Channel Connection API (Booking.com)

This is non-public API. The API methods and structure can be changed at any time which can cause problems with your connection!

To create a channel connection between Channex and an OTA you should create an entity called `channel` with mappings between Rate Plans and Room Types at Channex side and OTA side.

Because OTA's have different API and structure at their side, we can't provide a unified API to create a channel. As a result a connection to Booking.com will have some differences from a connection to Airbnb.

Methods described in this document will allow you to create a connection for Booking.com, Expedia, Google Hotel Ads and Open Channel based OTA. Connection to Airbnb will be described in another document, because it requires an oAuth connection step.

To create a channel via API you should perform several steps:

- get details about Channel Connection
- collect channel settings from User and send Test Connection request
- if channel settings is correct, produce next requests:
  - get OTA mapping details
  - get OTA connection details
- get Channex mapping details
  - choose Property
  - room types
  - rate plans
- build mapping structure
- save channel

Get Details about Channel Connection

API method to get list of channels available with information about channel arguments and mapping structure.

Endpoint:
`GET /api/v1/channels/list`
* require Bearer auth token

Response:

```
{
    "data": [
        {
            "actions": [
                "load_future_reservations"
            ],
            "channel_restrictions": {
                "currency": "EUR",
                "min_price": 500
            },
            "code": "BookingCom",
            "kind": "meta",
            "params": {
                "hotel_id": {
                    "position": 0,
```

```json
                    "title": "Hotel ID",
                    "type": "string"
                },
                "machine_account": {
                    "position": 1,
                    "title": "Machine Account ID",
                    "type": "hidden"
                }
            },
            "rate_params": {
                "occupancy": {
                    "position": 2,
                    "title": "Occupancy",
                    "type": "integer"
                },
                "pricing_type": {
                    "options": [
                        "Standard",
                        "OBP"
                    ],
                    "position": 3,
                    "title": "Pricing Type",
                    "type": "select"
                },
                "primary_occ": {
                    "position": 4,
                    "title": "Primary Occupancy",
                    "type": "boolean"
                },
                "rate_plan_code": {
                    "position": 0,
                    "title": "Rate",
                    "type": "string"
                },
                "readonly": {
                    "position": 5,
                    "title": "Read Only",
                    "type": "boolean"
                },
                "room_type_code": {
                    "position": 1,
                    "title": "Room",
                    "type": "string"
                }
            },
            "title": "Booking.com"
        }
    ]
}
```

Fields

title

String. OTA Title

params

Object with OTA params, information required to create connection.

Key of object represent field, value contain information about required value.

rate_params
Object with Mapping params.
Key of object represent field, value contain information about required value.

code
String. System code for OTA.

channel_restrictions
Object. Information about OTA limitations.
At this example you can see `min_price` equal to `500` and currency `EUR`. This mean Booking.com does not allow prices less than 5.00 EUR.

actions
List of actions which can be called for an existing channel.
To call action, you should send request to `GET /api/v1/channels/{{channel_uuid}}/execute/{{action}}`.
load_future_reservation is method to pull bookings from OTA.

kind
String. Deprecated field. Please, don’t use it.

Params details

`params` and `rate_params` values contain next structure:

position
Integer. Field position at UI.

title
String. Field name at English language.

type
String. Type of expected field value.
Possible values:

- string
- integer
- select
- boolean
- hidden

We doesn’t expect any value for `hidden` field type. It is side effect from technical limitation.

options
List. Optional. If field type if `select`, this field will contain list of available options.

Collect information about OTA connection

When you get information about available OTA, you should collect Channel Settings.
For Booking.com this will be `hotel_id`. Keep in mind, `machine_account` is not required and will be filled automatically at our side.

Test Connection

To check connection details you should perform a Test Connection request.

