

FlightLookup



FlightLookup Timetable Lookup API

TimeTable Web Services OTAx Reference

Powered by FlightLookup
Version: 3.3
Updated: January 29, 2021



Contents

Overview.....	3
Method: TimeTable	4
Parameter Values	5
Result Options.....	6
Traffic Restriction Code filtering	6
Parameter Value Options Tables	7
Sort Order Options Table.....	7
Compression Options Table.....	8
Connection Options Table	9
Time Options.....	9
Interline Options Table	10
Output	11
OTA_AirDetailsRS	11
FlightDetails	11
FlightLegDetails	11
Understanding Seven Day Lookup Results	12
Understanding Error Responses.	12
Error Response	13
XML Output Document Definition	13
Structure of Response	13
Example of a successful Route lookup	14
Method: FLSUUID.....	17
Example of a successful FLUUID lookup	18
What are Codeshares?	20
IATA Codes for General Traffic Restriction.....	21
IATA Regional Codes	22
IATA Service Type Codes	23

Overview

Description

This document contains FlightLookup Web Services descriptions for the following services.

Web Services

Service	Description
FlightLookup Web Services	FlightLookup Web Services, powered by FlightLookup Routing Engine API provides a powerful information source for developing sophisticated applications requiring flight related schedule data. Whether you are a software company developing products for sale or an in-house developer, the FlightLookup Web Services APIs are unmatched in terms of breadth and richness of information. Our development tools and documentation will get you up to speed quickly enabling short development cycles.

Notes

The OTA XML format we use is based on this XSD file:

http://www.opentravel.org/2011B/OTA_AirDetailsRS.xsd

The output, by default, includes fields which start with the letters “FLS” that are not part of the OTA specification. If full OTA compliance is required, the “device” file should be “ota-xml-compliant”, and lookups should always be individual lookups and never batch lookups.

This document covers single requests to the FlightServer.

If you require the ability to perform Batch Lookups review the BatchLookup Webservice documentation.

Method: TimeTable

Description

This method returns timetable data from a specified airport to a specified airport on a specified date.

TimeTable Method

- **Required:**
 - From - 3 letter IATA code for the departure airport
 - To - 3 letter IATA code for the destination airport
 - Date - Departure date (YYYYMMDD)
- **Optional:** (a fill list is in the table below)
 - Airline – letter IATA airline code. Restricts results to routes flown by a certain carrier.
 - Connections - See below.
 - Count (Default=20) - Number of routes to return.
 - Interline (Default="Y") or "N"
 - SortOrder (Default =0) - Default sort order. See below.

Input

Single Day Lookup Format

```
/v1/xml/TimeTable/From/To/Date/?<param>=<value>&<param>=<value>
```

7 Day Lookup Format

```
/v1/xml/TimeTable/From/To/Date/?7Day=Y&<param>=<value>
```

For a display of which days of the week the flights operate on, add the 7Day parameter. The first day of the week is Monday. Check the description of the output variable FLSFlightDays, in the section "Understanding Seven Day Lookup Results", for a full explanation. "1.3.567", for example, means "Monday, Wednesday, Friday, Saturday, and Sunday".

Example input

```
http://example.com/v1/xml/TimeTable/JFK/LON/20140413/?7Day=Y&Sort=Departure&Airline=BA&Connection=NONSTOP&Count=20&Language=en&TRC=N
```

In this example, we look up flights from New York (JFK) to London (LON) on April 13, 2013, over a 7-day period, sorted by departure time, on British Airways (BA). The flights are direct

(NONSTOP) flights; we only show up to 20 flights. The output language is English and we disable traffic restriction code filters (TRC=N).

