Checkfront API Documentation

Release 3.0

Checkfront

Jan 04, 2018

Contents

Note: The v3.0 API documentation is a work in progress and subject to change. If you are developing an application and have additional questions, or find any errors, please contact us **for assistance.**

The **Checkfront API** allows developers to expand and build on the Checkfront Booking Platform. The API makes it easy to create web, desktop & mobile applications that directly integrate and interact with your Checkfront account.

This API is built around open standards and secure technologies to streamline development and maintain the integrity of your data.

The documentation is divided into the following sections:

CHAPTER 1

Overview

1.1 Getting Started

Note: Use of the API requires an active Checkfront subscription. Sign up for a free 21 day trial here.

1.1.1 Prerequisites

This documentation assumes you have some knowledge of web programming and API implementations – if not, please see our list of existing plug-ins and extensions, or contact us for an introduction to a qualified Checkfront developer in your area.

1.1.2 Where to Start

Depending on your familiarity with API implementations, your technical skill level, and your language/environment of choice, there may be a number of places for you to get started.

In most cases, you will likely want to start by downloading our *Sample Code* and running the included examples. You could then go through this documentation to and reference the example code to find out exactly how it works.

If you are comfortable working in your own environment or are working in a language not covered by our SDKs, you may want to begin by experimenting using *The Developer Console* to get a feel for the system, then review *Connecting to the API* to begin building your application.

1.1.3 Sample Code

Our SDKs (software development kit) handles most of the complex authentication (OAuth2) issues involved with consuming the Checkfront API. They also include sample code and additional documentation and up and running with the API in no time.

All of our SDKs are open source and are available on Github. If you have a module to contribute, please let us know.

1.1.4 Finding Help

If you're looking for help figuring out how to get your application working, or just have something you want to ask about, there are a few places you can get in touch with us:

- Contact us directly through the support link in your Checkfront account.
- Talk with other developers on our forum.
- Fork us (and contribute!) on GitHub.com/Checkfront.

1.1.5 Rate Limits

API throttle limit: We reserve the right to tune the limitations, but they are always set high enough to allow a well-behaving interactive program to do its job.

When the rate limit is exceeded Checkfront will send an HTTP 503 status code. The number of seconds until the throttle is lifted is sent via the "Retry-After" HTTP header, as specified in RFC 2616.

1.1.6 Terms of Service

Use of this API is strictly bound by the terms as specified in Checkfront API Terms of Service.

Some functionality documented here may not be available to you based on your plan, or access level of your account.

1.2 Connecting to the API

The **Checkfront API** is accessable via a secure authenticated HTTPS connection to our hosted services, and is isolated to your subscription. This requires your application to have the ability to connect to external servers using SSL and one of the identity token services provided to your Checkfront account.

1.2.1 Setting Up Your Application

To start setting up your application, open your Checkfront account page and use the menu to navigate your browser to **Manage > Developer**. This page will provide access to a listing of your active application clients, as well as containing your webhook notifications configuration under the "**Webhooks**" tab, and *The Developer Console*.

Click the "**New Application**" button in the upper-left corner, and *carefully* read the terms of service provided.

Choosing Your Workflow

When adding your application, you will be given a choice between two "authentication types". The method you choose for your application is largely determined by the intended use and scope of your application, and in certain cases you may choose to add more than one access method to your account.

Token Authentication

Token authentication makes use of a simple static private key and secret that can be used for *server-to-server communication only*. For example, this can be used by an application running on your **private web server** to create a customer-facing booking page. Applications using this method can only be used privately within your organization, and *must not be distributed*.

OAuth2 Authentication

OAuth2 is a secure open standard providing a simple transparent avenue of authenticating with the API. Using this method of authentication can allow your application to act on behalf of individual members of your staff by allowing them to "log in" and grant permission to the app. This is ideal for any case where you want to allow your staff to (for example) make or change bookings in the system.

Please see our SDKs or support libraries for OAuth2 in your preferred environment. If we don't have an example for your language, there are many libraries and documentation available for various programming languages that provide all the functionality you'll need to implement a custom OAuth2 solution.

