

# Sputnik Fares API: Documentation



Functionality

## OVERVIEW + ENDPOINTS

`https://openair-{{envVariable}}.airtrfx.com/airfare-sputnik-service/v3/{{tenantCode}}/fares/{{endpoint}}`

### SEARCH

`/search`

Returns flight fares for a specific Tenant based on a number of filtering criteria.

### ROUTE SEARCH

`/route-search`

Returns flight fares for a specific Tenant based on a number of filtering criteria for one, or many, sets of routes. The response can group together all the information, or return it separated by route request within the body. Currently used by the Standard Fare Module.

### HISTOGRAM DISTRIBUTION

`/histogram-distribution`

Returns daily, weekly, or monthly property prices for a specific Tenant based on a **specific route** and number of filtering criteria.

### TRENDING DESTINATIONS

`/trending-destinations`

Returns flight fares for a specific Tenant based on a number of filtering criteria and organizes the **response based on the Destination**. The request allows to specify the number of Destinations that can be returned, and how many fares per destination.

### PRICE STATS

`/price-stats`

Returns price stats based on a number of criteria. Used internally for the buckets filtering.

*Documentation Pending*

### Global Filters

Can be applied across all endpoints. See below for some of the shared capabilities. *For endpoint specific info, please see specific API documentation listed above.*

## AUTOMATED

### Smart Requests

#### ALL ENDPOINTS

Smart Requests with auto-loaded setting configs provides industry and customer business logic to front-end services in order to simplify the required request parameters.

In order to enable the Smart Sputnik response, the "autoSettings" object is required with fields "language" and "market": "autoSettings": {"language": "en", "market": "us"}. Additionally, any of the automated settings can be overwritten by sending the field in the request.

#### Automated Settings:

- Date Format: format required for specific Site Edition.
- Output Currencies: currencies required for specific Site Edition.
- Currency Format: currency format for specific Site Edition.
- Journey Type: journey type required for specific Site Edition.
- Data Expiration: data expiration required for specific Site Edition.
- Language: language code required for specific Site Edition.
- Output Fields: output fields required for tenant.
- Sort Order: sort order required for specific tenant.

Please note that all items listed as "for specific Site Edition" will use the value designated within the Site Edition Central Settings, and if not will use the Default Tenant Settings. For more information, please see: [Central Settings API: Endpoints - Airline, Package, Bus and How to configure Site Edition Settings in TRFX CMS?](#)

## DATA FILTERING

### Origin + Destination Filtering

#### Iata Code **LIMITED**

Filtering is available based on one, or many, iata code(s) for both Origin and/or Destination. When sending a city iata code (i.e. 'NYC') all fares available to the city iata code or to Airport iata codes associated to the City Code. See more: [How are fares returned for City Codes in Sputnik?](#).

All endpoints, except for Route Search.

#### Market Code **ALL ENDPOINTS**

Filtering is available based on one, or many, Country Market code(s) for both Origin and/or Destination. This will base the Origin(s) and/or Destination(s) based on the Country Market of the Airport / Bus Code associated with the fare. See more: [How does Country Market Code Filtering + Outputting work?](#).

All endpoints.

#### Route **LIMITED**

Route filtering, based on iata codes, is available. When sending a city iata code (i.e. 'NYC') all fares available to the city iata code or to Airport iata codes associated to the City Code. See more: [How are fares returned for City Codes in Sputnik?](#).

- Histogram requires one route to be sent. See: [Sputnik: histogram-distribution](#).
- Route Search allows one, or many, routes to be sent. See: [Sputnik: route-search](#).

Only Histogram Distribution, and Route Search.

#### TRFX Routes **ALL ENDPOINTS**

Filtering is available to only include fares that are related to routes in the TRFX Route List.

All endpoints.

#### Geo ID **ALL ENDPOINTS**

Filtering is available based on specific Geo IDs for each Hierarchy level for Origins and/or Destinations. This includes: Airport, City, State, Country, and Region.

All endpoints.

#### Category + Tag **ALL ENDPOINTS**

Filtering is available for categories and/or tags for the Origin and/or Destination (*although most commonly used only for Destination*). This can be specified for the overall hierarchy of a location, or at a specific hierarchy level. See: [How does Categories / Tags Filtering + Outputting work?](#).

All endpoints.

#### Geo-coordinate **ALL ENDPOINTS**

Filtering is available based on geo-coordinates. Geo Coordinate object filter includes two optional attributes, "origin" and "destination". Each of them allows to specify to search by a quadrant, "square", or from a specific point and a radius surrounding it, "radius".

When "square" is used, two required objects must be included: "topLeft" and "bottomRight". Each Object will include two required Double values: "latitude" and "longitude".

When "radius" is used, the "point" object is required with "latitude" and "longitude" Double Values required within, and the "point" Double Value.

All endpoints.

### Itinerary Filtering

## Currency + Redemption Type **ALL ENDPOINTS**

Filtering is available for requesting different types of redemption fares, and will include additional information in the output.

All endpoints.

## Budget Range **ALL ENDPOINTS**

Filtering is available for a set budget, based on a minimum and / or a maximum fare amount. Each is optional, but at least one is required. Additionally, the filter allows to specify the source, whether the Total Price (*localize currency*) or USD Total Price (*normalized currency*) should be used.

The latter is optional and applies to USD Total Price when no currency is specified in the request, or Total Price when a currency is specified in the request.

All endpoints.

## Travel Class **ALL ENDPOINTS**

Filtering is available for none, one, or many Travel Classes, and returns fares within the listed travel classes. This requires the Tenant to have the data enabled. The Travel Class refers to the normalized FareNet Travel Classes.

All endpoints.

## Branded Travel Class **ALL ENDPOINTS**

Filtering is available for none, one, or many of the Branded Travel Classes, and returns fares within the listed travel classes. This requires the Tenant to have the data enabled. The Branded Travel Class vary by tenant and can be retrieved from the Organization Service > Central Settings API. See: [Central Settings API: Endpoints - Airline, Package, Bus](#).

All endpoints.

## Codeshare / Interline **ALL ENDPOINTS**

Filtering is available to allow requesting fares that are either codeshare, interline, or neither.

All endpoints.

## Route Type **ALL ENDPOINTS**

Filtering is available to allow requesting fares that are a specific route type, such as nonstop, direct, and / or connecting.

All endpoints.

## Number of Stops **ALL ENDPOINTS**

Filtering is available based on the number of stops an itinerary has for the outbound and/or inbound. Flight Stop Count filters based on a minimum and / or a maximum number of stops. This applies to flightStopCount of the overall document, which is the maximum number of stops between outbound and inbound.

All endpoints.

### ▼ [Date Filtering](#)

## Specific Dates **ALL ENDPOINTS**

Filtering is enabled for both Departure and/or Return date ranges, that can include a start or start and end. This filter requires specific dates to be added for each, and returns fares falling within the range specified, or after the start specified.

All endpoints.

## Departure / Return Intervals **ALL ENDPOINTS**

Filtering is enabled for both Departure and/or Return interval ranges, that can include a start or start and end. This filter requires an integer to be added for each that amounts to number of days added to today's date, and returns fares falling within the range specified, or after the start specified.

All endpoints.

## Blackout Dates **ALL ENDPOINTS**

Filtering is available to specify blackout dates from which fares should not be returned. This is enabled for one, or many, specific departure range(s).

All endpoints.

## Weekdays **ALL ENDPOINTS**

Filtering is available to specify flights departing on specific days of the week.

All endpoints.

## Duration **ALL ENDPOINTS**

Filtering is available for the trip duration, based on the total number of days between the departure and return dates. The filter allows you to apply a minimum and/or maximum number of days.

All endpoints.

## Histogram Type **LIMITED**

Allows to determine if the iteration over dates is based on the Outbound or Inbound dates. The histogram will iterate based on a price per day on Outbound or Inbound dates. Ideally, when using INBOUND, the departure date would be limited to a range of one day, so as to provide information on round-trip fares based on a specific departure.

Limited, only Histogram Distribution

### ✓ Trip Type Filtering

## Journey Type **ALL ENDPOINTS**

Filtering is available to specify the Journey Type. The value is filtered at the journey level, and will return fares that are One Way or Round trip. For Each Way, the fare returned will be Round Trip, but the total price will be halved. For Combined, the fares will be returned based on Domestic and International setup in the Tenant MS. Rule-based returns fares depending on the journey type tagged for each route.

All endpoints.

## Flight Type **ALL ENDPOINTS**

Filtering is available to return fares for Domestic or International flights only.

All endpoints.

### ✓ Tenant Filtering

## Tenant Codes **ALL ENDPOINTS**

Filtering is available for requesting a subset of tenants belonging to an Alliance. This is so that a subgroup of the Alliance can be selected.

All endpoints.

### ✓ Fare Validation

## IBE Site Edition **ALL ENDPOINTS**

Filtering is available for returning fares collected from the same market, or language+market combo of the customer's IBE. Option to send full site edition code, or country-market code of the site edition, *example "en-us" or "us"*. The site edition refers to the Site Edition that the Fare was collected from in the customer's IBE.

All endpoints.