Parameter Values

(Optional Parameters are shaded in grey; Default values are listed)

Argument Name	Valid Values	Examples
From	IATA 3 letter Airport Code	BOS
To	IATA 3 letter Airport Code	LAX
Date*	Date in YYYY/MM/DD format 2017/04/13	2017-04-13
7Day	Setting to "Y" results in 7 days of results.	Y
Connection	Displays non-stops, directs and logical single and multiple connections up to the maximum number of results requested - See table for option descriptions. AUTO/NONSTOP/DIRECT/1STOP/MORE - See table (default AUTO)	AUTO
Compression	See table (default ALL)	ALL
Sort	Departure/Arrival/Duration/Flights See table for more options	1
Time	See table (default ANY)	ANY
Language	ISO Language Code (see table below for available Languages) (default en)	en
Interline	Y/N/A/I - See table (default N)	N
Results	1-500 (default 100)	100
Count	Number of items to return (default set equal to results)	100
Index	Index of the first result Item to be used with Count	1
Airline	Restrict by specific airline 2 or 3 letter airline IATA code	AA
FlightNumber	Restrict by specific flight number must be used with Airline	100
TRC	Y/N Traffic Restriction Code filters to include all restricted flights that may have extra conditions for purchase or booking. See Notes. (default is Y)	Y
CodeShare	Y/N Display CodeShare (Marketing) flights. See Notes (default is Y)	Y
MaxConnect	Default is 390 minutes, max value is 1200 minutes	390

Result Options

Numeric value from 1 to 500 with a default of 100 which controls the maximum number of flights returned from the query. FlightLookup advises 40 for internet timetables and 10 for mobile devices.

Paging Results

Consider using Count and Index parameters to page through results if necessary.

Traffic Restriction Code filtering

Traffic restriction codes are applied where appropriate connections are not built involving restricted flights. Fully describing Traffic Restriction Codes is beyond the scope of this document, but the most common issue that may utilize the overriding of Traffic Restrictions is:

Display all flights. Use this to show both Operating and Marketing Flights in an Origination/Destination market or airport. Sometime, an airplane may fly into an airport, but is either a Codeshare (Marketing Flight) or the airplane might not pickup or drop off passengers due to legal restrictions imposed by Governmental regulations, airline policy or inter airline policy. When this happens, a flight may not appear in the results; for them to appear this override is required.

In this example. we want all the flights into Hamburg to display.

The issue is that: UA flights do not appear. The cause: The Operating Carrier is Lufthansa and United is the Marketing Carrier. The United flights have a Traffic Restriction Code not to display the flight unless it is connecting with another United flight. With TRCChecking=Y the flights will not display, but with TRCChecking=N the flights will appear.

A list of TRC "Traffic Restriction Codes" is included, along with descriptions, at the end of the document to help better understand these restrictions.

An explanation of how to disable individual TRC's is beyond the scope of this document.

Parameter Value Options Tables

Sort Order Options Table

Sort order options may be specified as either numeric or string values. When this is used, the secondary Sort key is Departure Time. For example, if sorting by number of flights (5), flights with the same number of connections that depart early in the day will appear before flights that depart later on.

Value	Description
0 or Arrival	Sorts by Arrival Time
1 or Departure	Sorts by Departure Time
2 or Duration	Sorts by Duration
3	Flight Type (Nonstop/Direct/Connection)
4	Sorts by VIA city
5 or Flights	Sorts by connections, Note: direct flights get sorted in the list in a special way as they are specifically selected routes published by the carriers, and not selected through the flight schedule routing and connection rule process.
6	Prefer Operating Flights (non codeshare flights) (Operating flights will show up first base on the number of Results requested). Note: Codeshare flights are also referred to as Marketing Flights.

Compression Options Table

Value	Description
OPTIMAL	<p>Collapses flights based on 3 rules:</p> <p>Collapses multiples flights into one. This option is most visible on a 7-day lookup. Some airlines report flights on each day of the week in the SSIM so you will get flights operating on 1,2,3,4,5,6,7 all on separate reporting lines – as long as the flight arrives and leaves at the same time on all days we will collapse them into one line with a Daily operating schedule rather than showing seven separate lines.</p> <p>Remove redundancies. This looks at flights arriving or departing that are close to each other – for example a flight leaving 20 minutes or arriving within 20 minutes – and displays the most efficient flight.</p> <p>Funnel. This looks at all the legs in an itinerary and selects the most efficient based on the arrival time.</p>
MOST	<p>Collapses multiple flights into one. This option is most visible on a 7-day lookup. Some airlines report flights on each day of the week in the SSIM so you will get flights operating on 1,2,3,4,5,6,7 all on separate reporting lines – as long as the flight arrives and leaves at the same time on all days we will collapse them into one line with a Daily operating schedule rather than seven separate lines.</p>
ALL	No compression.

