

Hotelcloud WebHooks API

v1.0

Definitions

- `api_key` – A secret key assigned to a Hotel establishment on the Hotelcloud platform
 - `CRUD` – Create/Retrieve/Update/Delete operation
 - `timestamp` – ISO8601 UTC timestamp
 - `stay` – a guest's stay in a hotel
 - `user` – a 3rd party platform or application utilising the Webhooks
-

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Abstract

Webhooks are "user-defined HTTP callbacks". They are usually triggered by some event, such as new stay registration. When that event occurs, the platform makes an HTTP request to the URI configured for the webhook. Users can configure them to cause events on the Hotelcloud platform to invoke behaviour on another platform. The action taken may be anything. As webhooks use HTTP, they can be integrated into web services without adding new infrastructure.

This document describes the Hotelcloud WebHooks capability and the actions needed to be undertaken in order to successfully integrate it in a 3rd party platform or service.

- **Note:** For the generally available internet services, such as Google Calendar, Twilio, etc. it might be easier to use the [Zapier](#) Web application.
-

General requirements

1. The user SHALL store the secret key (`api_key`) of the Hotelcloud platform on a reasonably secure medium (e.g. this key SHALL be known only to the user's backend).
2. The user SHALL implement the API described below for communicating

with the Hotelcloud platform.

The actions to be undertaken by the user

1. The user SHALL set up a listening endpoint to be used for receiving Webhooks' notifications (see Webhook triggering below).
2. The user SHALL subscribe to notifications by creating a Webhook (see Creating a new Webhook below).
3. The user MAY unsubscribe from notification by either destroying the Webhook explicitly (see Destroying a Webhook below) or by returning a specific HTTP code in response to notification (see Webhook triggering below).

The API

General

1. The base URL for the API SHALL be <http://www.hotelcloud.co/api/hooks>
2. This endpoint implements part of the CRUD interface.
3. All parameters SHALL be provided in a JSON payload, e.g. in the body of the request.
4. All hooks SHALL use different callback URL.

Supported events

1. Stay:
 1. "stay_created" – when a new stay is registered on the platform.
 2. "stay_updated" – when there's a change in one of the stay's attributes is modified, for example when the guest check-outs.
2. Service request (Order):
 1. "service_request_created" – when a new order is placed.
 2. "service_request_updated" – when there's a change in one of the order's attributes, for example when the hotel or vendor started handling the order.

HTTP Headers

1. The user SHALL provide the api_key in the Authorization header in the following format: "Authorization: api_key=XXX" where XXX denotes the api_key.
2. The "Content-Type" header SHALL be "application/json"
3. The "Accept" header SHALL either be a wildcard, e.g. "*/*" or explicitly list "application/json" as one of the accepted content types.

Operations

Checking the validity of the api_key:

Request:

```
GET /api/hooks
```

Return code:

- 200 – The api_key is valid.
- 401 – The api_key is invalid.

Response body:

- Empty

Creating a new Webhook:

Request:

```
POST /api/hooks
```

Parameters:

- event (String) – the name of the events
- target_url (String) – the fully qualified URL to be notified about the event

Example:

```
{
  "event": "stay_created",
  "target_url": "http://www.example.com/notification/stay_created"
}
```

Return code:

- 201 – The hook was successfully created.
- 401 – The api_key is invalid.
- 409 – There's a hook with this notification URL already on the platform.

- 422 – There was a problem with the request, most probably one of the required parameters was invalid or blank.
-

Response body (example):

```
{"id":123}
```

Where 123 denotes the id of the hook on the Hotelcloud platform. The user SHOULD store this id for future reference.

Destroying a Webhook:

Request:

```
DELETE /api/hooks/:id
```

Where :id denotes the id of the Webhook.

Return code:

- 200 – The webhook was destroyed.
 - 401 – The api_key is invalid.
 - 404 – The webhook was not found.
-

Webhook triggering

Upon occurrence of the event to which the Webhook is subscribed, the Hotelcloud platform will place an HTTP POST request to the URL provided when subscribing (e.g. target_url).

The listener endpoint SHALL provide one of the following HTTP status codes:

- 200 – The user SHALL use this code to acknowledge receiving of the notification.
 - 410 – The user MAY use this code to communicate that it is no longer interested in receiving these notifications. Receiving of this status code by the Hotelcloud platform will destroy the Webhook, and no future notifications will be sent.
 - 50x – The user MAY use 50x errors to indicate a server side error on its side. This will get the failure logged on the Hotelcloud servers. Please note, that there won't be any retransmit attempt for those failed notifications.
-

Payload

Stay

```
{
  "id": Number
  "status": String ("registered"/"pending"/"archived")
  "check_in_at": Timestamp
  "check_out_at": Timestamp
  "last_name": String
  "room": String
}
```

Service Request

The service request MAY have extra fields not documented below.

```
{
  "id": Number
  "stay": Stay Object (see above)
  "proxy_establishment_name": The name of the establishment the
request was placed through
  "scheduled_at": Timestamp
  "delivery_charge": Number
  "tax_amount": Number
  "cost": Number (total)
  "status": String ("placed"/"accepted"/"rejected"/"on_hold")
  "request_items": [ // MAY have several objects
    {
      "id": Number
      "name": String
      "amount": Number
      "price": Number
      "cost": Number
      "tax_rate": Number or null
      "comment": String,
      "cover_image_url": relative URL
      "service_item_id": Number,
      "feature_items": Array
    }
  ],
  "visitor_attn": Boolean - needs guests's attention
  "staff_attn": Boolean - needs staff's attention
  "handler": Object,
```

```
"messages_count": Number
"created_at": Timestamp
"updated_at": Timestamp
"cc_payment_required": Boolean
}
```