## Data Expiration **ALL ENDPOINTS**

Filtering is available to returns fares that have been searched within the window of time specified. Allows numbers from 0-9, followed by minute (m), hours (h), or days (d), *example "2d"*.

All endpoints.

## FORMATTING + OUTPUT

#### ▼ Locations + Images

### Language **ALL ENDPOINTS**

Allows to configure the language for any location information returned.

All endpoints.

### Hierarchy **ALL ENDPOINTS**

The system returns the hierarchy specified in the output fields (*see below*).

All endpoints.

### Images **ALL ENDPOINTS**

The system returns the images for each hierarchy-level specified in the output fields (*see below*). The image URLs are aggregated from the TRFX DB. Synchronization occurs once per day.

All endpoints.

#### ▼ Currency + Date Formatting

### Price **ALL ENDPOINTS**

Allows to configure the format for the prices returned. This can be specified overall, or currency by currency. The options include: decimal separator; thousand separator; decimal places; and round type.

For the round type, all customers use "Ceiling" since this is legally required for marketing fares. The Sputnik always returns the shortened price for use by the front-end.

All endpoints.

### Date **ALL ENDPOINTS**

Allows to configure the date pattern that will be returned for the formatted date.

All endpoints.

### Mix Results **LIMITED**

The output can include Mixed Results which allows to define whether the output of fares should be one complete list when requesting as *true*, allowing you to sort across selections. However, if the results should be maintained in order of the selections provided (*how the CPW currently works*) then the *mixedResults* should be set to *false*.

Limited, only Route Search

#### ▼ Pricing Bucket Information

### Price Stats **LIMITED**

It provides Price Stats such as: if the response has mixed currencies; the lowest price in the response; the highest price in the response; the average price in the response. The price stats are based on the results of the Histogram Distribution response.

Limited, only Histogram Distribution

### Price Buckets **LIMITED**

It provides Price Bucket information for the results of the Histogram Distribution response, and compares it to the `/price-stats` endpoint to get a whole year worth of pricing data for the route.

The `priceBuckets` can be used to return all fare information categorized into 3 buckets to essentially note if the price is low, medium, or high for the route specified and over the next year.

Limited, only Histogram Distribution

#### ▼ Response Limits

### Fare Limit **ALL ENDPOINTS**

Creates an alternated list of the top results based on the total number of fares determined in this field. If more are specified in the other fields, faresLimit will limit the number. If less are listed, only those will be returned. For the Route Search, this can be applied at the level of each selection and/or for the overall request.

*Please note that the translated Grouped Routes output does not alternate the list of results.*

All endpoints.

## Route Limit **ALL ENDPOINTS**

Provides the number of routes the service should return. For the Route Search, this can be applied at the level of each selection and/or for the overall request.

All endpoints.

## Fares per Route Limit **ALL ENDPOINTS**

Provides the number of fares per route the service should return. For the Route Search, this can be applied at the level of each selection and/or for the overall request.

All endpoints.

## Destinations Limit **LIMITED**

Provides the number of total destinations that should be returned. Fares are grouped within each destinations

Limited, only Trending Destinations

### Output Fields

Output Fields can include anything from the FareNet Document. Recently included the option to request: "originMarket" and "destinationMarket".

## SORTING

### Fare Sorting

By default, if no sorting options (fareSorting or routeSorting) are sent, the aggregation of the data that is retrieved is based on the default, but the final output will be an alternated list. The alternated list is created by iterating through the routes and selecting fares from each, until the "faresLimit" is reached.

*Please note that the Sputnik V2 Grouped Routes output does not alternate the final list of results.*

fareSorting can be used to determine how fares are retrieved, and later how they are sorted.

The order by which objects are sent within the array, will determine the importance by which the information is aggregated to retrieve fares within each route.

Once that information is sent back, the final sorting can be determined by using the "weight" integer, which sorts based on the priority (lowest weight) of the objects between "fareSorting" and "routeSorting". If no "weight" is sent, then the first object of the fareSorting array is used to sort the final list.

fareSorting object includes the following:

- priceSpecification.totalPrice: Enumeration
  - ASC
  - DESC
- priceSpecification.usdTotalPrice: Enumeration
  - ASC
  - DESC
- flightDeltaDays: Enumeration
  - ASC
  - DESC
- departureDate: Enumeration
  - ASC
  - DESC
- weight: Integer
  - Each Sorting Element has a "weight" integer available

### Route Sorting

By default, if no sorting options (fareSorting or routeSorting) are sent, the aggregation of the data that is retrieved is based on the default, but the final output will be an alternated list. The alternated list is created by iterating through the routes and selecting fares from each, until the "faresLimit" is reached.

Please note that the Sputnik V2 Grouped Routes output does not alternate the final list of results.

routeSorting can be used to determine how routes are retrieved, and later how they are sorted. As of now, only one object should be sent within the array: popularity or airIndex.

Once that information is sent back, the final sorting can be determined by using the "weight" integer, which sorts based on the priority (lowest weight) of the objects between "fareSorting" and "routeSorting". If no "weight" is sent, then the first object of the fareSorting array is used to sort the final list.

routeSorting object includes the following fields:

- popularity: Enumeration
  - ASC
  - DESC
- airIndex: Enumeration
  - ASC
  - DESC
- weight: Integer
  - Each Sorting Element has a "weight" integer available

## Global Filters

**i** Global Filters that are available to be sent in the request across all [Sputnik V3](#) endpoints can be found below.

Please see [EM Platform: Basic Postman Setup](#) for the basic EM Platform Collection.

## Smart Settings

**i** The Smart Settings enable automatically retrieving Tenant settings for the Language / Market combination returned. This filter should always be applied to avoid business logic in the front-end.

Key	Value Type	Description	Example + Default
autoSettings	Object with <ul style="list-style-type: none"> <li>• language string               <ul style="list-style-type: none"> <li>• <i>Required</i></li> </ul> </li> <li>• market string               <ul style="list-style-type: none"> <li>• <i>Required only when market in context.</i></li> </ul> </li> </ul> <i>Optional</i>	Required for Smart Service. Allows Sputnik API to retrieve Tenant Settings and apply required business logic.  In order to ensure that Smart Sputnik / Globetrotter is enabled, the payload must include the autoSettings object. The market is optional, and should not be sent when no market is available via context. See example:  <b>Automated Settings:</b> <ul style="list-style-type: none"> <li>• <b>Date Format:</b> format required for specific Site Edition.</li> <li>• <b>Output Currencies:</b> currencies required for specific Site Edition.</li> <li>• <b>Currency Format:</b> currency format for specific Site Edition.</li> <li>• <b>Data Expiration:</b> data expiration required for specific Site Edition.</li> <li>• <b>Language:</b> language code required for specific Site Edition.</li> <li>• <b>Sort Order:</b> sort order required for specific tenant.</li> </ul> <i>Any of the automated settings can be overwritten by sending the field in the request.</i>	MARKET IN CONTEXT  <pre>"autoSettings": {   "language":   "en",   "market": "US" }</pre> NO MARKET IN CONTEXT  <pre>"autoSettings": {   "language": "en" }</pre> No default
autoSettings.language	String  <i>Required when sending autoSettings</i>	Should be sent as the Language that is returned within the site edition context of the page.	

autoSettings. market	String <i>Optional</i>	Should be sent as the Market that is returned within the site edition context of the page. If no market is returned, then the market string should not be sent.
-------------------------	---------------------------	---

### Route Attribute Filtering

**i** The below filters include restrictions for the information tagged within a specific document collected for a route. These can limit to only return data with the below included. *Please note that for the histogram endpoint, the originMarkets; destinationMarkets; origins; and destinations filters do not need to be used since the Route is required.*

Key	Value Type	Description	Example + Default
originMarkets	Array of strings <i>Optional</i>	Two letter country-market code, <i>example "US"</i> . Filters based on the Country Market of the Origin Airport / Bus Code.	<pre>{   "originMarkets":   [     "US",     "FR"   ] }</pre> <p><i>No default</i></p>
destinationMarkets	Array of strings <i>Optional</i>	Two letter country-market code, <i>example "US"</i> . Filters based on the Country Market of the Destination Airport / Bus Code.	<pre>{   "destinationMarkets":   [     "US",     "FR"   ] }</pre> <p><i>No default</i></p>
origins	Array of strings <i>Optional</i>	Airport / Bus Station Code, <i>example "MIA"</i> . Filters based on the Origin Codes.	<pre>{   "origins" :   [ "MIA", "NYC" ] }</pre> <p><i>No default</i></p>
destinations	Array of strings <i>Optional</i>	Airport / Bus Station Code, <i>example "MIA"</i> . Filters based on the Destination Codes.	<pre>{   "destinations" :   [ "ORD", "LAX" ] }</pre>



