Search to Booking Workflow

Static Data

You can download the master data of cities, areas or hotels through the API. Kindly refer to Static Data section in the API document.

Search Request

There are few important parameters in the search request which when used as suggested could help you to retrieve the results faster.

Hotel Code Search

To download hotels from area code, kindly download all cities associated with area. From the cities, you can download associated hotel codes.

Rates

This parameter can contain one of the 2 values

a. Concise
b. Comprehensive

Concise (Recommended):
The default is Concise under which we will return you the lowest rate for all qualifying hotels. We suggest you to use Concise parameter and mandatory to send 200-250 hotel Ids per search, you can send parallel searches (the lower the batch size, the lower would be response time)

Comprehensive (Not Recommended):
If your model doesn't allow you to search with concise rates then you can use comprehensive option, wherein we will return you all rates for all qualifying hotels. Under Comprehensive parameter it is mandatory to restrict the number of Hotel Ids between 100-150 hotel ids per search, you can send parallel searches. (The lower the batch size, the lower would be response time)

Cutoff Time and More results parameter

Cutoff time is important if you would like to receive responses faster and does not want to wait till all suppliers return the data. In this case, based on your requirement, you can configure the **cutoff_time** anywhere between 8000 (8 secs) to 50000 (50 secs).

Based on the **cutoff_time** that you have mentioned in the search request, we return the results back to you. Kindly note the result set that you receive (when cutoff_time is mentioned) might not always be the complete result set.

Recommended time is 15 seconds.

You can know whether the result set that you have received is complete or partial by looking at the parameter **more_results** in the search response. If it's true, then it means that the API is still fetching the

results and you shall call refetch API to get more search results. If it's false, then it means you have received all the data.

The attribute **more_results** can also be passed in the search request to inform API whether to get more results from the supplier even after the cutoff_time or not. If **more_results: false** is provided in the search request, then you will get the data from suppliers based on the mentioned cutoff_time and API will also stop getting further data from the suppliers.

If more_results is equal to true then fetch more results through Refetch by search ID endpoint.

Sample Payload

Here is how you can use these parameters to get faster search results. Refer the highlighted items in the search request.

```
{
  "hotel_codes": ["H!0049396", "H!0018105"],
  "checkin": "2018-12-29",
  "checkout": "2018-12-30",
  "client_nationality": "IN",
  "cutoff_time": 10000,
  "currency": "INR",
  "hotel_info": false,
  "rates": "concise" ,
  "hotel_category": [3, 5],
  "rooms": [
  {
    "adults": "1"
  },
  {
    "adults": "2",
    "children_ages": ["3"]
  }
]
```

Search Response

- The above search payload does a hotel code search with cutoff_time as 10 secs.
    - Since the rates:concise is requested, only 1 rate per hotel will be returned in the search.
- If "more_results" value in search response is true, then refetch rates by search ID. If you have defined lesser cutoff_time, then instead of doing refetch multiple times to get all the inventory, you can do an API call to know if all data is available at the HUB. You can do this by calling the following endpoint api/v3/hotels/availability/<sid>/more_results
    - This will tell you whether all results are available. If yes more_results will be false, otherwise, this will be true. Along with this, it will also give you the number of hotels available at the hub. Based on the number of hotels available at the HUB and value of more_results, you can decide when to call refetch API.
- If hotel is already selected from the initial search then refetch rates by hotel ID to get all the rates in that hotel (Refer to 8.3 section in API Document).
    - You can pass the parameter "bundled=true" in the above request to get bundled rates for that hotel.

In the search response, you will get the details of list of hotels that matches the requested search criteria.

- If you pass **rates:concise** in the search request, you will get only the lowest rate (one rate) for the requested criteria for each hotel.
- If you pass **rates:comprehensive** in the search request, you will get a set of rates for the requested criteria for each hotel.
- If you have requested for multiple rooms, then in the search response, you will get results as bundled or non-bundled or both.