Obtaining API Credentials

After you choose an application name (for your own reference), an authentication type, and accept the terms of service, you can click "**Create**" to save and return to your application listing.

Click on your new application to show the credentials used to connect your application to the API.

> **Warning:** Your API credentials provide your application with access to private data stored on your account and the ability to act on your behalf. Treat these as you would your password, and be careful not to distribute or send these to untrusted parties.

If you are using the SDK, see Setting Up The SDK

Note: Much of the following documentation is intended for custom integrations and understanding of the protocols involved in authentication. Our SDK resources abstract the complex authentication processes to simplify development of your application.

1.2.2 Authenticating with Token Pairs

If you are using token authentication, your application will provide you with two keys:

API Key	Will be set as your *HTTP BASIC* **Username**
API Secret	Will be set as your *HTTP BASIC* **Password**

When sent together in a request header using HTTP Basic authorization, these allow direct access to API endpoints without any additional preamble.

See `CURLOPT_USERPWD` if working with cURL libraries, or the documentation on *HTTP Basic authentication* relevant to your chosen framework. If building your requests manually, HTTP Basic credentials are **base64 encoded** in the sequence "`username:password`" and sent in the **request header** in the following format:

```
GET /booking/1 HTTP/1.1
Authorization: Basic␣
↪M2JlOTg2NDFmMDc0NWI2ZmU3ZGFjYzJkZjk0N2FkYmMxZGE3MzEyZDo0YzRkNTk4YTVkOTQwZjA4ZmRiNDM1YjY5YWY5ODZjNzI
```

1.2.3 Authenticating with OAuth2

Logging in with a Staff Account

1. Redirect your staff user to your **Authorize Token URL** (eg. `https://your-company.checkfront.com/oauth/`) with the following query string parameters set:

client_id	Your application's "Consumer Key" (see your application setup).
response_type	**MUST** be set to `code`
redirect_uri	The URI to return your user to after authorization.

2. Upon completion of authorization, the staff user will be sent to the redirect_uri specified, passing a **code** parameter in the query string that must be exchanged by your application for an access/refresh token pair within 60 seconds.

   To exchange the code for an access token, make a call to your **Access Token URL** (eg. `https://your-company.checkfront.com/oauth/token/`) with the following parameters set:

client_id	*Should be sent as HTTP Basic credentials.* See *Consumer Key / Consumer Secret*.
client_secret	*Should be sent as HTTP Basic credentials.* See *Consumer Key / Consumer Secret*.
grant_type	**MUST** be set to `authorization_code`
code	The authorization code as returned in the client's GET request to your page.
redirect_uri	The URI to return your user to after authorization.

3. Store the token returned by the previous call in a *secure* database along with a field containing the timestamp of most recent update to the token. Your tokens should be refreshed on a regular basis as long as the authorization continues to be used.

   Your application should store and make use of the following fields from the response:

access_token	*string*	See *Access Token*.
expires_in	*integer*	The time (in seconds) after which the *access* token will expire.
refresh_token	*string*	See *Refresh Token*.

Using and Maintaining OAuth2 Tokens

While your first authentication will provide a usable access token for identifying with the server, the access token has a fixed lifetime and must be refreshed in order to maintain access to the API.

See *Refresh Token* and other details below for information on performing a refresh.

OAuth2 Reference

Authorization Endpoints

There are two important endpoints used in authenticating tokens using OAuth2, which are both displayed when viewing the application key setup on your Checkfront developer page.

- Your **Authorize Token URL** is used when redirecting a user to grant permission to use their account. On success, this will return a code for you to pass to the *Access Token URL* to grant a token you can use to access the API.

```
https://your-company.checkfront.com/oauth/
```

- Your **Access Token URL** is used for granting access tokens from code requests, and refreshing existing access/refresh tokens.

```
https://your-company.checkfront.com/oauth/token/
```

Consumer Key / Consumer Secret

These are generated when setting up your application and can be found on your Checkfront developer page. Your consumer key and secret allow your application to grant and refresh tokens on behalf of your users.

