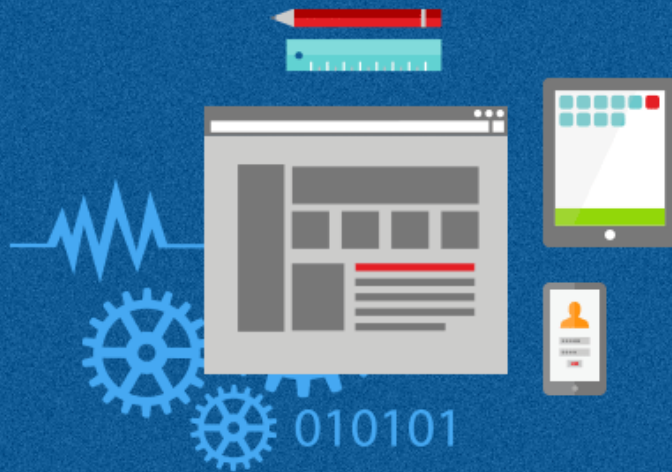


Sabre

**SABRE
APIs**



Developer Administration Guide

Bargain Finder Max API

Shop for NDC Carrier Content / Integrate NDC Offers in Shopping

Schema Version 1

Release 2019.03

(March 2018)

Schema version 1

Document Edition 1.0 (March 2019)

This documentation is the confidential and proprietary information of Sabre Inc. Any unauthorized use, reproduction, preparation of derivative works, performance, or display of this document, or software represented by this document, without the express written permission of Sabre Inc., is strictly prohibited.

Sabre and *Sabre Dev Studio* are trademarks and/or service marks of an affiliate of Sabre Corporation. All other trademarks, service marks, and trade names are the property of their respective owners.

© 2019 Sabre Inc. All rights reserved.

Table of Contents

1	API Overview	1-1
	1.1 Bargain Finder Max Introduction	1-1
	1.2 About This Guide	1-1
	1.3 Getting Started.....	1-2
	1.4 API Information	1-2
	1.5 Shop for NDC carrier content with Bargain Finder Max API	1-4
2	Quick Start Guide	2-5
	2.1 Product Usage and Examples	2-5
3	API Reference	3-13
	3.1 The beginner's guide to Air Shopping	3-13
	3.2 Schema definitions.....	3-13
	3.3 Parameters introduced to the Request schema.....	3-13
	3.4 Parameters introduced to the Response schema	3-15
4	Error Messages and Status Codes	4-1
	4.1 Introduction	4-1
	4.2 Status Codes	4-9
	Glossary	1

• • •

API Overview

1.1 Bargain Finder Max Introduction

Bargain Finder Max (BFM), our best-in-class low fare search product, is used to search for the best available priced itineraries based upon a specific date. Bargain Finder Max is an interface into the Air Shopping engine through a Universal Service Gateway (USG) and 2nd Generation Services Gateway (2SG) that allows users to use the shopping engine without a traditional Sabre terminal. When a shopper searches for low fares, the JSON (JavaScript Object Notation) within Bargain Finder Max communicates with the shopping engine and returns the appropriate response.

NDC Update

New enhancements to our shopping capabilities were made so Bargain Finder Max searches for air offers distributed through the NDC (New Distribution Capability) standard. The new capability allows a Sabre agency to shop for NDC content, Low Cost Carrier (LCC) content, and traditionally filed content (ATPCO), in one transaction. The returned content contains one integrated shopping response from all sources.

1.2 About This Guide

This guide provides information on the enhancements made to the *Bargain Finder Max API* that allows users to search for air offers distributed through the NDC (New Distribution Capability) standard. Each chapter is as follows:

- Chapter 1, “API Reference” describes the changes and new parameters added to the schemas.
- Chapter 2, “**Error! Reference source not found.**” describes the details of the product usage and examples.
- Chapter 3, “**Error! Reference source not found.**” describes the error messages for Bargain Finder Max functionality.
- Glossary, provides a glossary of terms used in this document.

1.3 Getting Started

This documentation is the confidential and proprietary product of Sabre Inc. Any unauthorized use, reproduction, or transfer of this documentation, in any medium, or incorporation of this software/documentation into any system or publication, is strictly prohibited. © 2008-2018 Sabre Inc. All rights reserved.

1.4 API Information

This service requires activation, if you are not a current BFM customer, please contact your Sabre Account Representative for assistance.

1.4.1 Target Audience

All Sabre APIs customers.