			No default
flightType	<p>Enumeration</p> <ul style="list-style-type: none"> <li>• DOMESTIC</li> <li>• INTERNATIONAL</li> </ul> <p>Optional</p>	Filters for Domestic or International fares only. Based on the departure.	<pre>{   "flightType" :   "INTERNATIONAL" }</pre>
			No default
journeyType	<p>Enumeration</p> <ul style="list-style-type: none"> <li>• ONE_WAY</li> <li>• ROUND_TRIP</li> <li>• EACH_WAY</li> <li>• COMBINED</li> </ul> <p>Optional</p>	Journey Type can only include one option, <i>example "ONE_WAY"</i> . The value is filtered at the journey level, and will return fares that are One Way or Round trip. For Each Way, the fare returned will be Round Trip, but the total price will be halved. For Combined, the fares will be returned based on Domestic and International setup in the Tenant MS.	<pre>{   "journeyType" :   "COMBINED" }</pre>
			No default
travelClasses	<p>Array of enumeration</p> <ul style="list-style-type: none"> <li>• ECONOMY</li> <li>• PREMIUM_ECONOMY</li> <li>• BUSINESS</li> <li>• FIRST</li> </ul> <p>Optional</p>	Travel Classes refer to Farenet Travel Classes. The filter allows you to filter by none, one, or many and returns fares within the listed flight classes.	<pre>{   "travelClasses" :   ["ECONOMY",   "BUSINESS"] }</pre>
			No default
brandedFareClasses	<p>Array of strings</p> <p>Optional</p>	<p>The Branded Fare Class filter allows filtering by the customer's overall Branded Travel Class. The filter allows you to filter by none, one, or many and returns fares within the listed branded fare classes.</p> <p>If using this filter in conjunction with the <code>travelClasses</code> filter, you may not see fares returned when a Branded Fare Class belongs to a different Travel Class than the one requested. It is best to use each filter separately.</p> <p>Please note, that <code>brandedFareClass</code> should be added as an output field when using this filter. Only customers with the Branded Travel Classes feature will be able to utilize this filter. Please see the Organization Service: Central Tenant Settings for more info.</p>	<pre>{   "brandedFareClasses" :   [     "ECO-PREMIUM"   ] }</pre>
			No default
ticketType	<p>Object with two Booleans and a string array</p> <p>Optional</p>	<p>Review documentation - T/F/null</p> <p><code>ticketType</code> object includes the following fields:</p> <ul style="list-style-type: none"> <li>• <code>codeshare</code>: Boolean</li> <li>• <code>interline</code>: Boolean</li> <li>• <code>bookingClass</code>: String array</li> </ul>	<pre>{   "isCodeshare" :   false,   "isInterline" :   false,   "bookingClass" :</pre>

			<pre>[ "Y", "H", "K", "L",   "M", "V" ] }</pre>
			<i>No default</i>
flightStopCount	Number Range <i>Optional</i>	Flight Stop Count filters based on a minimum and / or a maximum number of stops. This applies to flightStopCount of the overall document, which is the maximum number of stops between outbound and inbound.	<pre>{   "flightStopCount" : {   "minimum" : 0,   "maximum" : 2 } }</pre>
			<i>No default</i>
routeTypes	Array of enumeration <ul style="list-style-type: none"> <li>CONNECTING</li> <li>DIRECT</li> <li>NONSTOP</li> </ul> <i>Optional</i>	Filter allows you to request fares that are a specific route type, such as nonstop, direct, and / or connecting.	<pre>{   "routeTypes" : [ "NONSTOP", "DIRECT" ] }</pre>
			<i>No default</i>
trfxRoutes	Boolean <i>Optional</i>	Filter will only include fares that are related to routes in the TRFX Route List.	<pre>{   "trfxRoutes" : true }</pre>
			<i>Default: true</i>

## Date-based Filtering

**i** The below filters include restrictions for specific dates, types of days, or length of stay.

Key	Value Type	Description	Example + Default
departureDaysInterval	Object with: <ul style="list-style-type: none"> <li>start integer</li> <li>end integer</li> </ul>	<p>The departureDaysInterval is an interval setting for the start and end, that calculate the actual date dynamically based on the day the request is sent in. The settings will filters fares falling within the range specified. The start must be at least 0, and the end cannot exceed 500.</p> <p>The departureDaysInterval cannot be included if departure is included.</p>	<pre>"departureDaysInterval": {   "start": 0,   "end": 90 }</pre>
			<i>No default</i>
returnDaysInterval	Object with:	The returnDaysInterval is an interval setting for the start and end, that calculate	

	<ul style="list-style-type: none"> <li>• start integer</li> <li>• end integer</li> </ul>	<p>the actual date dynamically based on the day the request is sent in. The settings will filter fares falling within the range specified. The start must be at least 0, and the end cannot exceed 500.</p> <p>The <code>returnDaysInterval</code> cannot be included if <code>return</code> is included.</p>	<pre>"returnDaysInterval": {   "start": 7,   "end": 97 }</pre>	<i>No default</i>
departure	<p>Date Range</p> <p><i>Optional</i></p>	<p>Date Range including start and end. Start is Required, but End is not. Filters fares falling within the departure range specified.</p> <p>The <code>departure</code> cannot be included if <code>departureDaysInterval</code> is included.</p>	<pre>{   "departure" : {     "start" : "2020-03-01",     "end" : "2020-06-01"   } }</pre>	<i>No default</i>
return	<p>Date Range</p> <p><i>Optional</i></p>	<p>Date Range including start and end. Start is Required, but End is not. Filters fares falling within the return range specified.</p> <p>The <code>return</code> cannot be included if <code>returnDaysInterval</code> is included.</p>	<pre>{   "return" : {     "start" : "2020-03-01",     "end" : "2020-06-01"   } }</pre>	<i>No default</i>
weekDays	<p>Array of enumeration</p> <ul style="list-style-type: none"> <li>• MON</li> <li>• TUE</li> <li>• WED</li> <li>• THU</li> <li>• FRI</li> <li>• SAT</li> <li>• SUN</li> </ul> <p><i>Optional</i></p>	<p>Week Days allows to include only flights departing on specific days of the week.</p>	<pre>{   "weekDays" :   ["MON", "WED", "FRI"] }</pre>	<i>No default</i>
blackoutDates	<p>Array of Date Range</p> <p><i>Optional</i></p>	<p>Blackout Date filter fares out that fall within a specific departure range(s).</p>	<pre>{   "blackoutDates" :   [     {       "start" :       "2020-03-05",       "end" : "2020-</pre>	

			<pre> 03-10"     },     {       "start" : "2020-02-15",       "end" : "2020- 02-15"     }   ] } </pre>
			<i>No default</i>
tripDuration	Number Range <i>Optional</i>	Trip Duration filters based on the total number of days between the departure and return dates. The filter allows you to apply a minimum and / or maximum.	<pre> {   "tripDuration" : {     "minimum" : 2,     "maximum" : 5   } } </pre>
			<i>No default</i>

### Category + Tags and Geo ID Filtering

**i** Applies location hierarchy filters and / or Category and Tag filters to the results displayed. The results may be limited, or never match, when used with additional location filtering that is in opposition, such as: list of Property Codes; Geo-coordinate search; and Market.

Key	Value Type	Description	Example + Default
originHierarchy	Object <i>Optional</i>	Filters based on the hierarchy of a property.	<pre> {   "originHierarchy" : {   "categories": [],   "tags": [],   "airport": {},   "city": {},   "state": {},   "country" : {},   "regions": {} } } </pre>
			<i>No default</i>
destinationHierarchy		Filters based on the destination hierarchy of a route.	<pre> { </pre>

			<pre>"destinationHierarchy" : {   "categories": [],   "tags": [],   "airport": {},   "city": {},   "state": {},   "country" : {},   "regions": {} }</pre>
			<i>No default</i>

The below keys are available within both `originHierarchy` and `destinationHierarchy`. The examples provided are for `originHierarchy`.

<code>originHierarchy.categories</code>	<p>Array</p> <p><i>Optional</i></p>	<p>Applies Category filter at any level of the hierarchy. Will return documents where any level of the hierarchy has the category (ies) assigned.</p>	<pre>"originHierarchy": {   "categories": [     "cat_1"   ] }</pre>
			<i>No default</i>

<code>originHierarchy.tags</code>	<p>Array</p> <p><i>Optional</i></p>	<p>Applies Tag filter at any level of the hierarchy. Will return documents where any level of the hierarchy has the tag(s) assigned.</p>	<pre>"originHierarchy": {   "tags": [     "tag_1"   ] }</pre>
			<i>No default</i>

<code>originHierarchy.airport</code>	<p>Object with</p> <ul style="list-style-type: none"> <li>• <code>geoId</code>: Array of strings</li> <li>• <code>categories</code>: Array</li> <li>• <code>tags</code>: Array</li> </ul> <p><i>Optional</i></p>	<p>Filters based on the specific Geo ID of an airport.</p> <p>The <code>categories</code> and <code>tags</code> arrays can be nested within the <code>airports</code> object, which allows to search based on Categories / Tags being matched at the property level.</p>	<pre>{   "originHierarchy" : {     "airport" : {       "geoId" : ["111222"],       "categories": [         "category_1",         "category_2"       ],       "tags": [         "tag_21",         "tag_22"       ]     }   } }</pre>
--------------------------------------	--	--	---

