Travelport Trip Services AirSearch

The Travelport Trip Services APIs offer an AirSearch method that enables developers to quickly and efficiently perform a lowest-fare search against Travelport supplier content. The request is RESTful and can either be in XML or JSON. The Trip Services APIs are modeled on the Open Travel Alliance (OTA) API standard and Open Distribution Methodology (ODM) concepts, making them easy to work with, conceptually well-documented, and standardized.

AirSearch offers mobile-optimized search, using pagination and a streamlined number of request parameters to increase performance and decrease response time compared to Travelport's full-featured search API, Universal API Low Fare Search. By optimizing response messaging, Travelport was able to reduce payload size without sacrificing information important to the customer.

The AirSearch API is best suited for mobile applications requiring faster and lighter responses, for developers looking to quickly build a Minimum Viable Product (MVP) in the travel search space, and for developers who may be new to the travel technology domain. It is not intended to have every feature and search option available, and strikes a balance between speed, performance, and the majority of travel search developer needs.

AirSearch Requests

The AirSearch message has two types of transactions – a POST request for the initial query and a GET request for the follow-on paginated query:

1. An HTTP POST request generates a shop response and returns the first page of that response. If the query requests all response data, the first page in the response will be the only page.
2. An HTTP GET request retrieves any page from a previously generated response.

The POST request must include an OffersSearchQuery object containing the following:

- **PassengerCriteria:** One PassengerCriteria object for each distinct passenger type, including the Passenger Type Code (PTC) and the number of passengers of that type.
- **SearchCriteriaSegment:** Segment details of the origin and destination (O&D) pair and dates. Same day flights (there and back in one day, or two different flights with different departure cities in one day) require a departure time in addition to a departure date.
- **PseudoCityInfo:** The Pseudo City Code (PCC) identifying your provisioning settings.

A basic AirSearch XML request would look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<OffersSearchQuery xmlns="http://www.travelport.com/schema/catalog/air/v5"
maxNumberOfOffersToReturn="50">
  <PassengerCriteria number="1">ADT</PassengerCriteria>
  <SearchCriteriaSegment departureDate="2017-12-05">
        <From>LAX</From>
        <To>JFK</To>
  </SearchCriteriaSegment>
  <PseudoCityInfo>UP9</PseudoCityInfo>
</OffersSearchQuery>
```

AirSearch Responses

The response for both POST and GET requests includes a number of Offer objects. Each Offer summarizes the key information related to a unique search result.

An Offer includes all segments (Products) that make up an itinerary at a single price point (PriceAir). An Offer can contain multiple Products, each representing a unique segment/O&D pair. Each Products object will have multiple ProductTransport elements whenever there are multiple flight options for a specific segment of the itinerary. One of any of the ProductTransport objects can be chosen within a Products object, and one of each of the Products must be selected to complete the itinerary.

The following example response shows an Offer with a unique id of “o3”. It includes one Products object, which has one ProductTransport (unique O&D) with the sub-segments “s4” and “s5”, which make up both the segment and the itinerary because this was a simple one-way search. If the itinerary were expanded to a round trip search, there would be two Products (one for each O&D), and the potential for multiple ProductTransport objects within each Products, depending on the number of flight options. For example, the same airline may offer multiple flights a day at the same price between two cities. These options would each be listed as a separate ProductTransport.

Connections may also be displayed within each ProductTransport, indicated by the Connection element. In this example, there is one segment consisting of two flights, all included under one ProductTransport. The total duration of the segment is provided in hours (H) and minutes (M). In each segment, and for each flight, AirSearch returns a Class of Service (booking code) and Cabin Class (e.g., Economy).