1.4.2 Authentication

[Sessionless Token](#)

1.4.3 Version information

BargainFinderMax Request/Response Version 1

1.4.4 URI Scheme

Environment	Host	BasePath	Schemes
PROD	https://api.havail.sabre.com/v1/offers/shop	/v1	HTTP
CERT	https://api-cert.cert.havail.sabre.com/v1/offers/shop	/v1	HTTP
TEST	https://api.tsts.havail.sabre.com/v1/offers/shop	/v1	HTTP

1.4.5 Methods/Endpoints

1.4.6 POST /v1/offers/shop

1.4.6.1 Description

POST: Creates offers from the NDC, ATPCO, and LCC content sources.

1.4.7 Technology / Language

REST/JSON

1.4.8 Response Type

GIR – Group Itinerary Response

1.4.9 Contact information

Email: ShoppingWebservicesDocs@sabre.com

Url: <https://developer.sabre.com>

1.5 Shop for NDC carrier content with Bargain Finder Max API

The purpose of this guide is to describe the NDC (New Distribution Capability) within the *Bargain FinderSM Max* (BFM) API.

What is it?

The NDC Standard enhances the capability of communications between airlines and travel agents.

This capability enables Sabre to transform the way air products are retailed to leisure and business travelers, by addressing the industry's current distribution limitations:

- Product differentiation and time-to-market
- Access to full and rich air content
- Transparent shopping experience

What does it do?

Bargain Finder Max supports offer content provided by NDC-compliant carriers when shopping. This solution:

- Reaches out to NDC carriers for offer details
- Interacts with the Offer Store, a database that allows storing retrieving Offers and their Offer items.
- Enhances currently existing JSON BFM schemas

How does it work?

This enhancement adds the capability to search for offers distributed through the NDC Standard in one integrated response from all sources.

The enhancement introduces additional parameters to the Bargain Finder Max API's shopping capabilities. The shopping service is available to developers via REST/JSON with the GIR response type.

Quick Start Guide

This chapter includes the following:

- Product usage and example (see “**Error! Reference source not found.**,” page 2-5).
- JSON examples (see “**Error! Reference source not found.**,” page 2-5).

2.1 Product Usage and Examples

This section includes instructions on how to enable and get shopping offers with the different content sources in the Bargain Finder Max API request and response.

2.1.1 Trying the API

Use the [API Explorer](#) on [Sabre Dev Studio](#) to explore Sabre REST APIs by making calls to the test environment. Please note that you first need to [sign in](#) before testing our APIs. You can also view our [written documentation](#).

2.1.2 Use Cases & Examples

The following use case is denoted via REST/JSON only.

USE CASE 1 (*Bargain Finder Max request and response for simple one-way itinerary to return solutions in which content source for NDC only is requested*)

Example Request:

```
{
  "OTA_AirLowFareSearchRQ": {
    "TruncateMessages": false,
    "Target": "Production",
    "Version": "0.0.1",
    "POS": {
      "Source": [
        {
          "PseudoCityCode": "XXXX",
          "RequestorID": {
            "Type": "0.AAA.X",
            "ID": "REQ.ID",
            "CompanyName": {
              "Code": "TN"
            }
          }
        }
      ]
    }
  }
}
```

```

    }
  ],
  "OriginDestinationInformation": [
    {
      "RPH": "1",
      "DepartureDateTime": "2019-01-03T00:00:00",
      "OriginLocation": {
        "LocationCode": "NYC"
      },
      "DestinationLocation": {
        "LocationCode": "LAX"
      },
      "TPA_Extensions": {
        "SegmentType": {
          "Code": "0"
        }
      }
    }
  ],
  "TravelPreferences": {
    "TPA_Extensions": {
      "NumTrips": {
        "Number": 10
      }
    },
    "DataSources": {
      "NDC": "Enable",
      "ATPCO": "Disable",
      "LCC": "Disable"
    }
  },
  "TravelerInfoSummary": {
    "SeatsRequested": [
      1
    ],
    "AirTravelerAvail": [
      {
        "PassengerTypeQuantity": [
          {
            "Code": "ADT",
            "Quantity": 1
          }
        ]
      }
    ],
    "PriceRequestInformation": {
      "TPA_Extensions": {
      }
    }
  },
  "TPA_Extensions": {
    "IntelliSellTransaction": {
    },
  },

```

```

        "SabreAth": {
            "Value": "",
            "BinarySecToken": "",
            "ConversationID": ""
        }
    }
}

```

USE CASE 2 (*Bargain Finder Max request and response for simple one-way itinerary to return solutions in which content source for ATPCO only is requested*)

