**Sabre Connector**

The Subre [connector](https://docs.wso2.com/display/ESB490/Working+with+Connectors) allows you to access the travel information on demand with our [REST and SOAP APIs](https://developer.sabre.com/docs/home/). through WSO2 ESB. Sabre is a web and mobile application designed to help,

* Search for air, car, hotel, rail, and cruise
* Service travel reservations
* Build process automation tools
* Manage customer profiles

**Getting started**

To get started, go to[Configuring Sabre Operations](https://developer.sabre.com/login/login?r=https%3A%2F%2Fdeveloper.sabre.com%2Fdocs%2Fhome%2F&h=31cc361b15bc6c35bec6851225d92872). Once you have completed your configurations, you can perform various operations with the connector.

**Configuring Sabre Operations**

To use the Subre connector, add the <sabre.init> element in your configuration before carrying out any other Subre operations.

Sabre uses OAuth 2.0 Authentication for authenticating API calls. For more information on authentication, see<https://developer.sabre.com/docs/read/rest_basics/authentication>

|  |
| --- |
| **Init** |
| **<sabre.init>**  **<accessToken>{$ctx:accessToken}</accessToken>**  **<apiUrl>{$ctx:apiUrl}</apiUrl>**  **<apiVersion>{$ctx:apiVersion}</apiVersion>**  **<clientId>{$ctx:clientId}</clientId>**  **<clientSecret>{$ctx:clientSecret}</clientSecret>**  **</sabre.init>** |

**Properties**

●      **apiUrl**:   The URL of the Sabre REST API.

●      **apiVersion**: The Version of Sabre account.

●      **accessToken**:        The token to use in future requests against the API.

●      **clientId**:            The Client ID uniquely identifies the application making the request.

●      **clientSecret**:    The Client Secret belonging to the app, found in the details pane of

the developer console.

Now that you have connected to Sabre, use the information in the following topics to perform various operations with the connector.

●     Working with Air Search in Sabre

●     Working with Air Utility in Sabre

●      Working with Ground Transportation Search  in Sabre

●      Working with Utility in Sabre

●      Working with Hotel in Sabre

**Working with Air Search in Sabre**

**Overview**

The following operations allow you to work with Air Search.

For a sample proxy service that illustrates how to work with attachments, see Sample configuration.

|  |  |
| --- | --- |
| **Operations** | **Description** |
| instaFlightsSearch | To finds one-way or roundtrip flight itineraries (with fares) on future dates of travel for a given origin/destination. |
| flightsTo | Finds the 20 lowest published fares available for a given destination. Destination is always required. |
| leadPriceCalendar | Finds the lowest nonstop fare and the lowest overall fare on future dates of travel. |

**Operation details**

details on the operations related to Air Search.

**Insta Flights Search**

The instaFlightsSearch operation finds one-way or roundtrip flight itineraries (with fares) on future dates of travel for a given origin/destination.

|  |
| --- |
| **instaFlightsSearch** |
| <sabre.instaFlightsSearch>  <origin>{$ctx:origin}</origin>  <destination>{$ctx:destination}</destination> <departuredate>{$ctx:departuredate}</departuredate> <returndate>{$ctx:returndate}</returndate> <includedcarriers>{$ctx:includedcarriers}</includedcarriers> <excludedcarriers>{$ctx:excludedcarriers}</excludedcarriers> <outboundflightstops>{$ctx:outboundflightstops}</outboundflightstops>  <includedconnectpoints>{$ctx:includedconnectpoints}</includedconnectpoints>  <excludedconnectpoints>{$ctx:excludedconnectpoints}</excludedconnectpoints>  <inboundflightstops>{$ctx:inboundflightstops}</inboundflightstops>  <outboundstopduration>{$ctx:outboundstopduration}</outboundstopduration>  <inboundstopduration>{$ctx:inboundstopduration}</inboundstopduration>  <outbounddeparturewindow>{$ctx:outbounddeparturewindow}</outbounddeparturewindow>  <inbounddeparturewindow>{$ctx:inbounddeparturewindow}</inbounddeparturewindow>  <outboundarrivalwindow>{$ctx:outboundarrivalwindow}</outboundarrivalwindow> <inboundarrivalwindow>{$ctx:inboundarrivalwindow}</inboundarrivalwindow> <onlineitinerariesonly>{$ctx:onlineitinerariesonly}</onlineitinerariesonly>  <minfare>{$ctx:minfare}</minfare>  <maxfare>{$ctx:maxfare}</maxfare>  <limit>{$ctx:limit}</limit>  <offset>{$ctx:offset}</offset>  <eticketsonly>{$ctx:eticketsonly}</eticketsonly>  <sortby>{$ctx:sortby}</sortby>  <order>{$ctx:order}</order>  <sortby2>{$ctx:sortby2}</sortby2>  <order2>{$ctx:order2}</order2>  <pointofsalecountry>{$ctx:pointofsalecountry}</pointofsalecountry>  <passengercount>{$ctx:passengercount}</passengercount>  <view>{$ctx:view}</view>  </sabre.instaFlightsSearch> |