Endpoint:
`POST /api/v1/channels/test_connection`
* require Bearer auth token

Payload:

```
{
    "channel": "BookingCom",
    "settings": {
        "hotel_id": "5868189"
    }
}
```

Where `channel` is OTA Code from list responses, `settings` is object with collected information about connection based at `params` field from `list` response.

Possible responses:

```
200 OK

{
    "data": {
        "errors": null,
        "success": true
    }
}
```

```
200 OK

{
    "data": {
        "errors": null,
        "success": false
    }
}
```

If user provide correct details and property ready for connection at OTA side, you will receive `success: true` response.

Get OTA Mapping details

The next step is to collect mapping details from the OTA side. To do that, you should send `mapping_details` request:

Endpoint:
`POST api/v1/channels/mapping_details`
* required Bearer Token

Payload:

```
{
    "channel": "BookingCom",
    "settings": {
        "hotel_id": "5868189"
    }
}
```

Same as Mapping Details.

Response:

```
{
    "data": {
        "pricing_type": "OBP",
        "rooms": [
            {
                "id": 586818903,
                "max_children": 0,
                "rates": [
                    {
                        "id": 16385047,
                        "max_persons": 2,
                        "occupancies": [
                            1,
                            2
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "special rate"
                    },
                    {
                        "id": 16385048,
                        "max_persons": 2,
                        "occupancies": [
                            1,
                            2
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "non-refundable rate"
                    },
                    {
                        "id": 16385046,
                        "max_persons": 2,
                        "occupancies": [
                            1,
                            2
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "standard rate"
                    }
                ],
                "title": "Double Room"
            },
            {
                "id": 586818902,
                "max_children": 0,
                "rates": [
                    {
                        "id": 16385046,
                        "max_persons": 1,
                        "occupancies": [
                            1
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "standard rate"
                    },
```

```json
                    {
                        "id": 16385048,
                        "max_persons": 1,
                        "occupancies": [
                            1
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "non-refundable rate"
                    },
                    {
                        "id": 16385047,
                        "max_persons": 1,
                        "occupancies": [
                            1
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "special rate"
                    }
                ],
                "title": "Single Room"
            },
            {
                "id": 586818904,
                "max_children": 0,
                "rates": [
                    {
                        "id": 16385046,
                        "max_persons": 3,
                        "occupancies": [
                            1,
                            2,
                            3
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "standard rate"
                    },
                    {
                        "id": 16385048,
                        "max_persons": 3,
                        "occupancies": [
                            1,
                            2,
                            3
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "non-refundable rate"
                    },
                    {
                        "id": 16385047,
                        "max_persons": 3,
                        "occupancies": [
                            1,
                            2,
                            3
                        ],
                        "price_1": null,
                        "readonly": false,
                        "title": "special rate"
                    }
                ],
                "title": "Suite"
            }
        ]
    }
}
```

This is example for Booking.com. Different OTA's can provide different Mapping Details.

Booking.com returns the next information:

pricing_type
OBP or Standard
Pricing model where OBP mean Occupancy Based Model, Standard mean Per Room pricing model with Single Occupancy Rate plan.

rooms
List of Room Object. Information about Rooms available to connect at Booking.com side.

Room Object

id
Integer. Room ID

title
String. Room Title.

max_children
Integer. Count of maximum available children

rates
List of Rate Object. Information about available Rate Plans

Rate Object

id
Integer. Rate ID.

title
String. Rate Title.

max_persons
Integer. Count of maximum persons.

occupancies
List of Integers. Optional. Only for OBP Pricing Model.
List of available Occupancy Options.

price_1
Boolean. Optional. Only for Standard Pricing Model.
Flag to let us know have Rate Plan single occupancy option or not.

readonly
Boolean. Marker to represent can we manage this Rate Plan or not.
Read Only rate plans should be mapped to prevent problems with booking allocation in future.

Get OTA Connection details

Endpoint to get information about Connection details. It is temporary endpoint and this method can be merged with Mapping details in future.