Example Request:

```

{
  "OTA_AirLowFareSearchRQ": {
    "TruncateMessages": false,
    "Target": "Production",
    "Version": "0.0.1",
    "POS": {
      "Source": [
        {
          "PseudoCityCode": "XXXX",
          "RequestorID": {
            "Type": "0.AAA.X",
            "ID": "REQ.ID",
            "CompanyName": {
              "Code": "TN"
            }
          }
        }
      ]
    },
    "OriginDestinationInformation": [
      {
        "RPH": "1",
        "DepartureDateTime": "2019-01-03T00:00:00",
        "OriginLocation": {
          "LocationCode": "NYC"
        },
        "DestinationLocation": {
          "LocationCode": "LAX"
        },
        "TPA_Extensions": {
          "SegmentType": {
            "Code": "O"
          }
        }
      }
    ],
    "TravelPreferences": {
      "TPA_Extensions": {

```

```

        "NumTrips": {
            "Number": 10
        },
        "DataSources": {
            "NDC": "Disable",
            "ATPCO": "Enable",
            "LCC": "Disable"
        }
    },
    "TravelerInfoSummary": {
        "SeatsRequested": [
            1
        ],
        "AirTravelerAvail": [
            {
                "PassengerTypeQuantity": [
                    {
                        "Code": "ADT",
                        "Quantity": 1
                    }
                ]
            }
        ],
        "PriceRequestInformation": {
            "TPA_Extensions": {
            }
        }
    },
    "TPA_Extensions": {
        "IntelliSellTransaction": {
        },
        "SabreAth": {
            "Value": "",
            "BinarySecToken": "",
            "ConversationID": ""
        }
    }
}

```

In the below example, we have an extract of a BFM response for simple one-way itinerary to return solutions in which only the content source for ATPCO is requested. You can see an Offer with a unique offerId is returned by ATPCO, which will be stored for 35 minutes in a database.

Example Response (extract):

```
"itineraryGroups": [
  {
    "groupDescription": {
      "legDescriptions": [
        {
          "departureDate": "2019-01-01",
          "departureLocation": "LAX",
          "arrivalLocation": "JFK"
        }
      ]
    },
    "itineraries": [
      {
        "id": 1,
        "pricingSource": "ADVJR1",
        "legs": [
          {
            "ref": 1
          }
        ],
        "pricingInformation": [
          {
            "pricingSubsource": "MIP",
            "offer": {
              "offerId": "cdjnxalksm0-1",
              "ttl": 35,
              "source": "ATPCO"
            },
            "fare": {
              "offerItemId": "cdjnxalksm0-1-1",
              "validatingCarrierCode": "DL",
              "vita": true,
              "eTicketable": true,
              "lastTicketDate": "2018-11-01",
              "governingCarriers": "DL",
              "serviceId": "cdjnxalksm0-1-1-1",
              "passengerInfoList": [
                {
                  "passengerInfo": {
                    "passengerType":
"ADT",
                    "passengerNumber": 1,
                    "nonRefundable":
true,
                    "fareComponents": [
```

```

{
  "ref": 5,
  "segments": [
    {
      "segment": {
        "bookingCode": "E",
        "cabinCode": "Y",
        "seatsAvailable": 4,
        "availabilityBreak": true
      }
    }
  ],
  "ref": 4,
  "segments": [
    {
      "segment": {
        "bookingCode": "E",
        "cabinCode": "Y",
        "mealCode": "D",
        "seatsAvailable": 4,
        "availabilityBreak": true
      }
    }
  ],
  "taxes": [
    {
      "ref": 9
    },
    {
      "ref": 16
    },
    {
      "ref": 4
    },
    {
      "ref": 12
    },
    {
      "ref": 5
    }
  ]
}

```

```

        "ref": 7
    },
    {
        "ref": 2
    }
],
"taxSummaries": [
    {
        "ref": 5
    },
    {
        "ref": 1
    },
    {
        "ref": 2
    },
    {
        "ref": 3
    }
],
"currencyConversion": {
    "from": "USD",
    "to": "USD",
    "exchangeRateUsed": 1.0
},
"fareMessages": [
    {
        "type": "N",
        "code": "0",
        "carrier": "DL",
        "info": "NONREF/NOCHGS/NOPRE RSV DSEAT"
    },
    {
        "type": "W",
        "code": "0",
        "info": "VALIDATING CARRIER - DL"
    }
],
"passengerTotalFare": {
    "totalFare": 188.4,
    "totalTaxAmount": 39.56,
    "currency": "USD",
    "baseFareAmount": 148.84,
    "baseFareCurrency": "USD",
    "equivalentAmount": 148.84,
    "equivalentCurrency": "USD",
    "constructionAmount": 148.84,
    "constructionCurrency": "USD",
    "commissionPercentage": 0.0,
    "commissionAmount": 0.0,
    "exchangeRateOne": 1.0,
    "totalTtypeObFee": 0.0
}
}
}
}