```
……
<ctlg-0200:Offer id="o3">
    <ctlg-0200:Products>
      <ctlg-0200:ProductTransport duration="PT9H50M" id="p0">
        <ctlg-0200:Connection inboundRef="s4" outboundRef="s5" duration="PT2H13M"/>
        <ctlg-0200:SegmentID ref="s4 s5"/>
        <ctlg-0200:CabinClass segmentRef="s4 s5" classOfService="U">Economy</ctlg-0200:CabinClass>
      </ctlg-0200:ProductTransport>
    </ctlg-0200:Products>
    <ctlg-0200:PriceAir>
      <ctar-0200:AirPrice ptc="ADT" quantity="1" requestedPTC="ADT">
        <fin-0200:Amount>
          <fin-0200:Base>329.00</fin-0200:Base>
          <fin-0200:Taxes totalTaxes="55.80"/>
          <fin-0200:Total>384.80</fin-0200:Total>
        </fin-0200:Amount>
      </ctar-0200:AirPrice>
    </ctlg-0200:PriceAir>
    <ctlg-0200:TermsAndConditionsTransport/>
  </ctlg-0200:Offer>
……..
```

Pricing information is returned in PriceAir, above. The AirPrice object returns pricing for each unique PTC. In this example, the total price is $384.80 for the Adult PTC, which is ADT. This itinerary has only one passenger and one PTC, and therefore has only one PriceAir. Remember that each price is for one PTC, so you will need to sum all the PTC prices (multiplied by the number of passengers for each) to reach a total price for all passengers combined. This allows you to choose how to display pricing data, either at the passenger level or the itinerary (all passengers) level. Notice that the total is also broken down into a base fare and taxes to allow for flexibility of display.

The TermsAndConditionsTransport element within each Offer further explains what comes with the fare and may include structured sub-element information like baggage allowance, if applicable. In this example no further information was filed for that fare so an empty TermsAndConditionsTransport element is returned.

The response also includes an expanded segment details section (ReferenceListSegment), shown below, providing additional information about the individual flights, such as carrier, flight number, departure/arrival airports, date and time, duration, equipment type, and distance traveled. In this example, ReferenceListSegment provides additional details for both flights in the segment, identified above in ProductTransport and here as "s4" and "s5".

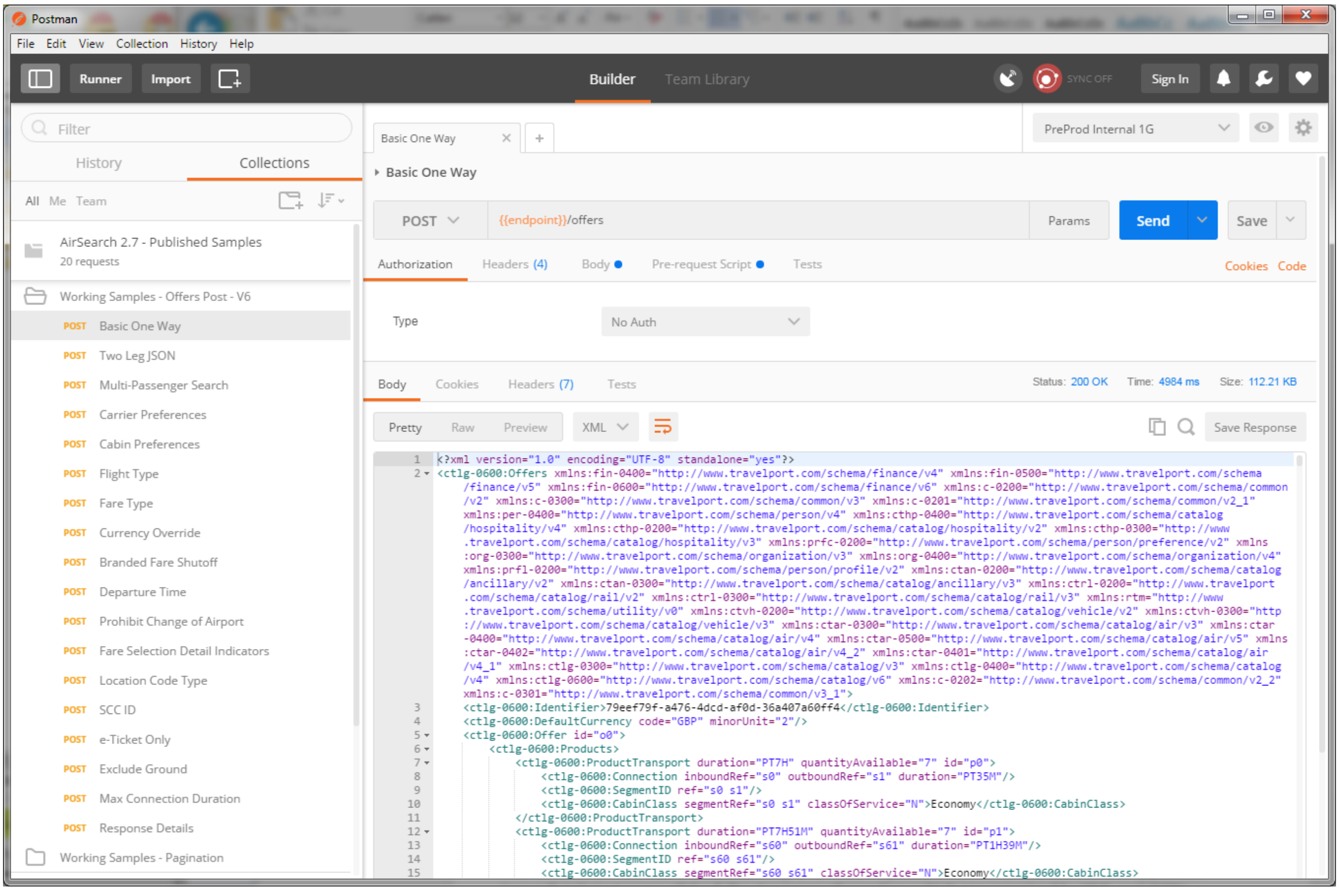
```
<ctlg-0200:ReferenceListSegment>
    ……..
    <ctlg-0200:Segment distance="2338" duration="PT4H41M" id="s4">
      <ctlg-0200:TransportFlight>
        <ctar-0200:Flight carrier="NK" number="0310" equipment="319"/>
      </ctlg-0200:TransportFlight>
      <ctlg-0200:Departure location="LAX" date="2017-12-05" time="21:25:00.000"/>
      <ctlg-0200:Arrival location="FLL" date="2017-12-06" time="05:06:00.000"/>
    </ctlg-0200:Segment>