Endpoint:
`POST api/v1/channels/connection_details`

- required Bearer Token

Payload:

```
{
    "channel": "BookingCom",
    "settings": {
        "hotel_id": "5868189"
    }
}
```

Same as Mapping Details.

Response:

```
{
    "data": {
        "attributes": {
            "currency": "GBP"
        },
        "type": "connection_details"
    }
}
```

For Booking.com this endpoint return information about expected Currency. Mapped Rate Plans should have same Currency.

Get Channex Mapping details

When you collect information from the OTA side, you should then collect information from Channex side.

Then first step at this process is to detect which property will be synced with the OTA. You should get one or several IDs of Properties which will be synced with the OTA. Please, keep in mind, Booking.com supports only one-to-one connections, so for Booking.com you should choose one Property ID.

Then, you should get information about Room Types for the selected property from Channex.

To do that, you can use `options` endpoint. More information can be found here - https://docs.channex.io/api-v.1-documentation/room-types-collection#room-type-options. Don’t forget to add filter by `property_id`.

And the next step is to collect information about available rate plans. To perform this operation you can use `options` endpoint (https://docs.channex.io/api-v.1-documentation/rate-plans-collection#rate-plan-options). Please, keep in mind, you should add property_id filter and enable `multi_occupancy` option to get all Rate Plans.

Example

```
https://staging.channex.io/api/v1/rate_plans/options?filter[property_id]=acb388d9-546b-42fc-9ae2-baf00e7f0d8c&mul
```

Build mapping structure

Okay, right now this is hardest part of this journey - build the mapping structure to associate Rooms and Rates between Channex and connected OTA.

To build mapping structure, you should build list of `channel_rate_plan` objects:

rate_plan_id

ID of rate plan from Channex side

settings

Object builded based at `rate_params` from `list` method.

Example for Booking.com:

```
{
  "occupancy": 1,
  "pricing_type": "OBP",
  "primary_occ": false,
  "rate_plan_code": 16385048,
  "readonly": false,
  "room_type_code": 586818903
}
```

Field description:

rate_plan_code

Integer. Rate Plan code from Booking.com side.

room_type_code

Integer. Room Type code from Booking.com side.

occupancy

Integer. Occupancy of Rate Plan at Booking.com side.

pricing_type

String. Pricing type. Just copy from Mapping Details.

primary_occ

Boolean. If Occupancy option is Primary, it will produce Restriction changes, in other case it was produce only Price changes.

readonly

Boolean. Read Only flag from Mapping details for selected Rate Plan.

Result mapping should looks like this:

```
[
    {
        "rate_plan_id": "a35f1fd4-63c6-4fbc-8fbe-359869bd9958",
        "settings": {
            "occ_changed": false,
            "occupancy": 2,
            "pricing_type": "OBP",
            "primary_occ": true,
            "rate_plan_code": 16385048,
            "readonly": false,
            "room_type_code": 586818903
        }
    },
    {
        "rate_plan_id": "2a0c416b-d8e6-4950-b52e-e7821030fd9d",
        "settings": {
            "occ_changed": false,
            "occupancy": 1,
            "pricing_type": "OBP",
            "primary_occ": false,
            "rate_plan_code": 16385048,
            "readonly": false,
            "room_type_code": 586818903
        }
    }
]
```

Save Channel

So, now we should prepare object to save and send it to API endpoint to create new channel at Channex.