Bundled rates: It means you will get a single price for the total rooms requested.

Non-Bundled rates: You will get the total price per room.

So in search response, few results, you might get as bundled and few as non-bundled.

To identify if a particular rate is bundled or non-bundled, look for **no_of_rooms** inside Rate array. If the value in **no_of_rooms** inside Rate array is equal to the number of rooms requested in search, then that rate is considered as bundled. Otherwise, it is considered as non-bundled.

So while booking, if you are going to choose a room, which is non-bundled, then the 2nd room that you will select for booking should always have the same **group_code** like the 1st room selected. This is important. (If you are searching for 3 rooms and select 1st room from non-bundled rates, then the next 2 rooms also should have same **group_code** and should be from non-bundled rates only.)

If you select bundled room, then you can directly use that for booking as it contains the total rooms within that single rate.

- In the search response, the room occupancy can be shown either in the form of “no_of_adults” or in the form of “max_room_occupancy”. (Refer to room object description in the section 8.1 in the API doc)
- In the search response, for each rate, you will have one of these three values for the **rate_type** field:
- If rate_type is **refetch,** then it is mandatory to perform refetch by using that particular hotel_id and search id (refer to section 8.3 in API doc) before doing the booking.
- If rate _type is **recheck,** then it is mandatory to perform rate_recheck by using rate_key and group_code (Refer to section 8.5 in API doc) before doing the booking.
- If rate_type is **bookable,** it is recommended to perform **rate_recheck** before proceeding to booking. This will minimize the risk of paying higher price because of last minute price updates from upstream suppliers.
- If rate_recheck/refetch is performed, **the values (rate_key, group_code, room code, room reference, etc) of rate_recheck/refetch response should be used in the booking request instead of the values from the search response.**
- If the rate has **non_refundable: true,** then it means the rate is non-refundable and you shall ignore cancellation policies (if any) for this rate.

Cancellation Policy

- In the cancellation policy, **under_cancellation:false** means that the booking is not under cancellation. So you can cancel the booking till the date mentioned in the **cancel_by_date** field without any cancellation charges. There could be multiple cancellation policy dates and charges associated for few rates. Kindly refer to API document for more details.
- If **under_cancellation:true,** then the booking is already under cancellation and charges will be applied based on the cancellation policy.

- **amount_type** field in the cancellation_policy details can have any of the three values -- amount, percent, no of nights. If the amount_type value is returned in the form of either percent or no of nights, then the cancellation charges should be shown in the same format only and should not be converting to actual amount. This is important because different suppliers calculate charges on different parameters.
- If the cancellation policy code is returned in the search response instead of the cancellation_policy, then you can fetch the cancellation details by using fetch cancellation details endpoint (refer to section 8.4 in the document for sample payload).

Booking Request and Response

- You can place the booking request with the values returned from search/rate_recheck/refetch response (refer to section 9 in the document for sample payload).
- If you have chosen **bundled** rate from the search response, then you need to place only one booking item in the booking request with the paxes details in multiple rooms. Refer to booking request in the API document.
- If you have chosen **non_bundled** rates from the search response, then you need to place one booking item for each room in the booking request with the paxes details. Refer to booking request in the API document.
- If the room_refernce value is returned in the search response, then you need to add the room_refernce in the booking item. Otherwise, you can ignore that field.
- In the booking response, you will get **status** as **confirmed** for successful booking.
- Once the booking is successful, you can get the booking details using view booking details endpoint (refer to section 14 in the document).
- The booking status **pending** indicates, the booking is under process at the supplier's end. To get the latest status of the booking, you have to call view bookings end-point.

Cancellation Request and Response

- If booking is successful, you can cancel the booking by using cancellation endpoint (refer to section 12 in the document for sample payload).
- If the booking you try to cancel is **under cancellation,** the cancellation charges will be applied.
- **status: confirmed** in the cancellation response indicates that the booking is cancelled successfully.