Connection Options Table

Value	Description
AUTO	Displays non-stops, directs and logical single and multiple connections
NONSTOP	Displays non-stops flights only
DIRECT	Displays non-stop and direct flights only (A Direct flight maintains the same flight number throughout the itinerary)
1STOP	Displays non-stops, directs, and flights with one via connection point
MORE	Displays non-stops, directs and flights with up to three via connection points

Time Options

Value	Description
ANY	All time periods. Note selection of flights based on number of results will be displayed. If flights are missing increase the number of Results.
AM	Preference given to AM flights, but more may appear based on number of results. Primary preference is the number of Results, if there are 5 AM flights and Results=20 then 20 flights will appear if there are 20 flights based on query.
PM	Preference given to PM flights, but more may appear based on number of results. Primary preference is the number of Results, if there are 5 PM flights and Results=20 then 20 flights will appear if there are 20 flights based on query.
NIGHT	Preference given to night flights from 6 PM to 6 AM. More may appear depending on the number of results

Interline Options Table

Value	Description
N	No interline
Y	Interlining is turned on. Codeshare variations will be dropped if it looks as though the non-codeshare flight will also show up. This option could occasionally fail to display a connection, but only in very rare situations caused by subtle Traffic Restriction Code rules.
A	Considers all variations of every flight segment when building connections. This is so that no valid connections get missed, no matter how complicated the Traffic Restriction Codes rules are.
I	This allows interlining for international connections only; domestic connections must be online.

Output

XML output is returned in a three level hierarchy: OTA_AirDetailsRS :: FlightDetails :: FlightLegDetails

Hierarchy	Fields
OTA_AirDetailsRS	TransactionIdentifier, FLSDevice, FLSResponseFields (FLSOriginCode, FLSOriginName, FLSDestinationCode, FLSDestinationName, FLSStartDate, FLSendDate)
FlightDetails	TotalFlightTime, TotalMiles, TotalTripTime, FLSDepartureDateTime, FLSDepartureTimeOffset, FLSDepartureCode, FLSDepartureName, FLSArrivalDateTime, FLSArrivalTimeOffset, FLSArrivalCode, FLSArrivalName, FLSFlightType, FLSFlightLegs, FLSFlightDays, FLSDayIndicator, FLSUUID
FlightLegDetails	DepartureDateTime, ArrivalDateTime, FLSArrivalTimeOffset, FlightNumber, JourneyDuration, SequenceNumber, FLSMeals, FLSInflightServices, FLSUUID, FLSUUIDActualFlight, DepartureAirport (LocationCode, FLSLocationName, Terminal, FLSDayIndicator, FLSDOTDisclosure), ArrivalAirport (LocationCode, FLSLocationName, Terminal), MarketingAirline (Code, CompanyShortName), OperatingAirline (Code, CodeContext, CompanyShortName, FlightNumber)

All fields which start with the letters “FLS” are not part of the OTA specification. If full OTA compliance is required, the “device” file should be “ota-xml-compliant”, and lookups should always be individual lookups and never batch lookups.

Understanding Seven Day Lookup Results

Seven day searches allow calculating flight results for a range of seven days from the start date.

The XML tag FlightDetails has an attribute called FLSFlightDays. FLSFlightDays is a string of 7 characters which represent the flight frequency mask. Each position in the string will contain either a number or a period. Numbers are the day of operation flags, and the periods represent days that the flights do not operate on. The numbers range from 1 to 7 where 1= Monday; 2= Tuesday; 3=Wednesday; 4=Thursday; 5=Friday; 6=Saturday; 7=Sunday.

A flight that operates every day will be displayed this way

```
FLSFlightDays="1234567"
```

A flight that only operates on Friday is displayed this way

```
FLSFlightDays="....5.."
```

A flight that operates every day except Tuesday and Thursday is displayed this way

```
FLSFlightDays="1.3.567"
```

These number are for the seven days following the search. So if the search date falls on a Friday, day 1 in the array for Monday represents the following Monday, not the Monday of the week which includes the search date.