```

```
],
"totalFare": {
  "totalPrice": 188.4,
  "totalTaxAmount": 39.56,
  "currency": "USD",
  "baseFareAmount": 148.84,
  "baseFareCurrency": "USD",
  "constructionAmount": 148.84,
  "constructionCurrency": "USD",
  "equivalentAmount": 148.84,
  "equivalentCurrency": "USD",
  "totalTtypeObFee": 0.0
},
"validatingCarriers": [
  {
    "ref": 1
  }
]
}
]
}],
```


API Reference

3.1 The beginner's guide to Air Shopping

What is required in my shopping request?

Please see attached samples.



E2E_Sample_OfferShop.zip

Looking for latest features and enhancements to Sabre APIs?

You are in the right place, check the [Release Notes](#).

Sabre APIs currently supports *up to 5* versions of an API.

Check the [API Versioning page](#).

3.2 Schema definitions

For information about parameter types and constraints, please refer to SDS resources for BargainFinderMax API.

3.3 Parameters introduced to the Request schema

3.3.1 DataSources: *object*

A parameter to choose which content source to drive at the request level. You will be able to drive all content source ATPCO/LCC/NDC, NDC Only, or ATPCO Only, or any combination of them, overriding current default.

ATPCO: *string*

Enables/Disables ATPCO content to be returned in the shopping response.

LCC: *string*

Enables/Disables LCC content to be returned in the shopping response.

NDC: *string*

Enables/Disables 'default' NDC content to be returned in the shopping response.

Example of an Offer request for NDC content only:

```
},
"DataSources": {
  "NDC": "Enable",
  "ATPCO": "Disable",
  "LCC": "Disable"
}
```

3.3.2 PreferNDCSourceOnTie: *object*

Allows sabre agency to choose which content source to drive at the request level. Customer will be able to drive all content sources ATPCO/LCC/NDC, or NDC Only or ATPCO Only, or any combination of them, overriding current default.

Value: *boolean*

Example of Prefer NDC Source on Tie:

```
},
"PreferNDCSourceOnTie": {
  "Value": true
}
}
```

3.4 Parameters introduced to the Response schema

3.4.1 Offer: *object*

An Offer that contains NDC, ATPCO, or LCC Offer data.

offerId: *string* (required)

The unique identifier of an Offer within the context of one message. An OfferId is assigned to an itinerary's price point.

Example: "cdjnxalks":

ttl: *integer* (required)

Time to Live in minutes. Specifies how long an offer is stored.

Example: 35

source: *string*, must match (ATPCO)|(LCC)|(NDC) (required)

Example: NDC, ATPCO, LCC

offerItemId: *string*

The unique identifier of the OfferItem instance. OfferItem is a priceable chunk of services.

Example: "cdjnxalksm0-1-1"

serviceId: *string*

The unique identifier of a service within the context of one message. A service is a feature/service included in a given flight.

Example: "cdjnxalksm0-1-1-1"

Example of a response with an Offer returned with NDC content only. You can identify the source of the content by looking at the value returned in the parameter **e.g.** "source": "NDC".

```
"itineraryGroups": [
  {
    "groupDescription": {
      "legDescriptions": [
        {
          "departureDate": "2019-01-03",
          "departureLocation": "EWR",
          "arrivalLocation": "LAX"
        }
      ]
    }
  },
  "itineraries": [
```

```

{
    "id": 1,
    "pricingSource": "ADVJR1",
    "legs": [
        {
            "ref": 13
        }
    ],
    "pricingInformation": [
        {
            "pricingSubsource": "NDC_CONNECTOR",
            "offer": {
                "offerId": "dx25jxjzpjpmqokbu0-9",
                "ttl": 29,
                "source": "NDC"
            },
            "fare": {
                "offerItemId": "dx25jxjzpjpmqokbu0-9-1",
                "validatingCarrierCode": "",
                "eTicketable": true,
                "serviceId": "dx25jxjzpjpmqokbu0-9-1-1",
                "passengerInfoList": [
                    {
                        "passengerInfo": {
                            "passengerType": ""
                        },
                        "passengerNumber": 1,
                        "fareComponents": [
                            {
                                "ref": 17,
                                "segments": [
                                    {
                                        "segment": {
                                            "bookingCode": "B",
                                            "cabinCode": "Y"
                                        }
                                    }
                                ],
                                "segment": {
                                    "bookingCode": "B",