Endpoint:

`POST api/v1/channels`

* require Bearer token

Payload:

```
{
    "channel": {
        "channel": "BookingCom",
        "group_id": "60674dd6-1aeb-4c41-9e0c-8ffb378a4570",
        "is_active": false,
        "title": "Opera",
        "known_mappings_list": [],
        "properties": ["acb388d9-546b-42fc-9ae2-baf00e7f0d8c"],
        "rate_plans": [
            {
                "rate_plan_id": "a35f1fd4-63c6-4fbc-8fbe-359869bd9958",
                "settings": {
                    "occ_changed": false,
                    "occupancy": 2,
                    "pricing_type": "OBP",
                    "primary_occ": true,
                    "rate_plan_code": 16385048,
                    "readonly": false,
                    "room_type_code": 586818903
                }
            },
            {
                "rate_plan_id": "2a0c416b-d8e6-4950-b52e-e7821030fd9d",
                "settings": {
                    "occ_changed": false,
                    "occupancy": 1,
                    "pricing_type": "OBP",
                    "primary_occ": false,
                    "rate_plan_code": 16385048,
                    "readonly": false,
                    "room_type_code": 586818903
                }
            }
        ],
        "settings": {
            "hotel_id": "5868189"
        }
    }
}
```

Field description

channel

String. OTA Code from `list` method.

group_id

UUID. ID of Group of connected Property.

is_active

Boolean. Marker to show should be channel will be active or not.

title

String. Channel connection title.

properties

List of UUID. List of connected property IDs.

rate_plans

List of mapping structures.

settings

Object with Channel settings based at `params`. Same as at Mapping Details request.

Response:

```
{
    "data": {
        "attributes": {
            "actions": [
                "load_future_reservations"
            ],
            "channel": "BookingCom",
            "id": "ca4ac55f-3be1-4039-9542-21e8285ffbf9",
            "is_active": false,
            "properties": [
                "acb388d9-546b-42fc-9ae2-baf00e7f0d8c"
            ],
            "rate_plans": [
                {
                    "id": "0f6fe97e-ab8b-4f0b-a1cd-dc3500f18295",
                    "rate_plan_id": "2a0c416b-d8e6-4950-b52e-e7821030fd9d",
                    "settings": {
                        "occ_changed": false,
                        "occupancy": 1,
                        "pricing_type": "OBP",
                        "primary_occ": false,
                        "rate_plan_code": 16385048,
                        "readonly": false,
                        "room_type_code": 586818903
                    }
                },
                {
                    "id": "9d7e45b3-367b-4286-a081-17a6c8d3c62e",
                    "rate_plan_id": "a35f1fd4-63c6-4fbc-8fbe-359869bd9958",
                    "settings": {
                        "occ_changed": false,
                        "occupancy": 2,
                        "pricing_type": "OBP",
                        "primary_occ": true,
                        "rate_plan_code": 16385048,
                        "readonly": false,
                        "room_type_code": 586818903
                    }
                }
            ],
            "settings": {
                "hotel_id": "5868189",
                "machine_account": "Channex-staging"
            },
            "title": "Opera"
        },
        "id": "ca4ac55f-3be1-4039-9542-21e8285ffbf9",
        "relationships": {
            "group": {
                "data": {
                    "id": "60674dd6-1aeb-4c41-9e0c-8ffb378a4570",
                    "type": "group"
                }
            },
            "known_mappings": {
                "data": []
            },
            "properties": {
                "data": [
                    {
                        "id": "acb388d9-546b-42fc-9ae2-baf00e7f0d8c",
                        "type": "property"
                    }
                ]
            }
        },
        "type": "channel"
    }
}
```

Update previously created Channels

To update channel, you can use endpoint:
`PUT /api/v1/channels/{{channel_id}}` with same payload as for Create Channel.
To remove mapping, please put it with settings equal to `null`.

Few words about testing

To play with Channel Connection API and experiment with Booking.com, you can use our 2 test properties with next IDs:

`5868189` - Occupancy Based model
`6519420` - Standard Pricing model

Please, keep in mind, at Channex.io only one Channel with same hotel_id is allowed. As result, you should remove channel after your tests or before repeat.

To remove channel you can use endpoint `DELETE api/v1/channels/{{channel_id}}`.

Postman collection for experiments

Please, use this Postman (https://www.postman.com/) collection to play around without writing any code.

Channel Conne… on.json
03 Nov 2020, 09:45 AM