</DriverId><Date>2011-11-01</Date><Description>Speeding</Description><
/Ticket><Ticket><DriverId>1</DriverId><Date>2011-12-01</Date><Description>Other Ticket</Description><
/Ticket><MajorViolation><DriverId>2</DriverId><Date>2012-03-01</Date><Description>Drunk Driving - no Injury<
/Description></MajorViolation><Accident><DriverId>2</DriverId><Date>2011-11-01</Date><Description>Chargeable
Accident - Injury</Description><PaidAmount>63677</PaidAmount></Accident><Claim><DriverId>1</DriverId><Date>2013-
07-01</Date><Description>Fire Hail Water Damage</Description><PaidAmount>25125</PaidAmount></Claim><
/Incidents><ApplicantInfo><Credit>Good</Credit><Residence>Own</Residence><
/ApplicantInfo><CurrentInsurance><CurrentlyInsured>false</CurrentlyInsured><CoverageType>Standard<
/CoverageType></CurrentInsurance><RequestedCoverage>Standard</RequestedCoverage></AutoInsurance><
/InsuranceRequest>
```

Auto Ping Raw Response

```
HTTP/1.1 200 OK
Content-Length: 329
Content-Type: text/plain
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Wed, 12 Nov 2014 18:06:25 GMT

<?xml version="1.0" encoding="utf-8"?><PricePresentationResult xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.insuranceleads.com/partner
/PricePresentationResult.xsd"><Result>OK</Result><LeadID>677739597</LeadID><Payout>4.53</Payout><
/PricePresentationResult>
```

Auto Post Raw Request

```
POST https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPostLead HTTP
/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 4.0.30319.1)
Host: dastaging-lm.dev.allwebleads.com
Content-Length: 3878
Expect: 100-continue

