

# ibelsa API - Bookings

All of the endpoints listed here and others can be found in our online documentation:

<https://rooms.ibelsa.com/api/doc>

This document is intended as a suggestion. Sometimes it can be useful to use endpoints other than those listed here.

This documentation should always be used in conjunction with the [online documentation](#) for the ibelsa API.

## A) Walkin

It is possible to create new bookings.

### 1. Rate

**GET /api/get/allactiverates**

<https://rooms.ibelsa.com/api/doc#get--api-get-allactiverates>

The end point supplies all rates that can be used for a booking.

### 2. Availabilities

**GET /api/get/availabilities/{start}/{end}/{rate\_id}/{person\_count}**

[https://rooms.ibelsa.com/api/doc#get--api-get-availabilities-{start}-{end}-{rate\\_id}-{person\\_count}](https://rooms.ibelsa.com/api/doc#get--api-get-availabilities-{start}-{end}-{rate_id}-{person_count})

This endpoint provides the availability for the defined period and rate for all room categories. The "person\_count" parameter specifies the minimum occupancy. This value has an influence on the selection of the room category.

**Important:** The output takes place on a daily basis. If a booking is to be made in the period, it must be checked for each booking day whether the availability is greater than zero.

### 3. Create a reservation

**POST /api/set/create/reservation**

<https://rooms.ibelsa.com/api/doc#post--api-set-create-reservation>

**Important:** This endpoint does not check whether the availability for the booked days is greater than zero. Overbooking is therefore possible.

## 4. Add room to a reservation

### POST /api/set/add/room\_to\_reservation

[https://rooms.ibelsa.com/api/doc#post--api-set-add-room\\_to\\_reservation](https://rooms.ibelsa.com/api/doc#post--api-set-add-room_to_reservation)

Endpoint C3 only allows one room to be created.

This endpoint allows adding a room reservation to an existing main reservation.

## B) Retrieve existing bookings

### 1. Get bookings in a time period

#### GET /api/get/timespan/{entity}/{start}/{end}/{last\_modified}

[https://rooms.ibelsa.com/api/doc#get--api-get-timespan-{entity}-{start}-{end}-{last\\_modified}](https://rooms.ibelsa.com/api/doc#get--api-get-timespan-{entity}-{start}-{end}-{last_modified})

#### **{entity} = reservations**

Returns a list of all room reservations that fully or partially fall within the specified period.

#### **{last\_modified}**

Reduces the result list to bookings that were created or changed on this date or later. This is particularly useful if the data is to be saved in your own system and updated regularly.

Important fields from the data:

- **id**  
the internal ID of the room reservation
- **status**  
Possible values are: BOOKED, CHECKED\_IN, CHECKED\_OUT, NO\_SHOW, NO\_SHOW\_GUARANTEED
- **self\_pay**  
Possible values are: NO, ADDITIONAL, ALL  
**NO**: the recipient of the whole invoice of the reservation is the “invoice recipient” of the booking.  
**ALL**: the recipient of the whole invoice of the reservation is the main guest.  
**ADDITIONAL**: two invoices are issued. The invoice items of the room rate go to the “invoice recipient”. Any additional services go to the main guest.
- **invoice\_items**:  
a list of all invoice items for the reservation.
- **main\_reservation**:  
Here you will find, among other things, the details of the invoice recipient

## 2. Get booking by ID

**GET /api/get/single\_id/{entity}/{id}**

[https://rooms.ibelsa.com/api/doc#get--api-get-single\\_id-{entity}-{id}](https://rooms.ibelsa.com/api/doc#get--api-get-single_id-{entity}-{id})

a)

**{entity} = roomreservations**

Returns a single reservation according to its internal ID.

The structure of the output is the same as in point 1.

**{id}**

The internal ID of the room reservation.

b)

**{entity}=mainreservations**

Delivers the main reservation with all the room reservations it contains.