```

```

        "cabinCode": "Y"
    }
}

],
}

"taxes": [
    {
        "ref": 18
    },
    {
        "ref": 13
    },
    {
        "ref": 26
    },
    {
        "ref": 24
    },
    {
        "ref": 26
    },
    {
        "ref": 24
    }
],
"passengerTotalFare": {
    "totalFare": 345.81,
    "totalTaxAmount": 45.34,
    "currency": "USD",
    "baseFareAmount": 300.47,
    "baseFareCurrency": "USD",
    "equivalentAmount": 300.47,
    "equivalentCurrency": "USD"
}
}
}
},
"totalFare": {
    "totalPrice": 345.81,
    "totalTaxAmount": 45.34,
    "currency": "USD",
    "baseFareAmount": 300.47,
    "baseFareCurrency": "USD",
    "equivalentAmount": 300.47,
    "equivalentCurrency": "USD"
},
"validatingCarriers": [
]
}
}

```

```
]
}
```

Error Messages and Status Codes

4.1 Introduction

4.1.1 API Specific Errors

Bargain Finder Max

#	Error	Cause	Resolution
1	No Availability	Either your PCC is not enabled to consume BFM or the tier level (50, 100, 200) you specified does not match your BFM agreement.	1. Ensure your PCC is enabled to consume BFM. Contact your Sabre account manager to validate if your PCC has been configured to consume BFM. 2. Adjust the BFM tier level to the appropriate one according to your BFM agreement.
2	NO FLIGHT SCHEDULES FOR QUALIFIERS USED	No flights results could be found with the qualifiers used in your request.	Review the qualifiers used on your request to ensure those are not overly-restrictive – Example, requesting non-stop flights in a route where non-stop flight do not operate.
3	Communication error code=104: Connection reset by peer	Temporary internal processing error.	1. Retry your request. 2. Contact Sabre support if error persists.
4	Communication error: Transmission timeout	Temporary internal processing error.	1. Retry your request. 2. Contact Sabre support if error persists.
5	DSF server returned an error: Insufficient resources to process the request	Temporary internal processing error.	1. Retry your request. 2. Contact Sabre support if error persists.

#	Error	Cause	Resolution
6	MAX NBR COMBINATIONS EXCEEDED	Temporary internal processing error.	1. Retry your request. 2. Contact Sabre support if error persists.

4.1.2 REST APIS Common errors

REST APIs: Verify your token credentials were set-up correctly.

#	HTTP Status Code	Message	Text	Resolution
1	400	Bad Request	Invalid format for request	1. Verify your request includes valid parameters or values. 2. Verify the grant_type in the request payload.
2	400		ERR.2SG.CLIENT.INVALID_REQUEST	Verify your request includes valid parameters or values. Check API documentation.
3	401	Unauthorized	Not authorized to make this request. Check access level and refer to API documentation for access information.	Verify your token credentials.
4	401	Unauthorized	invalid_client	Verify your token credentials.
5	401	Unauthorized	Credentials are missing or the syntax is not correct	Verify your base64-encoded token credentials

#	HTTP Status Code	Message	Text	Resolution
				were constructed properly.
6	401	Unauthorized	Wrong clientID or clientSecret	Verify password for the client ID.
7	401	Unauthorized	ERR.2SG.SEC.MISSING_CREDENTIALS	Verify you are using the correct data type.
8	401	Unauthorized	ERR.2SG.SEC.INVALID_CREDENTIALS	Verify your token credentials.
9	403	Forbidden	Request is for a resource that is forbidden	<p>1. Verify your token credentials allow authorization to call this API.</p> <p>2. Contact your Sabre account manager to verify access.</p>
10	403		ERR.2SG.SEC.NOT_AUTHORIZED	<p>1. Verify your token credentials allow authorization to call this API.</p> <p>2. Contact your Sabre account manager to verify access.</p>
11	403		ERR.2SG.CLIENT.SERVICE_UNKNOWN	<p>1. Verify URL is valid.</p> <p>2. Verify whether your request is</p>

#	HTTP Status Code	Message	Text	Resolution
				required segments, such as a version number. Check API documentation.
1 2	404	Not Found	Response does not contain any data	<p>1. No data found to fulfill the request. You might to consider using fewer parameters/filters.</p> <p>2. URL has missing segments, such as a version number. Check API documentation.</p> <p>3. Verify URL is valid. Check API documentation.</p>
1 3	405	Method Not Allowed		Method is not valid or authorized for this endpoint. Check API documentation.
1 4	406	Not Acceptable	Requested resource is only capable of generating content not acceptable according to the Accept headers sent in the request.	