If today is Friday 2014-05-30 and FLSFlightDays="1.3.567" then the flight operates today (Friday), tomorrow (Saturday), Sunday, Monday, Wednesday. To get more information beyond 7 days requires another look up changing the date by 7 days.

Understanding Error Responses.

Each item allows for the potential of containing bad input, or an invalid lookup request. These will result in an error message. This response assumes that the request failed to find a valid IATA city ID called ZZZ.

Note that there are multiple indicators here to assist you in understanding these error messages, and determining what could possibly have gone wrong.

The most notable is the content of the <Errors> block. Within the <Errors> block will be a detailed <Error> message which will contain an error type, an error code, and a text description of the error message. A comment will also follow the message with a longer version of the error message.

The formatting of the actual request processed by the FlightServer is contained in the parent block <Lookup> block in the Request attribute. Within that attribute is both the BatchID causing the error and the request as submitted to the flight server. In the event that you believe that an error code should not have been produced, it is the Request Attribute that you should share with the support department when communicating about this any questions.

Error Response

```
<OTA_AirDetailsRS PrimaryLangID="eng" Version="1.0" TransactionIdentifier=""
xmlns="http://www.opentravel.org/OTA/2003/05">
  <Errors>
    <Error Type="ERR" FLSErrorCode=-2 FLSErrorName="Invalid from" />
  </Errors> <!-- Reason for error: invalid from -->
</OTA_AirDetailsRS>
```

XML Output Document Definition

Flight schedule results adhere to the following general structure:

OTA_AirDetailsRS :: FlightDetails :: FlightLegDetails

Structure of Response

```
<OTA_AirDetailsRS [...]>
  <FlightDetails>
    <FlightLegDetails>
      [...]
    </FlightLegDetails>
  </FlightDetails>
</OTA_AirDetailsRS>
```

```
        [...]
      </FlightDetails>
    [...]
  </OTA_AirDetailsRS>
```

Example of a successful Route lookup

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirDetailsRS
  PrimaryLangID="eng"
  Version="1.0"
  TransactionIdentifier="03a1b41a050cf9139150901f08050979609062e30609377a"
  FLSNote="This XML adds attributes not in the OTA XML spec. All such attributes start with FLS"
  FLSDevice = "ota-xml-expanded"
  xmlns="http://www.opentravel.org/OTA/2003/05"
>
  <Success>
  </Success>

  <FLSResponseFields
    FLSOriginCode="ORD"
    FLSOriginName="Chicago O'Hare"
    FLSDestinationCode="AVV"
    FLSDestinationName="Melbourne Avalon"
    FLSStartDate="2014-03-25"
    FLSEndDate="2014-03-25"
  />

  <FlightDetails
    TotalFlightTime="PT21H05M"
    TotalMiles="9715"
    TotalTripTime="PT29H00M"
    FLSDepartureDateTime="2014-03-25T17:40:00"
    FLSDepartureTimeOffset="-0500"
    FLSDepartureCode="ORD"
    FLSDepartureName="Chicago O'Hare"
    FLSArrivalDateTime="2014-03-27T14:40:00"
    FLSArrivalTimeOffset="+1100"
    FLSArrivalCode="AVV"
```

FLSArrivalName="Melbourne Avalon"
FLSFlightType="Connect"
FLSFlightLegs="3"
FLSFlightDays=".2....."
FLSDayIndicator="+2"
>

<FlightLegDetails
DepartureDateTime="2014-03-25T17:40:00"
FLSDepartureTimeOffset="-0500"
ArrivalDateTime="2014-03-25T20:10:00"
FLSArrivalTimeOffset="-0700"
FlightNumber="241"
JourneyDuration="PT4H30M"
SequenceNumber="1"
LegDistance="1743"
FLSMeals=""
FLSInflightServices=" 9"
FLSUUID="ORDLAX20140325VX241"
>

<DepartureAirport
CodeContext="IATA"
LocationCode="ORD"
FLSLocationName="Chicago O'Hare"
Terminal="3"
FLSDayIndicator=""
FLSDOTDisclosure=""
>

<ArrivalAirport
CodeContext="IATA"
LocationCode="LAX"
FLSLocationName="Los Angeles"
Terminal="3"
FLSDayIndicator=""
>

<MarketingAirline
Code="VX"
CodeContext="IATA"