			<pre> } } </pre>
			No default
originHierarchy.city	<p>Object with</p> <ul style="list-style-type: none"> <li>• geoId: Array of strings</li> <li>• categories: Array</li> <li>• tags: Array</li> </ul> <p><i>Optional</i></p>	<p>Filters based on the specific Geo ID of a city, returning documents where properties belong to the city.</p> <hr/> <p>The categories and tags arrays can be nested within the city object, which allows to search based on Categories / Tags being matched at the city level.</p>	<pre> {   "originHierarchy" : {   "city" : {     "geoId" : ["111222"],     "categories": [       "category_1",       "category_2"     ],     "tags": [       "tag_21",       "tag_22"     ]   } } } </pre>
			No default
originHierarchy.state	<p>Object with</p> <ul style="list-style-type: none"> <li>• geoId: Array of strings</li> <li>• categories: Array</li> <li>• tags: Array</li> </ul> <p><i>Optional</i></p>	<p>Filters based on the specific Geo ID of a state, returning documents where properties belong to the state.</p> <hr/> <p>The categories and tags arrays can be nested within the state object, which allows to search based on Categories / Tags being matched at the state level.</p>	<pre> {   "originHierarchy" : {   "state" : {     "geoId" : ["111222"],     "categories": [       "category_1",       "category_2"     ],     "tags": [       "tag_21",       "tag_22"     ]   } } } </pre>
			No default
originHierarchy.country	<p>Object with</p> <ul style="list-style-type: none"> <li>• geoId: Array of strings</li> </ul>	<p>Filters based on the specific Geo ID of a country, returning documents where properties belong to the country.</p>	

	<ul style="list-style-type: none"> <li>categories: Array</li> <li>tags: Array</li> </ul> <p><i>Optional</i></p>	<p>The categories and tags arrays can be nested within the <code>country</code> object, which allows to search based on Categories / Tags being matched at the country level.</p>	<pre> {   "originHierarchy" : {   "country" : {     "geoId" : ["111222"],     "categories": [       "category_1",       "category_2"     ],     "tags": [       "tag_21",       "tag_22"     ]   } } </pre> <p><i>No default</i></p>
<p><code>originHierarchy.region</code></p>	<p>Object with</p> <ul style="list-style-type: none"> <li>geoId: Array of strings</li> <li>categories: Array</li> <li>tags: Array</li> </ul> <p><i>Optional</i></p>	<p>Filters based on the specific Geo ID of a region, returning documents where properties belong to the region.</p> <hr/> <p>The categories and tags arrays can be nested within the <code>region</code> object, which allows to search based on Categories / Tags being matched at the region level.</p>	<pre> {   "originHierarchy" : {   "region" : {     "geoId" : ["111222"],     "categories": [       "category_1",       "category_2"     ],     "tags": [       "tag_21",       "tag_22"     ]   } } </pre> <p><i>No default</i></p>

### Distance-based Filtering

**i** Returns documents within a specific distance from a set of coordinates. The results may be limited, or never match, when used with additional location filtering that is in opposition, such as: list of Property Codes; Hierarchy or Category/Tag search; and Market.

Key	Value Type	Description	Example + Default
<code>geoCoordinate</code>	Object	The <code>geoCoordinate</code> object includes two optional	

- origin
- destination

Optional

attributes (*one must be included when using*) that are origin and destination objects. Each object filter allows to specify to search by a quadrant (*square*) or from a specific point and a radius surrounding it (*radius*). Best practice would be to use either quadrant or radial search, not both.

```
{
  "geoCoordinate": {
    "origin": {
      "square": {
        },
        "radius": { }
      },
    "destination": {
      "square": {
        },
        "radius": { }
      }
    }
  }
```

No default

The below examples for `square` and `radius` are available within both `origin` and `destination`. There is an example for each type of object.

`geoCoordinate.origin.square`

Object with properties

- `topLeft`: Required object with
  - `latitude`: integer
  - `longitude`: integer
- `bottomRight`: Required object with
  - `latitude`: integer
  - `longitude`: integer

Filters based on geo-coordinates within a quadrant. When used, both `topLeft` and `bottomRight` objects with `latitude` and `longitude` are needed.

```
{
  "geoCoordinate": {
    "origin": {
      "square": {
        "topLeft": {
          "latitude": 111.11,
          "longitude": 222.33
        },
        "bottomRight": {
          "latitude": 111.11,
          "longitude": 222.33
        }
      }
    }
  }
```

No default



<p>geoCoordinate. destination.radius</p>	<p>Object with properties</p> <ul style="list-style-type: none"> <li>point: Required object with properties <ul style="list-style-type: none"> <li>latitude</li> <li>longitude</li> </ul> </li> <li>radius: Integer (<i>optional</i>)</li> </ul>	<p>Filters based on geo-coordinates from a specific point and a radius surrounding it . When used, at least the point object with latitude and longitude are required.</p>	<pre> {   "geoCoordinate": {     "destination": {       "radius": {         "point": {           "latitude": 111.11,            "longitude": 222.33         },         "radius": 40       }     }   } } </pre> <p>No default</p>
--	--	--	--

### Output Restrictions and Format

**i** The following are a number of filters that allow to restrict data, as well as some inputs allowing for the formatting of data. All of the below are completely optional when using the Smart autoSettings, as these will be retrieved from the Tenant's configurations.

Key	Value Type	Description	Example + Default
<p>dataExpirationWindow</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>String, with a specific format</p> <p><i>Optional</i></p>	<p>Allows numbers from 0-9, followed by minute (m), hours (h), or days (d), <i>example "2d"</i>.</p> <p>Returns fares that have been searched within the window of time specified.</p> <p><i>Customers with Index Expiration false should not configure this.</i></p>	<pre> {   "dataExpirationWindow" : "1d" } </pre> <p>No default</p>
<p>fareSorting</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>Array of Objects</p> <ul style="list-style-type: none"> <li>priceSpecification.totalPrice: Enumeration <ul style="list-style-type: none"> <li>ASC</li> <li>DESC</li> </ul> </li> <li>priceSpecification.usdTotalPrice: Enumeration <ul style="list-style-type: none"> <li>ASC</li> <li>DESC</li> </ul> </li> <li>flightDeltaDays: Enumeration</li> </ul>	<p>By default, if no sorting options (fareSorting or routeSorting) are sent, the aggregation of the data that is retrieved is based on the default, but the final output will be an alternated list. The alternated list is created by iterating through the routes and selecting fares from each, until the "faresLimit" is reached.</p> <p>fareSorting can be used to determine how fares are retrieved, and later how they are sorted. The order by which objects are sent within the array, will determine the importance by which the information is aggregated to retrieve fares within each route.</p> <p>Once that information is sent back, the final sorting can be determined by using the "weight" integer, which sorts based on the priority (lowest weight) of</p>	<pre> {   "fareSorting" : [   {     "priceSpecification.totalPrice" : "ASC",     "weight" : 3   }, </pre>

- ASC
- DESC
- departureDate: Enumeration
  - ASC
  - DESC
- weight: Integer
  - Each Sorting Element has a "weight" integer available

Optional

the objects between "fareSorting" and "routeSorting". If no "weight" is sent, then the first object of the fareSorting array is used to sort the final list.

```
{
  "departureDate" :
  "DESC",
  "weight":1
}
```

**Default**

```
{
  "fareSorting" :
  [
    {
      "priceSpecification
      .totalPrice" :
      "ASC"
    }
  ]
}
```

routeSorting

**SMART SETTING - NOT NEEDED**

- Array of Objects
- popularity: Enumeration
    - ASC
    - DESC
  - airIndex: Enumeration
    - ASC
    - DESC
  - weight: Integer
    - Each Sorting Element has a "weight" integer available

Optional

By default, if no sorting options (fareSorting or routeSorting) are sent, the aggregation of the data that is retrieved is based on the default, but the final output will be an alternated list. The alternated list is created by iterating through the routes and selecting fares from each, until the "faresLimit" is reached.

routeSorting can be used to determine how routes are retrieved, and later how they are sorted. As of now, only one object should be sent within the array: popularity or airIndex.

Once that information is sent back, the final sorting can be determined by using the "weight" integer, which sorts based on the priority (lowest weight) of the objects between "fareSorting" and "routeSorting". If no "weight" is sent, then the first object of the fareSorting array is used to sort the final list.

```
{
  "routeSorting" :
  [
    {
      "popularity"
      : "DESC",
      "weight":2
    }
  ]
}
```

**Default**

```
{
  "routeSorting" :
  [
    {
      "popularity"
      : "DESC"
    }
  ]
}
```

			<pre> ] } </pre>
<p>currencies</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>Array of strings</p> <p><i>Optional</i></p>	<p>Three letter currency code, <i>example "USD"</i>. Filters based on the Currency Code(s) that are sent.</p>	<pre> {   "currencies" :   [ "USD" ] } </pre> <p><i>No default</i></p>
<p>outputFormat</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>Object</p> <p><i>Optional</i></p>	<p>Allows to configure a number of formats for the output returned by the system.</p>	<pre> {   "outputFormat": {     "currencyCode":     "USD",     "datePattern":     "MM/dd/yyyy",     "languageCode":     "en",     "currencySettings":     [       {         "currencyCode":         "USD",         "price":         {           "decimalSeparator":           ".",           "thousandSeparator":           ",",           "decimalPlaces": 0         }       }     ]   } } </pre> <p><b>Default</b></p>

