

Commercial Guide API Booking.com

Introduction

This guide is intended primarily for developers, as a supplement to the technical documentation. The content of this guide will allow you to have a more complete understanding of how you can use the information obtained from the calls made from our API repositories.

We suggest to use the technical and this commercial documentation always together while developing the website.

Content

- 1 Getting Started
 - 1.1 Authentication
 - 1.2 Sample URLs
 - 1.3 Output Format
 - 1.4 Pagination
- 2 Usage Information
 - 2.1 Endpoints
- 3 How to Book
 - 3.1 Linking to Booking.com
- 4 Permitted Use of the API
 - 4.1 Maximum API Connections
 - 4.2 Price Comparison
 - 4.3 No data Forwarding
 - 4.4 Caching of availability content or heavy caching of static content
 - 4.5 Prohibitions
 - 4.6 Current Data
 - 4.7 Closed properties
- 5 Implementation and Best Practices

1 Getting Started

1.1 Authentication

When making requests from our repositories you will be asked for sign up details.

This is part of our HTTP Basic Authentication. You can use the username and password which will be given to you if access has been granted to the API resources.

1.2 Sample URLs and Output

General syntax:

`https://[domain]/[version]/[xml or json]/[endpoint]?[parameters]`

JSON and XML examples for v2.0:

- GET `https://distribution-xml.booking.com/2.0/json/hotels?hotel_ids=10004`
- GET `https://distribution-xml.booking.com/2.0/xml/hotels?hotel_ids=10004`

The full technical documentation contains JSON URL samples for every endpoint.

The type of output differs according to the selection of either XML or Json.

Json output example:

```
{
  result => [
  ],
  errors => [
    {
      code => 1000,
      message => 'Sample error message'
    }
  ],
  warnings => [
    {
      code => 1001,
      message => 'Sample warning message'
    }
  ],
  request_params => {
  }
}
```

XML output example:

```
<EndpointName>
  <result></result>
  <result></result>
  <errors>
    <code>1000</code>
    <message>Sample error message</message>
  </errors>
  <warnings>
    <code>1001</code>
    <message>Sample warning message</message>
  </warnings>
  <request_params></request_params>
</EndpointName>
```

Please note: To add `request_params` (requested parameters) in the output, `show_request_params=1` input parameter needs to be set. This can be useful for debugging.

1.4 Pagination

Pagination is done with **offset** and **rows** input parameters.

- **offset:** Where in the resulting list to start.
- **rows:** The maximum number of entries to return.

Usage Examples:

Get first 100 entries: **cities?offset=0&rows=100**

Get the next 100 entries starting from the 100th row: **cities?offset=100&rows=100**

2 Usage Information

This part of the documentation will provide a general explanation about the type of endpoints available in the API repositories and their basic usage.

For detailed information about the specific endpoints and calls please check the technical documentation.

2.1 Endpoints

The API consists of three general endpoint classes:

- **Availability:** real-time pricing and availability which is constantly changing. It is not permissible to cache availability. Instead, request the data each time it is needed. In general, we suggest to request this type of calls in real time.
- **Static data:** property descriptions, facility information, city names, hotel types, and other data that changes less frequently in comparison to availability. It is recommended to cache this data. This type of information is updated less often, this means that you can download the data in your machine and upload it to your website through CSV. We suggest to refresh the information that you cache once a week or every few days.
- **Booking endpoints:** these are related to processing bookings via the API. In other words, in case you are eligible to process reservations and payments on your website, you can use direct Booking endpoints.
Please note: This type of API implementation is unfortunately not supported. At the moment, you can only make use of the static and availability endpoints.

Destination information with translations can be obtained from the following endpoints:

- **countries**
- **cities**
- **districts**
- **regions**

English is the default language and the translations are not available for every language since the returning value might be the same in English.

In any case, do use the language to be applied on your website, if no value is returned then use the English value.