    <ctlg-0200:Segment distance="1085" duration="PT2H56M" id="s5">
      <ctlg-0200:TransportFlight>
        <ctar-0200:Flight carrier="NK" number="0604" equipment="320"/>
      </ctlg-0200:TransportFlight>
      <ctlg-0200:Departure location="FLL" date="2017-12-06" time="07:19:00.000"/>
      <ctlg-0200:Arrival location="LGA" date="2017-12-06" time="10:15:00.000"/>
    </ctlg-0200:Segment>
    ……..
</ctlg-0200:ReferenceListSegment>
……
```

Trying AirSearch in Postman

Travelport provides Postman to use as a test environment for AirSearch. You can run all currently available AirSearch options either completely within Postman, or you can test AirSearch by copying preformatted requests from Postman into your own application.

Start by setting up the Postman environment, then import the AirSearch project available for download from the Trip Services online help:

1. Download Postman at https://www.getpostman.com/.
2. Download the latest release of AirSearch from the online help.
3. Import the downloaded collection and environment files.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ctlg-0600:Offers xmlns:fin-0400="http://www.travelport.com/schema/finance/v4" xmlns:fin-0500="http://www.travelport.com/schema
    /finance/v5" xmlns:fin-0600="http://www.travelport.com/schema/finance/v6" xmlns:c-0200="http://www.travelport.com/schema/common
    /v2" xmlns:c-0300="http://www.travelport.com/schema/common/v3" xmlns:c-0201="http://www.travelport.com/schema/common/v2_1"
    xmlns:per-0400="http://www.travelport.com/schema/person/v4" xmlns:cthp-0400="http://www.travelport.com/schema/catalog
    /hospitality/v4" xmlns:cthp-0200="http://www.travelport.com/schema/catalog/hospitality/v2" xmlns:cthp-0300="http://www
    .travelport.com/schema/catalog/hospitality/v3" xmlns:prfc-0200="http://www.travelport.com/schema/person/preference/v2" xmlns
    :org-0300="http://www.travelport.com/schema/organization/v3" xmlns:org-0400="http://www.travelport.com/schema/organization/v4"
    xmlns:prfl-0200="http://www.travelport.com/schema/person/profile/v2" xmlns:ctan-0200="http://www.travelport.com/schema/catalog
    /ancillary/v2" xmlns:ctan-0300="http://www.travelport.com/schema/catalog/ancillary/v3" xmlns:ctrl-0200="http://www.travelport
    .com/schema/catalog/rail/v2" xmlns:ctrl-0300="http://www.travelport.com/schema/catalog/rail/v3" xmlns:rtm="http://www
    .travelport.com/schema/utility/v0" xmlns:ctvh-0200="http://www.travelport.com/schema/catalog/vehicle/v2" xmlns:ctvh-0300="http
    ://www.travelport.com/schema/catalog/vehicle/v3" xmlns:ctar-0300="http://www.travelport.com/schema/catalog/air/v3" xmlns:ctar
    -0400="http://www.travelport.com/schema/catalog/air/v4" xmlns:ctar-0500="http://www.travelport.com/schema/catalog/air/v5" xmlns