**{id}**

The internal ID of the main reservation

## 3. Get single entity by determined number

**GET /api/get/single\_number/{entity}/{number}**

[https://rooms.ibelsa.com/api/doc#get--api-get-single\\_number-{entity}-{number}](https://rooms.ibelsa.com/api/doc#get--api-get-single_number-{entity}-{number})

**{entity} = mainreservations**

**{number} -> reservation number of the main reservation, visible in the application.**

The main reservation is the superordinate construct that contains the room reservation or several room reservations. The reservation number is assigned to the main reservation.

This means that a reservation with several rooms has only one reservation number. We are talking about a group reservation.

See a diagram in the appendix that shows the relationships graphically.

The structure of the output is the same as in point 2.

## C) Edit contacts

### 1. Update existing contact

**POST /api/set/contactdata**

<https://rooms.ibelsa.com/api/doc#post--api-set-contactdata>

This endpoint enables the data of an existing contact to be changed. The internal ID of the contact must be known.

Due to current events (Corona), one contact field is described in more detail here:

**smart\_check** -> **yes / no**

After the guest status regarding Corona has been checked, this flag can be set accordingly. **yes** means that the corona check was carried out successfully. This flag should only be set for guests with a reservation. This status is indicated in the reservation in ibelsa.

This information is automatically deleted after the reservation is checked out. The **smart\_check** field then has no value

## 2. Update the invoice recipient of a reservation

**POST /api/set/roomreservation/invoicerecipient**

<https://rooms.ibelsa.com/api/doc#post--api-set-roomreservation-invoicerecipient>

This endpoint needs the internal ID of the room reservation. A room reservation is always assigned to exactly one main reservation. Because the invoice recipient is linked to the main reservation.

## 3. Handle additional guests of a reservation

**POST /api/set/roomreservation/handle\_additional\_guest**

[https://rooms.ibelsa.com/api/doc#post--api-set-roomreservation-handle\\_additional\\_guest](https://rooms.ibelsa.com/api/doc#post--api-set-roomreservation-handle_additional_guest)

This endpoint is not intended to change the data of an existing contact. All additional guests can be manipulated. The main guest remains unaffected.

Likewise, the number of guests for the room cannot be changed.

Three actions are possible:

- 1. remove**

The contact with the defined internal ID is removed as a guest of the reservation.

- 2. add**

The contact details of the new guest must be transmitted. If the contact does not exist, it will be created. Otherwise the existing contact will be used as an additional guest.

- 3. replace**

Works like "add", only that the guest is exchanged for another contact.

## D) Check in

### 1. Check in a reservation

**GET /api/set/roomreservation/checkin/{entity\_id}/{do\_register\_guest}**

[https://rooms.ibelsa.com/api/doc#get--api-set-roomreservation-checkin-{entity\\_id}-{do\\_register\\_guest}](https://rooms.ibelsa.com/api/doc#get--api-set-roomreservation-checkin-{entity_id}-{do_register_guest})

**{entity\_id}** -> internal ID of the room reservation

A reservation can be checked in with this endpoint. A clean room must already be assigned to the room. Otherwise it is not possible to check in.

### 2. Check in with room allocation

**GET /api/set/roomreservation/allocate/checkin/{entity\_id}/{do\_register\_guest}**

[https://rooms.ibelsa.com/api/doc#get--api-set-roomreservation-allocate-checkin-{entity\\_id}-{do\\_register\\_guest}](https://rooms.ibelsa.com/api/doc#get--api-set-roomreservation-allocate-checkin-{entity_id}-{do_register_guest})

**{entity\_id}** -> internal ID of the room reservation

Same function as in 1. If no room is assigned, a free, clean room is automatically assigned to the booking.

### 3. Allocate room to a reservation

**GET /api/set/allocate\_room/{entity\_id}**

[https://rooms.ibelsa.com/api/doc#get--api-set-allocate\\_room-{entity\\_id}](https://rooms.ibelsa.com/api/doc#get--api-set-allocate_room-{entity_id})

The endpoint makes it possible to automatically assign a free, clean room to a reservation that does not yet have a room.

## E) Upselling

### 1. All available products

**GET /api/get/products/available**

<https://rooms.ibelsa.com/api/doc#get--api-get-products-available>

This endpoint provides a list of products that can be booked via the ibelsa API. These products must be labeled accordingly in ibelsa so that they appear here.