**Properties**

* **origin**:3-letter IATA airport code or city code of the departure airport.
* **destination**:3-letter IATA airport code or city code of the arrival airport.
* departuredate:Format: YYYY-MM-DD.
* returndate:Format: YYYY-MM-DD.
* includedcarriers:List of IATA airline codes.
* excludedcarriers:List of IATA airline codes. Format: XX,YY,ZZ.
* outboundflightstops:Max number of flight connections. Values 0-99.
* includedconnectpoints:List of 3 letter IATA airport codes. Format: XXX,YYY,ZZZ. Max 3 airport codes allowed.
* excludedconnectpoints:List of 3 letter IATA airport codes. Format: XXX,YYY,ZZZ. Max 3 airport codes allowed.
* inboundflightstops:Max number of flight connections. Values 0-99."/>
* outboundstopduration:The sum of the total waiting time for all connections on an outbound itinerary. Values 0-9999.
* inboundstopduration:The sum of the total waiting time for all connections on an inbound itinerary. Values 0-9999.
* outbounddeparturewindow:Time range during which a first outbound flight segment can depart on the departure date. Format: HHMMHHMM.
* inbounddeparturewindow :Time range during which a first inbound flight segment can depart on the departure date. Format: HHMMHHMM.
* outboundarrivalwindow:Time range during which a last outbound flight segment can arrive, after the departure date. Format: HHMMHHMM.
* inboundarrivalwindow:Time range during which a last inbound flight segment can arrive on the return date. Format: HHMMHHMM.
* onlineitinerariesonly:An indicator to base the response on online or interline itineraries.
* minfare:Minimum overall lead fare.
* maxfare:Maximum overall lead fare.
* limit:Limit the number of itineraries returned.
* offset:Retrieves the itineraries that start with the number in offset through the quantity in limit, inclusive.
* eticketsonly:An indicator to only retrieve itineraries that are e-ticket eligible.
* sortby:Primary sort object in the response.
* order:Sorting order for the primary sortby.
* sortby2:Secondary sort object in the response.
* order2:Sorting order for sortby2.
* pointofsalecountry:2-letter country code of the point of sale.
* passengercount:Valid values: 1 to 10, inclusive.
* view:A Sabre response view definition.

**Sample request**

Following is a sample REST/JSON request that can be handled by the instaFlightsSearch  operation.

|  |
| --- |
| **Sample request for instaFlightSearch** |
| {  "apiUrl":"https://api.test.sabre.com",  "apiVersion":"v1",  "clientId":"V1:ywxj6o73qm3jz21j:DEVCENTER:EXT",  "clientSecret":"8rYYyLg4",  "destination":"LAX",  "departuredate":"2017-03-07",  "returndate":"2017-03-08",  "origin":"JFK",  "limit":"1"  } |

**Flights To**

The flightsTo Finds the 20 lowest published fares available for a given destination. Destination is always required.

|  |
| --- |
| **flightsTo** |
| <sabre.flightsTo>  <destination>{$ctx:destination}</destination> <pointofsalecountry>{$ctx:pointofsalecountry}</pointofsalecountry>  </sabre.flightsTo> |

**Properties**

* destination:3-letter IATA airport code or city code of the arrival airport.
* pointofsalecountry:2-letter country code of the point of sale.

**Sample request**

Following is a sample REST/JSON request that can be handled by the flightsTo operation.

|  |
| --- |
| **Sample request for flightsTo** |
| {  "apiUrl":"https://api.test.sabre.com",  "apiVersion":"v1",  "clientId":"V1:ywxj6o73qm3jz21j:DEVCENTER:EXT",  "clientSecret":"8rYYyLg4",  "destination":"LAX"  } |

**Sample configuration**

Following is a sample proxy service that illustrates how to connect to Sabre with the init operation and use the flightsTo operation.