LeadID=677739597&XmlString=<?xml version="1.0" encoding="utf-8"?><InsuranceRequest xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><AffiliateInfo><Username>12696<
/Username><Password>MyPassword</Password><TrackingCampaign>MyTrackingCampaign<
/TrackingCampaign><LeadSourceID>SourceID</LeadSourceID><LeadIdToken>8731d15b-793c-4ea0-b7e2-35b35a66f063<
/LeadIdToken></AffiliateInfo><IsMobile>false</IsMobile><LeadMetaData><LeadBornOnDateTimeUtc>2014-11-11T03:31:
21Z</LeadBornOnDateTimeUtc><IpAddress>204.205.123.125</IpAddress><UserAgent>UserAgentTextGoesHere<
/LeadMetaData><ZipCode>34747</ZipCode><ContactInfo><FirstName>Roger</FirstName><LastName>Garcia<
/LastName><Address>42078 Main St</Address><ZipCode>34747</ZipCode><City>KISSIMMEE</City><County>Osceola<
/County><State>FL</State><PhoneDay>2547458235</PhoneDay><PhoneEve>2546493044</PhoneEve><PhoneCell>2547458235<
/PhoneCell><Email>RogerGarcia1528621581@test.org</Email><Comment>None</Comment><
/ContactInfo><AutoInsurance><Vehicles><Vehicle><Id>1</Id><Vin>*****************</Vin><Year>2008<
/Year><Make>VOLVO</Make><Model>S60 2.5T AWD</Model><SubModel>_</SubModel><Owner>Leased</Owner><Garage>No Cover<
/Garage><PrimaryUse>Commute Work</PrimaryUse><OneWay>90</OneWay><AnnualMileage>3000<
/AnnualMileage><Salvaged>true</Salvaged><Comprehensive>No Coverage</Comprehensive><Collision>50<
/Collision><Motorcycle>false</Motorcycle></Vehicle><Vehicle><Id>2</Id><Vin>*****************</Vin><Year>2008<
/Year><Make>VOLVO</Make><Model>S60 2.5T AWD</Model><SubModel>_</SubModel><Owner>Owned</Owner><Garage>No Cover<
/Garage><PrimaryUse>Business</PrimaryUse><OneWay>29</OneWay><AnnualMileage>5001</AnnualMileage><Salvaged>true<
/Salvaged><Comprehensive>250</Comprehensive><Collision>500</Collision><Motorcycle>false</Motorcycle></Vehicle><
/Vehicles><Drivers><Driver><Id>1</Id><FirstName>Scarlett</FirstName><LastName>Johnson</LastName><Gender>Male<
/Gender><Marital>Separated</Marital><DOB>1953-07-15</DOB><Relation>Self</Relation><DLicenseState>TX<
/DLicenseState><DLicenseStatus>Active</DLicenseStatus><LicensedAge>18</LicensedAge><VehicleId>1<
/VehicleId><Filing>false</Filing><Education>High School Diploma</Education><Occupation>Pharmaceutical/Biotech<
/Occupation><GoodStudent>true</GoodStudent><Suspension>false</Suspension></Driver><Driver><Id>2<
/Id><FirstName>Grace</FirstName><LastName>Johnson</LastName><Gender>Male</Gender><Marital>Single<
/Marital><DOB>1943-11-24</DOB><Relation>Parent</Relation><DLicenseState>TX<
/DLicenseState><DLicenseStatus>Active</DLicenseStatus><LicensedAge>17</LicensedAge><VehicleId>1<
/VehicleId><Filing>false</Filing><Education>Other</Education><Occupation>Advertising/Public Relations<
/Occupation><GoodStudent>false</GoodStudent><Suspension>false</Suspension></Driver><
/Drivers><Incidents><Ticket><DriverId>1</DriverId><Date>2012-07-01</Date><Description>Speeding</Description><
/Ticket><Ticket><DriverId>2</DriverId><Date>2011-11-01</Date><Description>Speeding</Description><
/Ticket><Ticket><DriverId>1</DriverId><Date>2011-12-01</Date><Description>Other Ticket</Description><
/Ticket><MajorViolation><DriverId>2</DriverId><Date>2012-03-01</Date><Description>Drunk Driving - no Injury<
/Description></MajorViolation><Accident><DriverId>2</DriverId><Date>2011-11-01</Date><Description>Chargeable
Accident - Injury</Description><PaidAmount>63677</PaidAmount></Accident><Claim><DriverId>1</DriverId><Date>2013-
07-01</Date><Description>Fire Hail Water Damage</Description><PaidAmount>25125</PaidAmount></Claim><
/Incidents><ApplicantInfo><Credit>Good</Credit><Residence>Own</Residence><
/ApplicantInfo><CurrentInsurance><CurrentlyInsured>false</CurrentlyInsured><CoverageType>Standard<
/CoverageType></CurrentInsurance><RequestedCoverage>Standard</RequestedCoverage></AutoInsurance><
/InsuranceRequest>
```

Auto Post Raw Response

```
HTTP/1.1 200 OK
Content-Length: 328
Content-Type: text/plain
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Wed, 12 Nov 2014 18:06:25 GMT

<?xml version="1.0" encoding="utf-8"?><PricePresentationResult xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.insuranceleads.com/partner
/PricePresentationResult.xsd"><Result>OK</Result><LeadID>52694908</LeadID><Payout>4.53</Payout><
/PricePresentationResult>
```

Health Ping Raw Request

```
POST https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPingLead HTTP
/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 4.0.30319.1)
Host: dastaging-lm.dev.allwebleads.com
Content-Length: 2732
Expect: 100-continue
Connection: Keep-Alive