```
CompanyShortName="Virgin America"  
>
```

```
<OperatingAirline  
Code="VX"  
CodeContext="IATA"  
CompanyShortName="Virgin America"  
FlightNumber="241"  
>
```

```
<Equipment  
AirEquipType="320"  
>
```

```
</FlightLegDetails>
```

```
[. . . .]
```

```
</FlightDetails>  
</OTA_AirDetailsRS>
```

Method: FLSUUID

Description

This method returns one individual flight record based on a FLSUUID

FLSUUID="LAXATL20120812DL2354"

This method is a convenient method for retrieving detailed records based on the FLSUUID as returned in a list of flights. It complements our BatchLookup Webservice, which is more fully described in a separate document; Batch Processing of FLSUUID Lookups is done with our separate BatchLookup Service.

FLSUUID Method

- **Required:**
 - FLSUUID String

Input

FLSUUID Lookup Example

/v1/xml/FLSUUID/BOSCDG20120904AA146

This example shows a flight from Boston (BOS) to the Charles de Gaulle (CDG) airport in France, on September 4 2012, using American Airlines (AA) flight 146.

FLSUUID String Format

+++---YYYYMMDDCCFFFF

+++ DEP

--- ARR

[~] **Optional** Arrival Date Indicator

YYYY Year (4 digit)

MM Month Number (zero padded)

DD Day Number (zero padded)

CC Airline IATA CODE

FFFF FlightNumber

+++---MMDDYYYYCCFFFF

BOSCDG20120904DL8399

BOSCDG20120904AF337

BOSCDG20120904AA146

Example input

<http://example.com/v1/xml/FLSUUID/BOSCDG20120904DL8399>

This example shows a flight from Boston (BOS) to the Charles de Gaulle (CDG) airport in France, on September 4 2012 (20120904), using Delta (DL) flight 8399.

Output

XML output is returned in a three level hierarchy: OTA_AirDetailsRS :: FlightDetails :: FlightLegDetails as described in the OUTPUT section of this Documentation

SEE ALSO - BATCHLOOKUP SERVICE DOCUMENTATION for Bulk Requests and additional options.

Example of a successful FLUID lookup

```
curl -X GET "http://example.com/v1/xml/FLSUUID/BOSLAX20140515"
```

```
<FLSBatch>
<FLSLookup FLSUUID="BOSLAX20140515" From="BOS" To="LAX" Date="2014-05-15">
<OTA_AirDetailsRS
  PrimaryLangID="eng"
  Version="1.0"
  TransactionIdentifier="03a2983d050ea0fd9150901f08050979609021ca06097453"
  FLSNote="This XML adds attributes not in the OTA XML spec. All such attributes start with FLS"
  FLSDevice = "ota-xml-expanded"
  xmlns="http://www.opentravel.org/OTA/2003/05">
<Success></Success>
<FLSResponseFields
  FLSOriginCode="BOS"
  FLSOriginName="Boston"
  FLSDestinationCode="LAX"
  FLSDestinationName="Los Angeles"
  FLSStartDate="2014-05-15"
  FLSEndDate="2014-05-15"
/>
<FlightDetails TotalFlightTime="PT6H20M"
  TotalMiles="2608"
  TotalTripTime="PT6H20M"
  FLSDepartureDateTime="2014-05-15T06:00:00"
  FLSDepartureTimeOffset="-0400"
```