			<pre> {   "outputFormat": {     "price": {  "decimalSeparator": "." ,  "thousandSeparator" : ", " ,  "decimalPlaces": 0     }   } } </pre>
<p>outputFormat. currencyPricing Settings</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>Array of Objects</p>	<p><i>By default the system rounds fares up, when decimal places are 0. The shortenedFormattedPrice is always sent.</i></p> <p>currencyPricingSettings object includes the following fields:</p> <ul style="list-style-type: none"> <li>• currencyCode: String</li> <li>• priceFormat: Object</li> </ul> <p>priceFormat object includes the following fields:</p> <ul style="list-style-type: none"> <li>• decimalSeparator: String</li> <li>• thousandSeparator: String</li> <li>• decimalPlaces: Integer</li> <li>• shortenedFormattedPrice: Boolean</li> </ul> <p><i>Optional</i></p>	<pre> {   "outputFormat": {  "currencyPricingSet tings": [     {  "currencyCode": "USD" ,  "priceFormat": {  "decimalSeparator": "." ,  "thousandSeparator" : ", " ,  "decimalPlaces": 0 ,  "roundType": "CEIL"     }   } ] } } </pre>
<p>outputFormat. languageCode</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>String</p> <p><i>Optional</i></p>	<p>Allows to configure the language for any location information returned.</p>	<pre> {   "outputFormat": { </pre>

			<pre> "languageCode" : "en" } } </pre> <p><i>No default</i></p>
<p>outputFormat. datePattern</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>String</p> <p><i>Optional</i></p>	<p>Allows to configure the date pattern that will be returned for the formatted date.</p>	<pre> {   "outputFormat" :   {     "datePattern" :     "MM/dd/yyyy"   } } </pre> <p><i>No default</i></p>
<p>outputFormat. price</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>Object</p> <p><i>Optional</i></p>	<p>Allows to configure the format for the prices returned. This will apply to any currency, unless otherwise specified in the <code>currencyPricingSettings</code>.</p> <p><code>price</code> object includes the following fields:</p> <ul style="list-style-type: none"> <li>• <code>decimalSeparator</code>: String</li> <li>• <code>thousandSeparator</code>: String</li> <li>• <code>decimalPlaces</code>: Integer</li> <li>• <code>roundType</code>: Enumeration <ul style="list-style-type: none"> <li>• CEIL</li> <li>• FLOOR</li> </ul> </li> <li>• <code>shortenedFormattedPrice</code>: Boolean</li> </ul>	<pre> {   "outputFormat" : {     "price" : {  "decimalSeparator" : "." ,  "thousandSeparator" : ", " ,  "decimalPlaces" : 0 } } } </pre> <p><b>Default</b></p> <pre> {   "outputFormat" : {     "price" : {       "roundType" : "CEIL" } } } </pre>

<p>siteEdition</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>String</p> <p><i>Optional</i></p>	<p>Option to send full site edition code, or country-market code of the site edition, <i>example "en-us" or "us"</i>. The site edition refers to the Site Edition that the Fare was collected from in the customer's IBE. Filters fares collected in the site edition, or site edition market, sent.</p>	<pre>{   "siteEdition" :   "us" }</pre> <p><i>No default</i></p>
<p>outputFields</p> <p><b>SMART SETTING - NOT NEEDED</b></p>	<p>Array of strings</p> <p><i>Optional</i></p>	<p>Output Fields can include anything from the Farenet Document and Locations Service.</p> <p>The endpoints will have specific defaults that return the required base level of information.</p>	<pre>{   "outputFields" :   [   ] }</pre> <p><i>Example provided is the default sent by the system.</i></p>

### Price Filtering

Key	Value Type	Description	Example + Default
<p>requestedTenants</p>	<p>Array of strings</p> <p><i>Optional</i></p>	<p>When requesting to an Alliance Tenant, the filter allows you to select specific tenant codes (only belonging to the alliance being requested) so that you can select a subgroup.</p>	<pre>{   "requestedTenants"   : [ "ua", "cx",     "cm" ] }</pre> <p><i>No default</i></p>
<p>redemptionUnit</p>	<p>Enumeration</p> <ul style="list-style-type: none"> <li>• MILES</li> <li>• POINTS</li> </ul> <p><i>Optional</i></p>	<p>Enables requesting different types of redemption fares, and will include additional information in the output.</p>	<pre>{   "redemptionUnit"   : "MILES" }</pre> <p><i>No default</i></p>
<p>budget</p>	<p>Object</p> <p><i>Optional</i></p>	<ul style="list-style-type: none"> <li>• Budget filters based on a minimum and / or a maximum budget. Each is optional, but at least one is required when the budget object is used. Also, we can specify the reference price for the budget through the source parameter: <ul style="list-style-type: none"> <li>• minimum - Integer that determines minimum budget</li> <li>• maximum - Integer that determines maximum budget</li> <li>• source - Enumeration (TOTAL_USD_PRICE    TOTAL_PRICE) that determines reference price for budget</li> </ul> </li> </ul>	<pre>{   "budget" : {     "minimum" :     15,     "maximum" :     500,     "source" :     "TOTAL_USD_PRICE"   } }</pre>

- When `source` is not sent or sent as `null`, this applies to `totalUsdPrice` when no currency is sent, and applies to `totalPrice` when a currency is sent.
- If `source` is sent as empty string, this should be considered a bad request

```

}
}

```

*No default*

## Search

**i** The search endpoint returns flight fares for a specific Tenant based on a number of filtering criteria. Please see [Sputnik Translator V3](#) for all Translator endpoints.

Please see [EM Platform: Basic Postman Setup](#) for the basic EM Platform Collection.

## Example Sample Request:

```

{
  "autoSettings": {
    "language": "en",
    "market": "us"
  },
  "origins": [],
  "destinations": [],
  "journeyType": "ONE_WAY",
  "departureDaysInterval": {
    "start": 0,
    "end": 365
  },
  "faresPerRoute": 1,
  "routesLimit": 1,
  "faresLimit": 1
}

```

## Search Endpoint Request Parameters

Please see: [Sputnik: Global Filters \(all endpoints\)](#), for all filtering options. The `autoSettings` should always be sent to avoid business logic being requested at the front-end level.

Key	Value Type	Description	Example + Default
<code>routesLimit</code>	Integer <i>Optional</i>	Provides the number of routes the service should return. <ul style="list-style-type: none"> <li>• Maximum is 100</li> </ul>	<pre> {   "routesLimit" : 1 } </pre> <p><i>Example provided is the default sent by the system.</i></p>
<code>faresPerRoute</code>	Integer <i>Optional</i>	Provides the number of fares per route the service should return. <ul style="list-style-type: none"> <li>• Maximum is 100</li> </ul>	

			<pre>{   "faresPerRoute" : 1 }</pre>
			<i>Example provided is the default sent by the system.</i>
faresLimit	Integer <i>Optional</i>	<p>Creates an alternated list of the top results based on the total number of fares determined in this field. If more are specified in the other fields, faresLimit will limit the number. If less are listed, than only those will be returned.</p> <p><i>Please note that the Sputnik V2 Grouped Routes output does not alternate the list of results.</i></p> <ul style="list-style-type: none"> <li>• Maximum is 100</li> </ul>	<pre>{   "faresLimit" : 1 }</pre>
			<i>Example provided is the default sent by the system.</i>
outputFields	Array of strings <i>Optional</i>	<p>Output Fields can include anything from the Farenet Document and Locations Service.</p> <p>The endpoints will have specific defaults that return the required base level of information.</p>	<pre>{   "outputFields" : [     "priceSpecification",     "outboundFlight.fareClass",     "inboundFlight.fareClass",     "outboundFlight.fareClassInput",     "inboundFlight.fareClassInput",     "datacoreId",     "market",     "outboundFlight.departureAirportIataCode",     "outboundFlight.arrivalAirportIataCode",     "searchDate",     "formattedDepartureDate",     "formattedReturnDate"   ] }</pre>

**SMART SETTING - NOT NEEDED**



```

"departureDate",
    "returnDate",

"journeyType",
    "flightType",
    "origin.
airport.name",
    "destination.
airport.name",
    "origin.city.
name",
    "destination.
city.name",
    "origin.
country.name",
    "destination.
country.name",
    "destination.
city.image",
    "destination.
country.image",
    "origin.
airport.
prepositions",
    "destination.
airport.
prepositions",
    "origin.city.
prepositions",
    "destination.
city.prepositions",
    "origin.
country.
prepositions",
    "destination.
country.prepositions"
]
}

```

*Example provided is the default sent by the system.*

## Route Search

**i** Returns flight fares for a specific Tenant based on a number of filtering criteria for one, or many, sets of routes. The response can group together all the information, or return it separated by route request within the body.

Please see [EM Platform: Basic Postman Setup](#) for the basic EM Platform Collection.