Warning: You consumer key and secret should **only** be sent together when making calls to your **Access Token URL**. When your application is making calls to endpoints requiring a valid access token, the key/secret pair **should not** be sent.

As with token pairs (see above), these can (and should) be sent as HTTP Basic credentials. *However*, these can only be sent in this manner to the **/oauth/token/** (code/refresh) endpoint. Your request will be *rejected* if you attempt to send these to an /api/ endpoint.

See `CURLOPT_USERPWD` if working with cURL, or the documentation on HTTP Basic authorization relevant to your chosen framework. If building your requests manually, HTTP Basic credentials are **base64 encoded** in the sequence "`username:password`" and sent in the **request header** in the following format:

```
Authorization: Basic␣
↪M2JlOTg2NDFmMDc0NWI2ZmU3ZGFjYzJkZjk0N2FkYmMxZGE3MzEyZDo0YzRkNTk4YTVkOTQwZjA4ZmRiNDM1YjY5YWY5ODZjNzI
```

Access Token

This is used by the API server to identify you and allow the application to act on your behalf. When using OAuth2 for your application, an access token is **required** to be sent with **all** API calls to secure data endpoints.

Access tokens have a lifetime of **14000 seconds** (this will be returned as `expires_in` when new tokens are granted), after which they must be *refreshed* to obtain a new token. Your application should keep track of when this token will be expiring and check if it needs refreshing before attempting a request.

When sending your token with an API request, it can be sent in a header in the following format:

```
Authorization: BEARER f58ef579d0bb5ffb3b5bb0985a85e21a
```

It will also be accepted in the form of the query string parameter `access_token` in the GET request or POST body if necessary for your application, although this is not recommended in a live application.

```
access_token=f58ef579d0bb5ffb3b5bb0985a85e21a
```

Refresh Token

After your current **access token** has expired, this token can be passed to create a new access/refresh token pair, which must completely replace your previously stored token (which will be invalidated).

To exchange the the refresh token for a new access/refresh token pair, make a call to your **Access Token URL** (eg. `https://your-company.checkfront.com/oauth/token/`) with the following parameters set:

client_id	*Should be sent as HTTP Basic credentials.* See *Consumer Key / Consumer Secret*.
client_secret	*Should be sent as HTTP Basic credentials.* See *Consumer Key / Consumer Secret*.
grant_type	**MUST** be set to `refresh_token`
refresh_token	The current (active) refresh token for this user.

Refresh tokens have a lifetime of **14 days** from issue, after which (if allowed to expire) you must generate a new access/refresh token pair to regain application authorization.

1.3 Request Formatting

All requests made to the server are sent over an encrypted SSL connection using standard HTTPS methods such as **GET** and **POST**.

When submitting input parameters with your request, these are encoded either in the query string for GET requests, or in the body of a POST request using commonplace url encoding of key-value pairs (*`application/x-www-form-urlencoded`*).

The headers of your request should also include the IP of the client you're forwarding for, as well as the credentials used to authenticate with the API. If you have an open session on the server, you can also pass in a **session_id** cooking containing the ID to have it automatically reopened.

X-Forwarded-For	The IP address of the connecting client.
X-On-Behalf	The staff account to act on behalf of. Use "off" to act as a customer.
Authorization	Credentials used to query the API. See *Authenticating with Token Pairs* or *OAuth2 Reference*.

For example, to query a list of bookings with a status of PAID starting today, your full raw request might look like the following:

`GET /api/3.0/*`

Example request:

```
GET /api/3.0/booking/index?start_date=today&status_id=PAID HTTP/1.1
Host: your-company.checkfront.com
Accept: application/json
X-Forwarded-For: 66.228.55.142
X-On-Behalf: 1
Authorization: Basic␣
↪M2JlOTg2NDFmMDc0NWI2ZmU3ZGFjYzJkZjk0N2FkYmMxZGE3MzEyZDo0YzRkNTk4YTVkOTQwZjA4ZmRiNDM1YjY5YWY5OD
```

Request Headers

- Authorization – Credentials used to query the API. See *Authenticating with Token Pairs* or *OAuth2 Reference*.
- *X-On-Behalf* – The staff account to act on behalf of. Use "off" to act as a customer.
- *X-Forwarded-For* – The IP address of the connecting client/customer.