    :ctar-0402="http://www.travelport.com/schema/catalog/air/v4_2" xmlns:ctar-0401="http://www.travelport.com/schema/catalog/air
    /v4_1" xmlns:ctlg-0300="http://www.travelport.com/schema/catalog/v3" xmlns:ctlg-0400="http://www.travelport.com/schema/catalog
    /v4" xmlns:ctlg-0600="http://www.travelport.com/schema/catalog/v6" xmlns:c-0202="http://www.travelport.com/schema/common/v2_2"
    xmlns:c-0301="http://www.travelport.com/schema/common/v3_1">
    <ctlg-0600:Identifier>79eef79f-a476-4dcd-af0d-36a407a60ff4</ctlg-0600:Identifier>
    <ctlg-0600:DefaultCurrency code="GBP" minorUnit="2"/>
    <ctlg-0600:Offer id="o0">
        <ctlg-0600:Products>
            <ctlg-0600:ProductTransport duration="PT7H" quantityAvailable="7" id="p0">
                <ctlg-0600:Connection inboundRef="s0" outboundRef="s1" duration="PT35M"/>
                <ctlg-0600:SegmentID ref="s0 s1"/>
                <ctlg-0600:CabinClass segmentRef="s0 s1" classOfService="N">Economy</ctlg-0600:CabinClass>
            </ctlg-0600:ProductTransport>
            <ctlg-0600:ProductTransport duration="PT7H51M" quantityAvailable="7" id="p1">
                <ctlg-0600:Connection inboundRef="s60" outboundRef="s61" duration="PT1H39M"/>
                <ctlg-0600:SegmentID ref="s60 s61"/>
                <ctlg-0600:CabinClass segmentRef="s60 s61" classOfService="N">Economy</ctlg-0600:CabinClass>
```

The left side of Postman includes pre-formed requests provided by Travelport. These are the currently supported AirSearch modifiers. Within this test environment, you can immediately test the following types of searches:

- Basic one-way search (XML)
- Basic roundtrip search (JSON)
- Pagination example
  - Initial POST request
  - Follow-on GET request
- Several modifiers allowing you to refine the basic search:
  - Preferences such as carrier, cabin, flight type, and/or maximum connection time.
  - Fare and ticketing modifiers such as private/public fares, branded fares, and e-ticketing.

The right side of Postman displays a panel with the Authorization, Headers, and Body tabs. These are all pre-populated, allowing you to click the Send button to test a given request and view a real-time response. You can copy the XML or JSON generated in the Postman test suite and paste it into your HTTP client with the appropriate header information to try it in your own application. You can also create or import environment files to handle the credentials (upper right-hand corner). See the online help for details about all currently supported AirSearch modifiers, which are also available for testing in Postman.

More Information

For more about using AirSearch and all Trip Services APIs, see the online help at https://support.travelport.com/webhelp/TripServices/.