XmlString=<?xml version="1.0" encoding="utf-8"?><InsuranceRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><AffiliateInfo><Username>12696<
/Username><Password>MyPassword</Password><TrackingCampaign>MyTrackingCampaign<
/TrackingCampaign><LeadSourceID>SourceID</LeadSourceID><
/AffiliateInfo><LeadMetaData><LeadBornOnDateTimeUtc>2014-11-11T03:31:21Z</LeadBornOnDateTimeUtc><IpAddress>204.
205.123.125</IpAddress><UserAgent>UserAgentTextGoesHere</UserAgent></LeadMetaData><ZipCode>87194<
/ZipCode><HealthInsurance><ApplicantInfo><FirstName>Uriel</FirstName><LastName>Jones</LastName><DOB>1951-02-20<
/DOB><Gender>Male</Gender><Marital>Married</Marital><Height_FT>6</Height_FT><Height_IN>3</Height_IN><Weight>200<
/Weight><Tobacco>false</Tobacco><Marital>Married</Marital><Occupation>Other</Occupation><USResidence>true<
/USResidence><HouseholdSize>7</HouseholdSize><Income>418945878</Income><
/ApplicantInfo><Dependents><Dependent><FirstName>Jane Smith</FirstName><LastName>Jones</LastName><DOB>1951-02-
20</DOB><Gender>Male</Gender><Height_FT>6</Height_FT><Height_IN>9</Height_IN><Weight>221</Weight><Tobacco>false<
/Tobacco><DependentType>Child</DependentType><Student>true</Student></Dependent><Dependent><FirstName>Sally
Smith</FirstName><LastName>Jones</LastName><DOB>1951-02-20</DOB><Gender>Male</Gender><Marital>Married<
/Marital><Height_FT>7</Height_FT><Height_IN>1</Height_IN><Weight>206</Weight><Tobacco>false<
/Tobacco><DependentType>Spouse</DependentType></Dependent></Dependents><SelfEmployed>false<
/SelfEmployed><DUI>false</DUI><ExpectantMother>false</ExpectantMother><MedicalHistory><Relative_Heart>false<
/Relative_Heart><Relative_Cancer>false</Relative_Cancer><Medication>false</Medication><Medical_Treatment>false<
/Medical_Treatment><Hospital>false</Hospital><Comments>None</Comments><
/MedicalHistory><MajorMedical><AIDS_HIV>true</AIDS_HIV><Alcohol_Drug_Abuse>true<
/Alcohol_Drug_Abuse><Alzheimers_Disease>true</Alzheimers_Disease><Asthma>true</Asthma><Cancer>false<
/Cancer><Cholesterol>false</Cholesterol><Depression>true</Depression><Diabetes>true<
/Diabetes><Heart_Disease>false</Heart_Disease><High_Blood_Pressure>false<
/High_Blood_Pressure><Kidney_Disease>false</Kidney_Disease><Liver_Disease>false<
/Liver_Disease><Mental_Illness>true</Mental_Illness><Pulmonary_Disease>false</Pulmonary_Disease><Stroke>false<
/Stroke><Ulcer>true</Ulcer><Vascular_Disease>true</Vascular_Disease><Other_Major_Disease>true<
/Other_Major_Disease></MajorMedical><CurrentInsurance><CurrentlyInsured>false</CurrentlyInsured><
/CurrentInsurance><RequestedCoverage>Individual Family</RequestedCoverage></HealthInsurance></InsuranceRequest>
```

Health Ping Raw Response

```
HTTP/1.1 200 OK
Content-Length: 329
Content-Type: text/plain
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Wed, 12 Nov 2014 18:00:17 GMT

<?xml version="1.0" encoding="utf-8"?><PricePresentationResult xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.insuranceleads.com/partner
/PricePresentationResult.xsd"><Result>OK</Result><LeadID>677739591</LeadID><Payout>3.33</Payout><
/PricePresentationResult>
```

Health Post Raw Request

```
POST https://dastaging-lm.dev.allwebleads.com/leads/4.0/LeadServiceHttpPost.svc/PricePresentationPostLead HTTP
/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 4.0.30319.1)
Host: dastaging-lm.dev.allwebleads.com
Content-Length: 3123
Expect: 100-continue