```
FLSDepartureCode="BOS"  
FLSDepartureName="Boston"  
FLSArrivalDateTime="2014-05-15T09:20:00"  
FLSArrivalTimeOffset="-0700"  
FLSArrivalCode="LAX"  
<!----- snip ----->  
  <MarketingAirline Code="DL"  
    CodeContext="IATA"  
    CompanyShortName="Delta Air Lines" />  
  <OperatingAirline Code="AF" CodeContext="IATA" CompanyShortName="Air France"  
FlightNumber="321" />  
  <Equipment AirEquipType="772" />  
</FlightLegDetails></FlightDetails>  
</OTA_AirDetailsRS>  
</FLSLookup>  
</FLSBatch>
```

What are Codeshares?

In the following example, both Marketing and Owned By conditions can be explained.

LH1343	Frankfurt	06:15	2A
AC9023	Frankfurt	06:15	2A
BD3659	Frankfurt	06:15	2A
SN7148	Frankfurt	06:15	2A
TP7967	Frankfurt	06:15	2A
UA9023	Frankfurt	06:15	2A
US5679	Frankfurt	06:15	2A

The Operating flight is LH1343. Note this flight is actually owned by 2A (Detsche Bahn)

Flights AC9023, BD3659, SN7148, TP7967, UA9023, and US5679 are marketing flights, referred to as codeshares for the LH1343 flight.

These are based on *Marketing Agreements* between LH and these other airlines, allowing the extension of their network with their own flight numbers on a given flight, such as a flight operated by LH as LH1343. While each one has a separate “Marketing Flight Number” there is only one physical aircraft with the primary carrier of LH, also known as the Operating Carrier. In the case above, the flight is owned by 2A (Deutsche Bahn) with Lufthansa having the primary relationship, other airlines are Marketing a Lufthansa Flight (in this case a Train!). Note: this type of relationship is quite common; the US Department of Transport requires this information as part of full disclosure when displaying flight schedule information.

IATA Codes for General Traffic Restriction

Traffic restriction codes are applied and where appropriate connections are not built involving restricted flights. Traffic restrictions are for a segment (one or more contiguous legs of a flight) and are applied as follows:

Traffic Restriction Code	Code Description
A	No boarding this city (flight item is not displayed and cannot be sold)
B	Not valid for connections
C	Not valid for international connections
D	Valid for international online connections only, except if D, E, or G restrictions apply into and out of all online connect points
E	Valid for online connections only, except if D, E, or G restrictions apply into and out of all online connect points
F	Not valid for interline connections
G	Valid for Online Connections Only; except if D, E, or G restrictions apply into and out of all online connect points (Note: Proper application of this rule is complicated).
H	Not valid for connections and the direct service is not displayed (flight can be sold)
I	No boarding/deplaning this city (flight item is not displayed and cannot be sold)
K	Valid for connections only
M	Valid for international online stopovers only
N	Valid for international connections only
O	Valid for international online connections only
Q	Valid for international online stopover and connections only
T	Valid for online stopovers only
V	Valid for stopover and connections only
W	Valid for international stopover and connections only
X	Valid for online stopover and connections only
Y	Valid for online connections only
Z	Multiple traffic restrictions apply

IATA Regional Codes

Code	Code Description
AFR	Africa (excluding Egypt)
CAR	Caribbean sea countries
CEM	Central America
EUR	Europe
JAK	Japan and Korea
MDE	Middle East (includes Egypt)
NOA	North America
SAS	South Asia
SCH	Schengen agreement countries
SEA	South-Eastern Asia
SOA	South America
SWP	South-West Pacific

IATA Service Type Codes

Service Type	Description
J	Scheduled: Passenger Normal Service
S	Scheduled: Passenger Shuttle Mode
U	Scheduled: Passenger Service operated by surface vehicle
F	Cargo/Mail: loose loaded cargo and/or preloaded devices
V	Cargo/Mail: service operated by surface vehicle
M	Cargo/Mail: Mail only
Q	Passenger/Cargo: in cabin
G	Additional Flights: Passenger Normal Service
B	Additional Flights: Passenger Shuttle Mode
A	Additional Flights: Cargo/Mail
R	Additional Flights: Passenger/Cargo in cabin
C	Charter: Passenger only
O	Charter: requiring special handling
H	Charter: Cargo and/or Mail
L	Charter: Passenger and Cargo and/or Mail
P	Other: Non-revenue (Positioning/Ferry/Delivery/Demo)
T	Other: Technical Test
K	Other: Training (School/Crew check)
D	Other: General Aviation
E	Other: Special (FAA/Government)
W	Other: Military
X	Other: Technical Stop