### Quick Links

- [Example Sample Request](#)
- [Route Search Endpoint Request Parameters](#)

- [selections](#)
- [selections.routes](#)
- [selections.hierarchies](#)
- [selections.faresPerRoute](#)
- [selections.routesLimit](#)
- [selections.faresLimit](#)
- [selections.id](#)
- [blacklistedRoutes](#)
- [tripDuration](#)
- [outputFormat.mixedResults](#)
- [routesLimit](#)
- [faresPerRoute](#)
- [faresLimit](#)
- [Query Outputs](#)
  - [Single List](#)
  - [Ordered by Selection](#)
- [cURL Examples](#)

### Example Sample Request:

```
{
  "autoSettings": {
    "language": "en",
    "market": "us"
  },
  "travelClasses": [
    "ECONOMY",
    "BUSINESS"
  ],
  "journeyType": "ROUND_TRIP",
  "departure": {
    "start": "2022-02-10",
    "end": "2022-03-11"
  },
  "selections": [
    {
      "routes": ["DFWMIA", "MIALAX", "LAXMIA"],
      "faresPerRoute": 1,
      "routesLimit": 36,
      "faresLimit": 36,
      "id": 1
    }
  ]
}
```

### Route Search Endpoint Request Parameters

Please see: [Sputnik: Global Filters \(all endpoints\)](#), for all filtering options. The `autoSettings` should always be sent to avoid business logic being requested at the front-end level.

Key	Value Type	Description	Example + Default
<code>selections</code>	Array of objects <i>Required</i>	Allows to request a number of routes and settings per route selection to return fares for each.	

			<pre> {   "selections": [     {       "routes": ["AAABBB", "CCDDDD", "EEEEFF"],       "faresPerRoute" : 1,       "routesLimit" : 1,       "faresLimit" : 1,       "id": "123"     },     {       "routes": ["GGGHHH", "HHHHIII", "IIIJJJ", "JJJKKK"],       "faresPerRoute" : 1,       "routesLimit" : 1,       "faresLimit" : 1,       "id": "4567"     },     {       "routes":["KKKLLL", "LLLMMM", "MMNNN", "NNNOOO", "OOOPPP"],       "faresPerRoute" : 1,       "routesLimit" : 1,       "faresLimit" : 1,       "id": "891011"     }   ] } </pre>
<b>selections.routes</b>	Array of strings <i>Required</i>	Allows specifying the routes that are requested for the selection. Routes is required, unless the hierarchies array is passed.	<p style="text-align: center;"><i>No default</i></p> <pre> {   "selections": [     {       "routes": [         "AAABBB",         "CCDDDD",         "EEEEFF"       ]     }   ] } </pre>
<b>selections.hierarchies</b>	Array of objects	Hierarchies is required, unless the routes array is passed.	<p style="text-align: center;"><i>No default</i></p>

		<p>Applies Category / Tag filter to the Origin or Destination within a selection. This can also be used with <code>geoId</code>.</p> <p>Includes <code>categories</code> and <code>tags</code> arrays. Both are optional, but at least one should be defined. If included directly in the <code>origin</code> or <code>destination</code> object then it can apply at any level of the hierarchy.</p> <p>The <code>categories</code> and <code>tags</code> arrays can be nested within specific hierarchy objects, which allows to search based on Categories / Tags being matched at the specific hierarchy level. This is included within the <code>origin</code> or <code>destination</code> object:</p> <ul style="list-style-type: none"> <li>• airport</li> <li>• city</li> <li>• state</li> <li>• country</li> <li>• region</li> </ul>	<pre> {   "selections": [     {       "hierarchies": [         {           "destination": {             "city": {               "tags": [                 "61153c27826a87226ca4c2b7"               ]             }           }         }       ]     }   ] } </pre>
<code>selections.faresPerRoute</code>	Integer <i>Required</i>	The total number of fares per route allotted for the selection specified.	<pre> {   "selections": [     {       "faresPerRoute" : 1     }   ] } </pre> <p><i>Example provided is the default sent by the system.</i></p>
<code>selections.routesLimit</code>	Integer <i>Required</i>	The total number of routes allotted for the selection specified.	<pre> {   "selections": [     {       "routesLimit" : 1     }   ] } </pre> <p><i>Example provided is the default sent by the system.</i></p>
<code>selections.faresLimit</code>	Integer <i>Required</i>	The total number of fares allotted for the selection specified.	<pre> {   "selections": [     {       "faresLimit" : 1     }   ] } </pre>

			<pre> } } </pre> <p><i>Example provided is the default sent by the system.</i></p>
selections.id	String <i>Required</i>	An ID that references the selection under which fares will be added in the response.	<pre> {   "selections": [     {       "id" : "123"     }   ] } </pre> <p><i>No default</i></p>
blacklistedRoutes	Array of strings <i>Optional</i>	Allows you to select specific routes that should not be returned.	<pre> {   blacklistedRoutes: ["MIAORD", "MADEWR" ] } </pre>
tripDuration	Number Range <i>Optional</i>	Trip Duration filters based on the total number of days between the departure and return dates. The filter allows you to apply a minimum and / or maximum.	<pre> {   "tripDuration" : {     "minimum" : 2,     "maximum" : 5   } } </pre> <p><i>No default</i></p>
outputFormat.mixedResults	boolean <i>Optional</i>	The mixedResults field allows you to define whether the output of fares should be one complete list when requesting as <i>true</i> , allowing you to sort across selections. However, if the results should be maintained in order of the selections provided ( <i>how the CPW currently works</i> ) then the mixedResults should be set to <i>false</i> .	<pre> {   "outputFormat": {     "mixedResults": false   } } </pre> <p><i>Example provided is the default sent by the system.</i></p>
routesLimit	Integer <i>Optional</i>	Provides the number of routes the service should return.	<pre> {   "routesLimit" : 1 } </pre> <p><i>Example provided is the default sent by the system.</i></p>

faresPerRoute	Integer <i>Optional</i>	Provides the number of fares per route the service should return.	<pre>{   "faresPerRoute" : 1 }</pre> <p><i>Example provided is the default sent by the system.</i></p>
faresLimit	Integer <i>Optional</i>	<p>Creates an alternated list of the top results based on the total number of fares determined in this field. If more are specified in the other fields, faresLimit will limit the number. If less are listed, than only those will be returned.</p> <p><i>Please note that the Sputnik V2 Grouped Routes output does not alternate the list of results.</i></p>	<pre>{   "faresLimit" : 1 }</pre> <p><i>Example provided is the default sent by the system.</i></p>

### Query Outputs

List Type	Description	Output Example
Single List	<p>"mixedResults": true</p> <p>Returns a single list of fares ordered by fareSorting and routeSorting specified. This means that the individual selections made are mixed together.</p>	<pre>[ ]</pre> <p><b>Example:</b></p> <pre>[   {     "datacoreId": "16128223332261511.4644     aaic-00001",     "journeyType": "ROUND_TRIP",     "priceSpecification": {       "totalPrice": 80.8,       "currencySymbol": "USD",       "usdTotalPrice": 80.8,       "currencyCode": "USD"     },     "passengerDetails": [       {         "count": 1       }     ],     "searchDate": "2021-02-08T22:12:59.026     +0000",     "departureDate": "2021-03-16",     "airline": {       "iataCode": "UA"     },   }, ]</pre>

```

    "outboundFlight": {
      "departureAirportIataCode": "MIA",
      "arrivalAirportIataCode": "ORD",
      "origin": {
        "airport": {},
        "city": {
          "name": "Miami",
          "image": "https://www.
united.com/en-us/flights/media-em/ua
/5dc5f0b70513c_UA_Dreamliner_New_Livery.png"
        },
        "state": {},
        "country": {
          "name": "United States",
          "image": "https://www.
united.com/cms/MediaBinImages/NewYork_944x360.
jpg"
        },
        "region": {}
      },
      "destination": {
        "airport": {},
        "city": {
          "name": "Chicago",
          "image": "https://www.
united.com/en-us/flights/media-em/ua
/5d9fe716e62e8_Chicago_UA.jpg"
        },
        "state": {},
        "country": {
          "name": "United States",
          "image": "https://www.
united.com/cms/MediaBinImages/NewYork_944x360.
jpg"
        },
        "region": {}
      }
    },
    "formattedDepartureDate": "03/16/21",
    "id": "1"
  }
]

```

**Ordered by Selection**

"mixedResults": false

Returns multiple nested lists of fares that ordered by fareSorting and routeSorting within each selection. This means

```

[
  {
    "id": "1",
    "fares": []
  }
]

```

that the individual selections made are not mixed together.

```
    },
    {
      "id": "2",
      "fares": []
    },
    {
      "id": "3",
      "fares": []
    }
  ]
```

**Example:**

```
[
  {
    "id": "1",
    "fares": [
      {
        "datacoreId":
"16128223332261511.4644aaic-00001",
        "journeyType": "ROUND_TRIP",
        "priceSpecification": {
          "totalPrice": 80.8,
          "currencySymbol": "USD",
          "usdTotalPrice": 80.8,
          "currencyCode": "USD"
        },
        "passengerDetails": [
          {
            "count": 1
          }
        ],
        "searchDate": "2021-02-08T22:12:
59.026+0000",
        "departureDate": "2021-03-16",
        "airline": {
          "iataCode": "UA"
        },
        "outboundFlight": {
          "departureAirportIataCode":
"MIA",
          "arrivalAirportIataCode":
"ORD",
          "origin": {
            "airport": {},
            "city": {
              "name": "Miami",
              "image":
"https://www.united.com/en-us/flights/media-em
```



```

/ua/5dc5f0b70513c-UA_Dreamliner_New_Livery.png"
    },
    "state": {},
    "country": {
        "name": "United
States",
        "image":
"https://www.united.com/cms/MediaBinImages
/NewYork_944x360.jpg"
    },
    "region": {}
},
"destination": {
    "airport": {},
    "city": {
        "name": "Chicago",
        "image":
"https://www.united.com/en-us/flights/media-em
/ua/5d9fe716e62e8_Chicago_UA.jpg"
    },
    "state": {},
    "country": {
        "name": "United
States",
        "image":
"https://www.united.com/cms/MediaBinImages
/NewYork_944x360.jpg"
    },
    "region": {}
}
},
"formattedDepartureDate": "03/16
/21"
    }
]
},
{
    "id": "2",
    "fares": [
        {
            "datacoreId":
"16128189595700501.1453aaic-00002",
            "journeyType": "ROUND_TRIP",
            "priceSpecification": {
                "totalPrice": 96.8,
                "currencySymbol": "USD",
                "usdTotalPrice": 96.8,
                "currencyCode": "USD"
            },
            "passengerDetails": [