|  |
| --- |
| **Sample Proxy** |
| **<?xml version="1.0" encoding="UTF-8"?>**  **<proxy xmlns="http://ws.apache.org/ns/synapse"**  **name="flightsTo"**  **transports="http,https"**  **statistics="disable"**  **trace="disable"**  **startOnLoad="true">**  **<target>**  **<inSequence>**  **<property name="accessToken" expression="json-eval($.accessToken)"/>**  **<property name="apiUrl" expression="json-eval($.apiUrl)"/>**  **<property name="apiVersion" expression="json-eval($.apiVersion)"/>**  **<property name="clientId" expression="json-eval($.clientId)"/>**  **<property name="clientSecret" expression="json-eval($.clientSecret)"/>**    **<property name="destination" expression="json-eval($.destination)"/>**  **<property name="pointofsalecountry" expression="json-eval($.pointofsalecountry)"/>**  **<sabre.init>**  **<accessToken>{$ctx:accessToken}</accessToken>**  **<apiUrl>{$ctx:apiUrl}</apiUrl>**  **<apiVersion>{$ctx:apiVersion}</apiVersion>**  **<clientId>{$ctx:clientId}</clientId>**  **<clientSecret>{$ctx:clientSecret}</clientSecret>**  **</sabre.init>**  **<sabre.flightsTo>**  **<destination>{$ctx:destination}</destination>**  **<pointofsalecountry>{$ctx:pointofsalecountry}</pointofsalecountry>**  **</sabre.flightsTo>**  **<respond/>**  **</inSequence>**  **<outSequence>**  **<log/>**  **<send/>**  **</outSequence>**  **</target>**  **<description/>**  **</proxy>** |

**Working with Air intelligence in Sabre**

**Overview**

The following operations allow you to work with events.

For a sample proxy service that illustrates how to work with events, see Sample configuration.

|  |  |
| --- | --- |
| **Operations** | **Description** |
| fareRange | Returns median, highest, and lowest published fares that were ticketed during the previous 4 weeks. |
| lowFareForecast | Forecasts the price range into which the lowest published fare is predicted to fall within the next 7 days, and returns a recommendation to buy or wait. |
| lowFareHistory | Returns each day of the past two weeks' lowest published fares for a given city/date pair. |
| travelSeasonality | Returns a traffic volume rating to a destination airport for each of the next 52 weeks |

**Operation details**

This section provides details on each of the operations.

**Fare Range**

The fareRange operation is returns median, highest, and lowest published fares that were ticketed during the previous 4 weeks.

|  |
| --- |
| **fareRange** |
| <sabre.fareRange>  <origin>{$ctx:origin}</origin> <destination>{$ctx:destination}</destination> <earliestdeparturedate>{$ctx:earliestdeparturedate}</earliestdeparturedate> <latestdeparturedate>{$ctx:latestdeparturedate}</latestdeparturedate>  <lengthofstay>{$ctx:lengthofstay}</lengthofstay>  <excludelcc>{$ctx:excludelcc}</excludelcc>  </sabre.fareRange> |

**Properties**

* origin:3-letter IATA airport code or city code of the departure airport.
* destination:3-letter IATA airport code or city code of the arrival airport.
* earliestdeparturedate:Format: YYYY-MM-DD. The earliest date of departure in the range. The date can be a maximum of 90 days in the future, including the day of the request.
* latestdeparturedate:Format: YYYY-MM-DD. The latest date of departure in the range. The date can be a maximum of 90 days in the future, including the day of the request.
* lengthofstay:Valid values: 0 to 16, inclusive.
* excludelcc:Limits the response to exclude low cost carriers (LCCs).

**Sample request**

Following is a sample REST/JSON request that can be handled by the fareRange operation.

|  |
| --- |
| **Sample Request for fareRange** |
| {  "apiUrl":"https://api.test.sabre.com",  "apiVersion":"v1",  "clientId":"V1:ywxj6o73qm3jz21j:DEVCENTER:EXT",  "clientSecret":"8rYYyLg4",  "destination":"LAX",  "origin":"JFK",  "earliestdeparturedate":"2017-04-09",  "latestdeparturedate":"2017-04-10",  "lengthofstay":"1"  } |

**Low Fare Forecast**

The lowFareForecast operation is forecasts the price range into which the lowest published fare is predicted to fall within the next 7 days, and returns a recommendation to buy or wait.

|  |
| --- |
| **lowFareForecast** |
| <sabre.lowFareForecast>  <origin>{$ctx:origin}</origin>  <destination>{$ctx:destination}</destination>  <departuredate>{$ctx:departuredate}</departuredate>  <returndate>{$ctx:returndate}</returndate>  </sabre.lowFareForecast> |

**Properties**

* origin:3-letter IATA airport code or city code of the departure airport.
* destination:3-letter IATA airport code or city code of the arrival airport.
* departuredate:Format: YYYY-MM-DD. Valid dates are a maximum of 60 future dates.
* returndate:Format: YYYY-MM-DD. Valid dates are a maximum of 60 future dates.