## 2. All optional package products

**GET /api/get/products/optional/{rate\_id}**

[https://rooms.ibelsa.com/api/doc#get--api-get-products-optional-{rate\\_id}](https://rooms.ibelsa.com/api/doc#get--api-get-products-optional-{rate_id})

**{rate\_id}** -> internal rate-id that is used in the booking

The end point delivers all optional products that are linked to this rate and can therefore be booked.

## 3. Add products to the reservation

**POST /api/set/add/products**

<https://rooms.ibelsa.com/api/doc#post--api-set-add-products>

With this endpoint, products can be booked for a reservation. To do this, the internal ID of the reservation must be known. The products can be obtained from the endpoints described above.

## F) Invoices

See a diagram at the end that shows the relationships graphically.

Before a reservation can be checked out, all open invoices must be closed.

Some of the following endpoints require either an invoice ID or the invoice item ID.

One way to do this is to take this information from the endpoints described above (B 2, B 3).

### 1. Payment methods and invoice templates

**GET /api/get/hotel/payment**

<https://rooms.ibelsa.com/api/doc#get--api-get-hotel-payment>

This endpoint provides the ID and name of all payment methods and invoice templates. In consultation with the hotel, a decision must be made which will be used when closing invoices.

The information of this end point is required in the following two end points (2, 3).

### 2. Close an invoice

**GET /api/set/invoice/paid/{entity\_id}/{template\_id}/{payment\_method\_id}/{send\_invoice}**

[https://rooms.ibelsa.com/api/doc#get--api-set-invoice-paid-{entity\\_id}-{template\\_id}-{payment\\_method\\_id}-{send\\_invoice}](https://rooms.ibelsa.com/api/doc#get--api-set-invoice-paid-{entity_id}-{template_id}-{payment_method_id}-{send_invoice})

**{entity\_id}** -> internal id of the invoice

**{send\_invoice}** -> should ibelsa send the invoice document?

If the invoice recipient has an email address, ibelsa can send the invoice document there.

### 3. Pay part of the bill

**POST /api/set/roomreservation/paidpartial**

<https://rooms.ibelsa.com/api/doc#post--api-set-roomreservation-paidpartial>

Via this end point, invoice items can be moved from one invoice to a new invoice and then automatically closed. ibelsa can send the invoice document to the invoice recipient on request.

Of course, you can only move invoice items from open invoices.

### 4. Split invoice of a reservation

**POST /api/set/roomreservation/splitinvoice**

<https://rooms.ibelsa.com/api/doc#post--api-set-roomreservation-splitinvoice>

Via this end point, invoice items can be moved from one invoice to a new invoice.

Both invoices remain open.

### 5. Update the invoice recipient of an invoice

**POST /api/set/invoice/invoicerecipient**

<https://rooms.ibelsa.com/api/doc#post--api-set-invoice-invoicerecipient>

The invoice recipient on the invoice is taken from the reservation. If necessary, the invoice recipient can be changed on the invoice without changing the data in the reservation.

A change to the invoice recipient is only possible for invoices that have not yet been closed.

## G) Check out

### 1. Check out a reservation

**GET /api/set/roomreservation/checkout/{entity\_id}/{pre\_checkout}**

[https://rooms.ibelsa.com/api/doc#get--api-set-roomreservation-checkout-{entity\\_id}-{pre\\_checkout}](https://rooms.ibelsa.com/api/doc#get--api-set-roomreservation-checkout-{entity_id}-{pre_checkout})

There are two requirements for a successful checkout.

- The day of booking in ibelsa must coincide with the day of departure of the reservation (exception: `pre_checkout=yes`).
- All invoices must be closed.
  - In the case of a group booking, you can also check out if the invoice is open. The last booking of a group reservation can only be checked out after all invoices have been closed.

#### **{pre\_checkout}**

In the event of early departure, you can check out before the end of the reservation. The parameter `pre_checkout` must have the value `yes`. The reservation in ibelsa is shortened to the check-out day.

**Relationship between reservation and invoice:**