```

```
        {
            "count": 1
        }
    ],
    "searchDate": "2021-02-08T21:16:
09.533+0000",
    "departureDate": "2021-05-05",
    "airline": {
        "iataCode": "UA"
    },
    "outboundFlight": {
        "departureAirportIataCode":
"LAX",
        "arrivalAirportIataCode":
"ORD",
        "origin": {
            "airport": {},
            "city": {
                "name": "Los
Angeles",
                "image":
"https://www.united.com/en-us/flights/media-em
/ua/5d9fe73d62c4f_Los_Angeles_UA.jpg"
            },
            "state": {},
            "country": {
                "name": "United
States",
                "image":
"https://www.united.com/cms/MediaBinImages
/NewYork_944x360.jpg"
            },
            "region": {}
        },
        "destination": {
            "airport": {},
            "city": {
                "name": "Chicago",
                "image":
"https://www.united.com/en-us/flights/media-em
/ua/5d9fe716e62e8_Chicago_UA.jpg"
            },
            "state": {},
            "country": {
                "name": "United
States",
                "image":
"https://www.united.com/cms/MediaBinImages
/NewYork_944x360.jpg"
            },
        },
    },
}
```

```

        "region": {}
      }
    },
    "formattedDepartureDate": "05/05
  /21"
}
]
]

```

## cURL Examples

 Please replace the EM-API-Key with the one used by your team.

### ▼ UA - Ordered by Selection

```

curl --location --request POST 'https://openair-dev.airtrfx.com
/airfare-sputnik-service/v3/ua/fares/route-search' \
--header 'Content-Type: application/json' \
--header 'EM-API-KEY: pleaseReplaceMe' \
--header 'Cookie:
__cfduid=d39384a48087214704e4205a68a96e2e91611772821' \
--data-raw '{
  "journeyType": "ROUND_TRIP",
  "currencies" : ["USD"],
  "outputFormat": {
    "mixedResults": false,
    "currencySettings": [
      {
        "currencyCode": "USD",
        "price": {
          "decimalSeparator": ".",
          "thousandSeparator": ",",
          "decimalPlaces": 0
        }
      }
    ]
  },
  "datePattern": "MM/dd/yy",
  "languageCode": "en"
},
"travelClasses" : ["ECONOMY"],
"routesLimit": 11,
"faresPerRoute": 1,
"dataExpirationWindow": "2d",
"departure": {
  "start": "2021-02-14",
  "end": "2022-02-24"
},

```

```
"return": {
  "start": "2021-02-15",
  "end": "2022-02-24"
},
"fareSorting": [
  {
    "priceSpecification.usdTotalPrice": "ASC"
  }
],
"outputFields": [
  "passengerDetails.count",
  "datacoreId",
  "origin.city.name",
  "origin.city.image",
  "destination.city.name",
  "destination.city.image",
  "origin.country.name",
  "destination.country.name",
  "origin.city.image",
  "destination.city.image",
  "origin.country.image",
  "destination.country.image",
  "airline.iataCode"
],
"selections": [
  {
    "routes": [
      "MIAJFK",
      "MIALAX",
      "MIABOS",
      "MIAORD"
    ],
    "faresPerRoute": 2,
    "routesLimit": 4,
    "faresLimit": 8,
    "id": "1"
  },
  {
    "routes": [
      "LAXORD",
      "LAXJFK",
      "LAXMIA",
      "LAXBOS"
    ],
    "faresPerRoute": 1,
    "routesLimit": 4,
    "faresLimit": 4,
    "id": "2"
  }
]
```

```

        "routes": [
            "BOSORD",
            "BOSMIA",
            "BOSJFK"
        ],
        "faresPerRoute": 3,
        "routesLimit": 4,
        "faresLimit": 12,
        "id": "3"
    }
]
}'

```

▼ CM - Order in 1 list

```

curl --location --request POST 'https://openair-dev.airtrfx.com
/airfare-sputnik-service/v3/cm/fares/route-search' \
--header 'Content-Type: application/json' \
--header 'EM-API-KEY: pleaseReplaceMe' \
--header 'Cookie:
__cfduid=d39384a48087214704e4205a68a96e2e91611772821' \
--data-raw '{
    "journeyType": "ROUND_TRIP",
    "currencies": [
        "USD"
    ],
    "outputFormat": {
        "mixedResults": true,
        "currencySettings": [
            {
                "currencyCode": "USD",
                "price": {
                    "decimalSeparator": ".",
                    "thousandSeparator": ",",
                    "decimalPlaces": 0
                }
            }
        ]
    },
    "datePattern": "MM/dd/yy",
    "languageCode": "en"
},
"travelClasses": [
    "ECONOMY"
],
"routesLimit": 11,
"faresPerRoute": 1,
"dataExpirationWindow": "2d",
"departure": {
    "start": "2021-02-14",

```

```
        "end": "2022-02-24"
    },
    "return": {
        "start": "2021-02-15",
        "end": "2022-02-24"
    },
    "fareSorting": [
        {
            "priceSpecification.usdTotalPrice": "ASC"
        }
    ],
    "outputFields": [
        "passengerDetails.count",
        "datacoreId",
        "origin.city.name",
        "origin.city.image",
        "destination.city.name",
        "destination.city.image",
        "origin.country.name",
        "destination.country.name",
        "origin.city.image",
        "destination.city.image",
        "origin.country.image",
        "destination.country.image",
        "airline.iataCode"
    ],
    "selections": [
        {
            "routes": [
                "MIAPTY",
                "MIADAV"
            ],
            "faresPerRoute": 1,
            "routesLimit": 2,
            "faresLimit": 8,
            "id": "1"
        },
        {
            "routes": [
                "PTYMIA"
            ],
            "faresPerRoute": 5,
            "routesLimit": 1,
            "faresLimit": 2,
            "id": "2"
        }
    ]
}'
```

## Histogram Distribution

**i** Returns daily, monthly, or yearly prices for a specific Tenant based on a **specific route** and number of filtering criteria. Please see [Sputnik Translator V3](#) for all Translator endpoints.

Please see [EM Platform: Basic Postman Setup](#) for the basic EM Platform Collection.

### Example Sample Request:

```
{
  "autoSettings": {
    "language": "en",
    "market": "us"
  },
  "origin": "DFW",
  "destination": "LAS",
  "journeyType": "ROUND_TRIP",
  "departureDaysInterval": {
    "start": 1,
    "end": 30
  },
  "returnDaysInterval": {
    "start": 2,
    "end": 60
  },
  "travelClasses": [
    "ECONOMY"
  ],
  "hierarchyRequired": true,
  "faresLimit": 1,
  "interval": "1d"
}
```

### Histogram Distribution Endpoint Request Parameters

Please see: [Sputnik: Global Filters \(all endpoints\)](#), for all filtering options. The `autoSettings` should always be sent to avoid business logic being requested at the front-end level.

Key	Value Type	Description	Example + Default
origin	String <b>Required</b>	Airport / Bus Station Code, <i>example "MIA"</i> . Filters based on the Origin Codes.	<pre>{   "origin" :   "MIA " }</pre> <i>No default</i>
destination	String <b>Required</b>	Airport / Bus Station Code, <i>example "MIA"</i> . Filters based on the Destintation Codes.	