For example, the translation of "Amsterdam" in Dutch is the same as the English translation, therefore no Dutch translation will be returned.

Using changedHotels and the hotels endpoint

To build a cache of property data for the first time, request all desired static content from the **hotels** endpoint.

For example, if you have a website that focuses on properties in the Netherlands, first use the **cities** endpoint and then use the **hotels** endpoint with the **city_ids** parameter. This will return data for all cities in the Netherlands.

After the initial data load into your database, use the **changedHotels** endpoint to get a list of properties that have been changed or closed since the given timestamp.

Important: If a property is marked as closed, you must remove all related data from your websites, apps, and databases. This is because closed properties may reopen and appear in the change list again. This means that you can show twice the same information for hotels that now have different names, confusing for your customers.

Once you have completed the basics with the cachable calls (hotels and cities for example) and you have compared them against the list of hotels that have been closed or changed, you can pass to the availability endpoints. Availability endpoint allow you to request more dynamic calls, and are used for updated pricing and availability in connection to the hotels.

The **hotelAvailability** endpoint should generally be used first to provide the least expensive rooms for the provided guest configuration and search parameters. This is comparable to the initial property search on Booking.com.

After a property is selected, **blockAvailability** can be used to return all available rooms for a specific property. This is comparable to the property page on Booking.com where rooms are selected.

Please remember: All availability endpoints should be used in real-time and never cached. The other endpoints can be used less frequently:

Once a day:

- **changedHotels** to receive a list of properties with changed data
- **hotels** to update the changed hotels

Once a week:

- The "types" endpoints such as **hotelFacilityTypes** and **hotelTypes**
- The destination endpoints such as **countries** and **cities**

3 How to Book

Reservations can be completed through the API via linking to Booking.com reservation system. In other words, the information uploaded on your website through API calls will have links that redirect to our engine where the customer can complete the reservation.

3.1 Linking to Booking.com

Important: These examples must be altered before using them in a production environment. Change the AID value and remove any test parameters. Contact support for more information.

Stage 0: Check availability

https://secure.booking.com/book.html?aid=303532&hostname=www.booking.com&hotel_id=98251&test=1

Stage 1: Select room

https://secure.booking.com/book.html?aid=303532&hostname=www.booking.com&hotel_id=98251&stage=0&checkin=2018-01-01&interval=2&test=1 **Note:** If no rooms are available the visitor will be redirected to stage 0.

Stage 2: Your details

https://secure.booking.com/book.html?aid=303532&hostname=www.booking.com&hotel_id=98251&stage=1&checkin=2018-01-01&interval=1&nr_rooms_9825102_80154322_0=2&nr_rooms_9825102_80154322_0=1&test=1

4 Permitted Use of the API

This section will explain what are the main limitations to the usage of the API calls.

Penalties for infringement of these rules can lead to the loss of the privileges related to the API usage via access termination, and in case of severe or severe damage to the Booking.com copyright content usage or any other clause stated in the Affiliate Agreement; termination of the account.

Please note: Booking.com does not accept any liability for loss of revenue caused by disabling API access for abuse reasons.

4.1 Maximum API Connections

The maximum number of connections allowed is 20 per API user. An automated process will terminate API access in case a user exceeds a specific number of requests per minute. This is done in order to avoid bots usage.

Please note: The limit should never be reached through normal usage and it cannot be increased.

4.2 Price Comparison

"Price comparison" refers to the comparison of accommodation prices made available from two or more online booking platforms. If you wish to use Booking.com properties for the purposes of price comparison, **you are not permitted** to use any of the content provided by Booking.com or made available through the Booking.com website. This includes the property description, photos, details of facilities, policies and more. As such, you are required to use your own content to appropriately describe each property.

4.3 No data Forwarding

Data Forwarding describes the act of providing (forwarding) Booking.com information to third parties that are not affiliated or part of the forwarding entity. This procedure is strictly forbidden.