#	HTTP Status Code	Message	Text	Resolution
15	413		ERR.2SG.CLIENT.INVALID_REQUEST	<ol style="list-style-type: none"> 1. Request URL too long. 2. Reduce the number of request parameters in the URL. Separate parameters into multiple requests.
16	413	FULL head		<ol style="list-style-type: none"> 1. Request URL too long. 2. Reduce the number of request parameters in the URL. Separate parameters into multiple requests.
17	429	too many requests	temporarily_unavailable	<ol style="list-style-type: none"> 1. Internal limit has been exceeded. 2. Wait at least 500 milliseconds and resend the request.
18	429	Throttled	Active token count is exceeded	<ol style="list-style-type: none"> 1. Maximum number of concurrent requests for the API have been exceeded. Contact your Sabre account manager to

#	HTTP Status Code	Message	Text	Resolution
				<p>determine or increase your allocated concurrent request limit for this API.</p> <p>2. Wait at least 500 milliseconds and resend the request.</p>
19	429		ERR.2SG.GATEWAY.REQUEST_THROTTLED	<p>1. Maximum number of concurrent requests for the API have been exceeded. Contact your Sabre account manager to determine or increase your allocated concurrent request limit for this API.</p> <p>2. Wait at least 500 milliseconds and resend the request.</p>
20	500	Server Error		<p>1. API encountered an unexpected condition that prevented it from fulfilling the client's request.</p> <p>2. Wait at least 500</p>

#	HTTP Status Code	Message	Text	Resolution
				milliseconds and resend the request.
21	500		ERR.2SG.SEC.INTERNAL_PROCESSING_ERROR	<p>1. API encountered an unexpected condition that prevented it from fulfilling the client's request.</p> <p>2. Wait at least 500 milliseconds and resend the request.</p>
22	500		ERR.2SG.GATEWAY.TIMEOUT	Wait at least 500 milliseconds and resend the request.
23	500		ERR.2SG.GATEWAY.INTERNAL_PROCESSING_ERROR	Wait at least 500 milliseconds and resend the request.
24	500		ERR.2SG.GATEWAY.INVALID_PROVIDER_RESPONSE	1. The server cannot understand the response of the service because of bad format or failure to comply with protocol. Contact the Sabre support desk.

#	HTTP Status Code	Message	Text	Resolution
				2. Wait at least 500 milliseconds and resend the request.
25	500		ERR.2SG.GATEWAY.PROVIDER_CONNECTION_ERROR	<p>1. Your request encountered a transport error.</p> <p>2. Wait at least 500 milliseconds and resend the request.</p>
26	503	Service Unavailable	Server is currently unavailable. Please try again later and report if problem persists.	Wait at least 500 milliseconds and resend the request.
27	504	Gateway Timeout	Server timed out trying to fulfill request. Please try again later and report if problem persists.	Wait at least 500 milliseconds and resend the request.

4.2 Status Codes

RESTful services use HTTP status codes to provide the result status of the executed service. For example, 200 for success, 404 when a resource is not found.

HTTP status codes are defined by section [10 of RFC 2616](#), the Internet Assigned Numbers Authority (IANA) maintains the official [registry of HTTP status codes](#).

Range	Meaning
2xx	Successful operation. Indicates that a request was successful, of which can succeed in multiple ways.
4xx	Client-side error. Usually, it indicates that there was a problem with the data sent in the request, but can also indicate infrastructure-related issues, e.g. authentication, throttling. In most cases the client can modify their request and resubmit.
5xx	Server-side error. Indicates a valid request was sent by the client, but it could not be processed due to software defects or external conditions like downtime system unavailability. 5xx range status codes should not be utilized for validation or logical error handling.

• • •

Glossary

Term	Description
2SG	2nd Generation Services Gateway
ATPCO	Airline Tariff Publishing Company
BFM	Bargain Finder Max
IATA	International Air Transport Association
NDC	New Distribution Capability, IATA xml communication standard.
OTA	Online Travel Alliance
USG	Universal Service Gateway