**Sample request**

Following is a sample REST/JSON request that can be handled by the lowFareForecast operation.

|  |
| --- |
| **Sample Request for fareRange** |
| {  "apiUrl":"https://api.test.sabre.com",  "apiVersion":"v2",  "clientId":"V1:ywxj6o73qm3jz21j:DEVCENTER:EXT",  "clientSecret":"8rYYyLg4",  "destination":"LAS",  "departuredate":"2017-03-08",  "returndate":"2017-03-09",  "origin":"DFW"  } |

**Low Fare History**

The lowFareHistory operation is returns each day of the past two weeks' lowest published fares for a given city/date pair.

|  |
| --- |
| **lowFareHistory** |
| <sabre.lowFareHistory>  <origin>{$ctx:origin}</origin>  <destination>{$ctx:destination}</destination>  <departuredate>{$ctx:departuredate}</departuredate>  <returndate>{$ctx:returndate}</returndate>  <pointofsalecountry>{$ctx:pointofsalecountry}</pointofsalecountry>  </sabre.lowFareHistory> |

**Properties**

* origin:3-letter IATA airport code or city code of the departure airport.
* destination:3-letter IATA airport code or city code of the arrival airport.
* departuredate:Format: YYYY-MM-DD.
* returndate:Format: YYYY-MM-DD.
* pointofsalecountry:List of IATA airline codes.

**Sample request**

Following is a sample REST/JSON request that can be handled by the lowFareHistory operation.

|  |
| --- |
| **Sample Request for lowFareHistory** |
| {  "apiUrl":"https://api.test.sabre.com",  "apiVersion":"v1",  "clientId":"V1:ywxj6o73qm3jz21j:DEVCENTER:EXT",  "clientSecret":"8rYYyLg4",  "destination":"LAX",  "departuredate":"2017-03-10",  "returndate":"2017-03-11",  "origin":"JFK"  } |

**Travel Seasonality**

The travelSeasonality operation is a traffic volume rating to a destination airport for each of the next 52 weeks.

|  |
| --- |
| **travelSeasonality** |
| <sabre.travelSeasonality>  <destination>{$ctx:destination}</destination>  </sabre.travelSeasonality> |

**Properties**

* destination:3-letter IATA airport code or city code of the arrival airport.

**Sample request**

Following is a sample REST/JSON request that can be handled by the travelSeasonality operation.

|  |
| --- |
| **Sample Request for travelSeasonality** |
| {  "apiUrl":"https://api.test.sabre.com",  "apiVersion":"v1",  "clientId":"V1:ywxj6o73qm3jz21j:DEVCENTER:EXT",  "clientSecret":"8rYYyLg4",  "destination":"DFW"  } |

**Sample configuration**

Following is a sample proxy service that illustrates how to connect to Sabre with the init operation and use the fareRange operation.

|  |
| --- |
| **Sample Proxy** |
| **<?xml version="1.0" encoding="UTF-8"?>**  **<proxy xmlns="http://ws.apache.org/ns/synapse"**  **name="fareRange"**  **transports="http,https"**  **statistics="disable"**  **trace="disable"**  **startOnLoad="true">**  **<target>**  **<inSequence>**  **<property name="accessToken" expression="json-eval($.accessToken)"/>**  **<property name="apiUrl" expression="json-eval($.apiUrl)"/>**  **<property name="apiVersion" expression="json-eval($.apiVersion)"/>**  **<property name="clientId" expression="json-eval($.clientId)"/>**  **<property name="clientSecret" expression="json-eval($.clientSecret)"/>**  **<property name="origin" expression="json-eval($.origin)"/>**  **<property name="destination" expression="json-eval($.destination)"/>**  **<property name="earliestdeparturedate" expression="json-eval($.earliestdeparturedate)"/>**  **<property name="latestdeparturedate" expression="json-eval($.latestdeparturedate)"/>**  **<property name="lengthofstay" expression="json-eval($.lengthofstay)"/>**  **<property name="excludelcc" expression="json-eval($.excludelcc)"/>**  **<sabre.init>**  **<accessToken>{$ctx:accessToken}</accessToken>**  **<apiUrl>{$ctx:apiUrl}</apiUrl>**  **<apiVersion>{$ctx:apiVersion}</apiVersion>**  **<clientId>{$ctx:clientId}</clientId>**  **<clientSecret>{$ctx:clientSecret}</clientSecret>**  **</sabre.init>**  **<sabre.fareRange>**  **<origin>{$ctx:origin}</origin>**  **<destination>{$ctx:destination}</destination>**  **<earliestdeparturedate>{$ctx:earliestdeparturedate}</earliestdeparturedate>**  **<latestdeparturedate>{$ctx:latestdeparturedate}</latestdeparturedate>**  **<lengthofstay>{$ctx:lengthofstay}</lengthofstay>**  **<excludelcc>{$ctx:excludelcc}</excludelcc>**  **</sabre.fareRange>**  **<respond/>**  **</inSequence>**  **<outSequence>**  **<log/>**  **<send/>**  **</outSequence>**  **</target>**  **<description/>**  **</proxy>** |