4.4 Caching of availability content or heavy caching of static content

You are not allowed to cache availability or prices. With the huge number of properties on Booking.com, availability and pricing changes very rapidly, so any cache of this data will be inaccurate very quickly. This is to ensure the presented information is always as accurate as possible in order to provide the best offer to your own customers.

4.5 Prohibitions

You are not allowed to alter the description provided via API. The content provided for the property description has been carefully curated by Booking.com to make sure it truly represents the property and its features. Changing any of this text is strictly forbidden as it may misrepresent the property.

You are not allowed to change the prices. The price returned in availability endpoints is the actual price the property has entered into our system and the price that bookers will pay. Changing the display of this price for any reason is strictly forbidden.

You are not allowed to alter or download property images. These images are provided directly from the accommodation, if they are tampered with or changed in any way it could mislead the user, and is therefore strictly forbidden.

Partners must store the URL to hotel images for use in HTML IMG tags. Downloading the image files themselves is not permitted.

4.6 Current Data

API users who are caching content are responsible for keeping the data current. This is to ensure content is updated or removed in cases where Booking.com may no longer have permission to use it, or when content is no longer valid. Failure to keep data current may result in complaints being brought against partners. Booking.com cannot take any responsibility for complaints and third-party claims about outdated content.

4.7 Closed properties

If a property has been flagged as closed, partners must remove all related data from their websites, apps, and/or databases.

5. Implementation and Best Practices

In this section we will lay out some best practices and tips for you to get started faster with the implementation of the API calls on your website. We have divided this in three areas: Website Architecture, Accommodation page organization, and testing.

a. Website Architecture.

What we usually suggest is to create a separate area for the accommodations. It is best to provide the option in a bar to the top of the page in this way it will be easier for your customers to be redirected to that area of your website. In case your website has traffic mainly on one page, it can be harder to shift traffic to the accommodation section. In this case we suggest to put wherever feasible links redirecting to the accommodation page. For example, if you have a page with many posts of different destinations, your readership might be used to browsing only that page. In order to modify this behavior, you can add a deep-link for each accommodation reference and direct it to the accommodation page.

Another choice to be made is how to organize the listings in the accommodation page.

There are two ways in which this can be addressed, the first is by providing a list of accommodations and their information, the second is to stream this even further by creating geographic locations.

The former is suggested in case the number of accommodation you want to display is limited to a maximum of 3 different countries. In this case we also suggest for these countries to border each other. This is because you will avoid confusing the clients on the choice provided.

Even if a list of accommodations is provided it is best to group them in some way. This can be done through geographic position, stars, type of accommodation or seasonality for example (summer vs winter).

The latter is suggested in case your website displays a wide array of accommodations in different countries.

In this situation it is best to create an order with:

- Continent/Area
- Countries per continent
- Regions
- Cities (if necessary)

b. Accommodation page organization

Each accommodation should have its own subpage; you can also use a drop down or a pop up.

In each accommodation there should be three separated sections:

- Pictures
- Description/ specs
- Prices and availability

Do not forget to add also the deep link redirecting to Booking.com where the reservations can be finalized.

We suggest to place it as a button in the prices and availability section.

c. Testing

Exactly as we do in Booking.com, we invite you to test your pages as much as possible in order to improve their performance.

Tests generally refer to changing:

- Colors: you can try to modify colors for different elements of the page, like buttons lines or separators
- Fonts: This is mainly related to text, You can try to use different sizes and fonts between names and descriptions for example
- Content as in pictures, text
- Sizes: for example enlarging a picture of an accommodation that you find better than the others, or increase the size of the price
- Order: you can change the order of the accommodation in the page, or the order of the pictures that are displayed.

Please note: the most important thing about testing is to make sure that you are able to track if a change has the expected returns. Before making a change estimate a measure like for example visitors. After a specific amount of time (for example two weeks) check if there are changes in the amount of visitors. Have they increased or dropped?