LeadID=677739591&XmlString=<?xml version="1.0" encoding="utf-8"?><InsuranceRequest xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><AffiliateInfo><Username>12696<
/Username><Password>MyPassword</Password><TrackingCampaign>MyTrackingCampaign<
/TrackingCampaign><LeadSourceID>SourceID</LeadSourceID><
/AffiliateInfo><LeadMetaData><LeadBornOnDateTimeUtc>2014-11-11T03:31:21Z</LeadBornOnDateTimeUtc><IpAddress>204.
205.123.125</IpAddress><UserAgent>UserAgentTextGoesHere</UserAgent></LeadMetaData><ZipCode>87194<
/ZipCode><ContactInfo><FirstName>Hadley</FirstName><LastName>Koufacos</LastName><Address>85131 Main St<
/Address><ZipCode>87194</ZipCode><City>ALBUQUERQUE</City><County>Bernalillo</County><State>NM<
/State><PhoneDay>7406802874</PhoneDay><PhoneEve>7403071944</PhoneEve><PhoneCell>7406802874<
/PhoneCell><Email>HadleyKoufacos207610736@test.net</Email><Comment>None</Comment><
/ContactInfo><HealthInsurance><ApplicantInfo><FirstName>Uriel</FirstName><LastName>Jones</LastName><DOB>1951-02-
20</DOB><Gender>Male</Gender><Marital>Married</Marital><Height_FT>6</Height_FT><Height_IN>3<
/Height_IN><Weight>200</Weight><Tobacco>false</Tobacco><Marital>Married</Marital><Occupation>Other<
/Occupation><USResidence>true</USResidence><HouseholdSize>7</HouseholdSize><Income>418945878</Income><
/ApplicantInfo><Dependents><Dependent><FirstName>Jane Smith</FirstName><LastName>Jones</LastName><DOB>1951-02-
20</DOB><Gender>Male</Gender><Height_FT>6</Height_FT><Height_IN>9</Height_IN><Weight>221</Weight><Tobacco>false<
/Tobacco><DependentType>Child</DependentType><Student>true</Student></Dependent><Dependent><FirstName>Sally
Smith</FirstName><LastName>Jones</LastName><DOB>1951-02-20</DOB><Gender>Male</Gender><Marital>Married<
/Marital><Height_FT>7</Height_FT><Height_IN>1</Height_IN><Weight>206</Weight><Tobacco>false<
/Tobacco><DependentType>Spouse</DependentType></Dependent></Dependents><SelfEmployed>false<
/SelfEmployed><DUI>false</DUI><ExpectantMother>false</ExpectantMother><MedicalHistory><Relative_Heart>false<
/Relative_Heart><Relative_Cancer>false</Relative_Cancer><Medication>false</Medication><Medical_Treatment>false<
/Medical_Treatment><Hospital>false</Hospital><Comments>None</Comments><
/MedicalHistory><MajorMedical><AIDS_HIV>true</AIDS_HIV><Alcohol_Drug_Abuse>true<
/Alcohol_Drug_Abuse><Alzheimers_Disease>true</Alzheimers_Disease><Asthma>true</Asthma><Cancer>false<
/Cancer><Cholesterol>false</Cholesterol><Depression>true</Depression><Diabetes>true<
/Diabetes><Heart_Disease>false</Heart_Disease><High_Blood_Pressure>false<
/High_Blood_Pressure><Kidney_Disease>false</Kidney_Disease><Liver_Disease>false<
/Liver_Disease><Mental_Illness>true</Mental_Illness><Pulmonary_Disease>false</Pulmonary_Disease><Stroke>false<
/Stroke><Ulcer>true</Ulcer><Vascular_Disease>true</Vascular_Disease><Other_Major_Disease>true<
/Other_Major_Disease></MajorMedical><CurrentInsurance><CurrentlyInsured>false</CurrentlyInsured><
/CurrentInsurance><RequestedCoverage>Individual Family</RequestedCoverage></HealthInsurance></InsuranceRequest>
```

Health Post Raw Response

```
HTTP/1.1 200 OK
Content-Length: 328
Content-Type: text/plain
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Wed, 12 Nov 2014 18:00:18 GMT

<?xml version="1.0" encoding="utf-8"?><PricePresentationResult xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://www.insuranceleads.com/partner
/PricePresentationResult.xsd"><Result>OK</Result><LeadID>52694906</LeadID><Payout>3.33</Payout><
/PricePresentationResult>
```

Sample Client Code Snippets

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)
{
}
?>
```