			<pre>{   "destination"   : "ORD" }</pre>
			<i>No default</i>
interval	String <i>Optional</i>	<p>Allows to define the histogram aggregation as daily, monthly, or yearly.</p> <p>Accepted values:</p> <ul style="list-style-type: none"> <li>• 1d</li> <li>• 1m</li> <li>• 1y</li> </ul>	<pre>{   "interval" :   "1d" }</pre>
			<i>Example provided is the default sent by the system.</i>
faresLimit	Integer <i>Optional</i>	<p>Determines the total number of fare documents returned per interval selected.</p> <p><i>The total number of fares returned by the endpoint will depend on the interval and date range that have been selected.</i></p>	<pre>{   "faresLimit" :   1 }</pre>
			<i>Example provided is the default sent by the system.</i>
"hierarchyRequired": true	Boolean <i>Optional</i>	<p>Outputs the Location information in the requested language within a location object that includes an origin Iata Code and destination Iata Code object named by its value, e.g. MIA.</p>	<pre>{   "hierarchyRequired" : true }</pre>
			<i>Example provided is the default sent by the system.</i>
histogramType	Enumeration <ul style="list-style-type: none"> <li>• OUTBOUND</li> <li>• INBOUND</li> </ul> <i>Optional</i>	<p>Allows to determine if the iteration over dates is based on the Outbound or Inbound dates. The histogram will iterate based on a price per day on Outbound or Inbound dates. Ideally, when using INBOUND, the departure date would be limited to a range of one day, so as to provide information on round-trip fares based on a specific departure.</p>	<pre>{   "histogramType" :   "OUTBOUND" }</pre>
			<i>Example provided is the default sent by the system.</i>
priceBuckets	Object <i>Optional</i>	<p>Can be used to request two separate features in the response: Price Buckets; and Price Stats. Please see below for more info, and also <a href="#">How does the Price Bucket work in histogram-distribution?</a>.</p> <ul style="list-style-type: none"> <li>• <b>Example (default when departureRange not sent):</b></li> </ul>	<pre>{   "priceBuckets" :   {</pre>



		<pre> "priceBuckets": {   "active": true,  "priceStats": true,  "departureRange" : {   "start": 0,   "end": 365 } } </pre>	<pre> "priceStats": false,   "active": false,   "field": "TOTAL_PRICE",  "departureRange": {   "start": 0,   "end": 365 } } </pre>
<pre>priceBuckets .priceStats</pre>	<p><b>Boolean</b></p>	<p>When the priceStats is marked as true, the response will include a responsePriceStats object that provides the Price Stats based on the results of the Histogram Distribution response (<i>fares returned from the request</i>). The priceStats should be included when requesting price stats and buckets.</p>	<pre> {   "priceBuckets": {  "priceStats": false } } </pre> <p><i>Example provided is the default sent by the system.</i></p> <p> <a href="#">responsePriceStats Object Response Example</a> </p> <pre> {"responsePriceStats": {   "mixedCurrencies": false,   "min": {     "price": 878.17,     "formattedPrice": "879USD",     "shortenedFormattedPrice": "879USD"   },   "avg": {     "price": 977.42,     "formattedPrice": "978USD",     "shortenedFormattedPrice": "978USD"   },   "max": {     "price": 1076.67,     "formattedPrice": "1,077USD", </pre>

			<pre>"shortenedFormattedPrice": "1.1 KUSD"  }  }  }</pre>
priceBuckets         .         departureRange	<ul style="list-style-type: none"> <li>• <b>Object</b> with:             <ul style="list-style-type: none"> <li>• start integer</li> <li>• end integer</li> </ul> </li> </ul>	<p>The <code>departureRange</code> is an interval setting for the start and end that calculates the actual date dynamically based on the day the request is sent in. The setting can be added to the <code>priceBuckets</code> to determine the time period from which the data buckets will be calculated. The start must be at least 0, and the end cannot exceed 500.</p>	<pre>"priceBuckets": {   "active": true,  "priceStats": true,  "departureRange": {   "start": 0,   "end": 365 } }</pre> <p><i>Default when priceBuckets.active.true</i></p>
priceBuckets         .active	Boolean	<p>The <code>priceBuckets</code> can be used to return all fare information categorized into 3 buckets to essentially note if the price is low, medium, or high for the route specified and over the next year, or whatever configuration is used in <code>priceStats.departureRange</code>.</p>	<pre>{   "priceBuckets": {   "active": true,   "field": "TOTAL_PRICE" } }</pre> <p><i>Default for active is false, and field does not need to be included unless the default should be overwritten to USD_TOTAL_PRICE.</i></p> <p>▼ <b>buckets Object Response Example</b></p> <pre>"buckets": { "max": { "priceRange": { "minimum": 1787.6666, "formattedMinimum": "1,788USD", "shortenedFormattedMinimum": "1.8 KUSD", "maximum": 2267.0, "formattedMaximum": "2,267USD", "shortenedFormattedMaximum": "2.3 KUSD" }, "lowestFare": null, "formattedLowestFare": null, "shortenedFormattedLowestFare": null, "totalResults": 0 }, "min": { "priceRange": { "minimum": 826.0, "formattedMinimum": "826USD", "shortenedFormattedMinimum":</pre>
priceBuckets         .field	Enumeration <ul style="list-style-type: none"> <li>• TOTAL_PRICE (default)</li> <li>• USD_TOTAL_PRICE</li> </ul>	<p>When requested, the response will include a <code>buckets</code> object with the Price Range for each bucket, the total number of results in the response for each bucket, and the best available fare in each bucket that is in the response. All prices include a formatted version. Additionally, the <code>priceSpecification</code> field for each one of the fares in the response is tagged with a <code>bucket</code> field that specifies if it is in the min, avg, or max bucket.</p> <p>In order to request bucket information for the prices returned, the <code>active</code> field must be set to <code>true</code>. Additionally, the <code>field</code> enum can be used to rank the fares based on the Total Price (<code>TOTAL_PRICE</code>) or the USD normalized total price (<code>USD_TOTAL_PRICE</code>), which can make sense for multi-currency site editions. The <code>field</code> attribute is not required and will be defaulted to <code>TOTAL_PRICE</code>.</p> <p>To generate the buckets, both the <code>/price-stats</code> and <code>/histogram</code> endpoints are requested in parallel. The exact same request is used, except that the <code>departureDateInterval</code> is set to a range of 0 to 365 days for the <code>/price-stats</code>, or whatever configuration is used in <code>priceStats.departureRange</code>. To learn more about how the buckets are calculated, please see: <a href="#">How does the Price Bucket work in histogram-distribution?</a></p>	

```
"826USD", "maximum": 1306.3334,
"formattedMaximum": "1,307USD",
"shortenedFormattedMaximum": "1.3
KUSD" }, "lowestFare": 877.97,
"formattedLowestFare": "878USD",
"shortenedFormattedTLowestFare":
"878USD", "totalResults": 2 },
"avg": { "priceRange": {
"minimum": 1307.3334,
"formattedMinimum": "1,308USD",
"shortenedFormattedMinimum": "1.3
KUSD", "maximum": 1786.6666,
"formattedMaximum": "1,787USD",
"shortenedFormattedMaximum": "1.8
KUSD" }, "lowestFare": null,
"formattedLowestFare": null,
"shortenedFormattedTLowestFare":
null, "totalResults": 0 } }
```

priceSpecification Example with "bucket"

```
"priceSpecification": {
"totalPrice": 1230.97,
"usdTotalPrice": 1230.97,
"currencyCode": "USD",
"formattedTotalPrice": "1,231USD",
"shortenedFormattedTotalPrice":
"1.2KUSD", "bucket": "min" },
```

## Trending Destinations

**i** Returns flight fares for a specific Tenant based on a number of filtering criteria and organizes the **response based on the Destination**. The request allows to specify the number of Destinations that can be returned, and how many fares per destination. Please see [Sputnik Translator V3](#) for all Translator endpoints.

Please see [EM Platform: Basic Postman Setup](#) for the basic EM Platform Collection.

## Example Sample Request:

```
{
  "autoSettings": {
    "language": "en",
    "market": "us"
  },
  "origins": [],
  "journeyType": "ROUND_TRIP",
  "travelClasses": [
    "ECONOMY"
  ],
  "routesLimit": 30,
  "faresPerRoute": 10,
  "destinationsLimit": 10,
  "departure": {
    "start": "2022-02-01",
    "end": "2022-10-01"
  }
}
```

Please see: [Sputnik: Global Filters \(all endpoints\)](#), for all filtering options. The `autoSettings` should always be sent to avoid business logic being requested at the front-end level.

<p><code>routesLimit</code></p>	<p>Integer <i>Optional</i></p>	<p>Provides the number of routes the service should return.</p> <ul style="list-style-type: none"> <li>• Maximum is 100</li> </ul>	<pre>{   "routesLimit" : 1 }</pre> <p><i>Example provided is the default sent by the system.</i></p>
<p><code>faresPerRoute</code></p>	<p>Integer <i>Optional</i></p>	<p>Provides the number of fares per route the service should return.</p> <ul style="list-style-type: none"> <li>• Maximum is 100</li> </ul>	<pre>{   "faresPerRoute"   : 1 }</pre> <p><i>Example provided is the default sent by the system.</i></p>
<p><code>destinationsLimit</code></p>	<p>Integer <i>Optional</i></p>	<p>Provides the number of total destinations that should be returned. Fares are grouped within each destinations</p> <ul style="list-style-type: none"> <li>• Maximum is 100</li> </ul>	<pre>{   "destinationsLimit"   : 1 }</pre> <p><i>Example provided is the default sent by the system.</i></p>
<p><code>faresLimit</code></p>	<p>Integer <i>Optional</i></p>	<p>Creates an alternated list of the top results based on the total number of fares determined in this field. If more are specified in the other fields, <code>faresLimit</code> will limit the number. If less are listed, than only those will be returned.</p> <p><i>Please note that the Sputnik V2 Grouped Routes output does not alternate the list of results.</i></p> <ul style="list-style-type: none"> <li>• Maximum is 100</li> </ul>	<pre>{   "faresLimit" : 1 }</pre> <p><i>Example provided is the default sent by the system.</i></p>