1.4 Response Formatting

All responses from the API are formatted as **JSON** (JavaScript Object Notation) objects containing information related to the request, and any status. Every modern language should have libraries capable of quickly parsing JSON objects. See json.org for more information.

See also:

For JSON decoding libraries in common environments, see:

- PHP: json_decode
- Python: json.JSONDecoder
- Ruby: JSON.parse
- .Net: Json.NET
- JavaScript: JSON3.parse / jQuery.getJSON

All responses from the API will include some information related to the processing of the request, such as the following:

Field	Type	Description
version	*string*	Current API version used to process the request.
account_id	*integer*	The staff account that the request was made on behalf of.
host_id	*string*	Checkfront host account to which the request was made.
name	*string*	Display name of the company the account belongs to.
locale	*array*	*Contains the following fields:*
locale -> **id**	*string*	Selected locale for the account (eg. **en_US**)
locale -> **lang**	*string*	Selected language for the account (eg. **en**)
locale -> **currency**	*string*	Selected currency for the account (eg. **CAD**)
request	*array*	*Contains the following fields:*
request -> **status**	*string*	Status of the request. WIll be "**OK**" on success.
request -> **resource**	*string*	Resource accessed by the request.
request -> **records**	*integer*	Number of records returned by the request.
request -> **limit**	*integer*	The limit on number of records returned (where applicable).
request -> **page**	*integer*	The current page returned for multi-page requests.
request -> **pages**	*integer*	Number of pages of response records available.
request -> **time**	*float*	The time taken (in ms) to generate the response.
request -> **method**	*string*	HTTP request method used in the API request.

If any **critical errors** (e.g. failed credentials) occur, the API will not return any of these fields, but will instead return an object containing any available error information.

CHAPTER 2

Developer Guide

2.1 The Developer Console

The developer console provides easy, direct access to the API to aid in development or debugging your API calls. This allows you to interact with the API without setting up an environment, where you can manually perform any of the actions available to the API.

Note that all parameters for both GET and POST requests should be URL encoded in the query string.

The console can be found via the "Console" link under **Manage > Developer**.

Note: All calls made in the console are **live** requests on your account, not examples. Any changes you make to bookings through the console will persist outside of the console.

2.2 Setting Up The SDK

Once you have set up credentials for your application (see *Connecting to the API*), you can set up the SDK and its examples to get a head start on working with the API.

Warning: Your API credentials provide your application with access to private data stored on your account and the ability to act on your behalf. Treat these as you would your password, and be careful not to distribute or send these to untrusted parties.

2.2.1 With Token Authentication

Using the PHP-SDK with your extended Checkfront class, you would connect to the API by passing your token key and secrets into the settings, similar to the following:

```
$Checkfront = new CheckfrontAPI(
        array(
                'host' => 'your-company.checkfront.com',
                'account_id' => 'off', // act on behalf of the customer
                'auth_type' => 'token',
                'api_key' => '3be98641f0745b6fe7dacc2df947adbc1da7312d',
                'api_secret' =>
↪'b300e27b07bfc6fed944a0116b595ec2130e9efc6410424d12c68c7766f2d861',
        )
);
```

Note that the **api_key** and **api_secret** fields must be set to the values obtained when setting up your application in your Checkfront account.

Once this is done, the application will be ready to run calls against the API.

2.2.2 With OAuth2 Authentication

If you are not familiar with how OAuth2 works, you may want to review *OAuth2 Reference* and the preceding authentication overview.

When using OAuth2 authentication, your application will require a place to store access/refresh tokens for your users. While the examples do provide a simple implementation of this, we highly recommend writing your own secure storage procedure suited to the application you're building. These are typically best stored in a database (such as one you may already be using for other requirements of your application), as they are replaced on a regular basis as long as the application continues to be used.

Setup

As with token authentication, a few fields are passed into the object on creation to configure your connection to the API:

```
$Checkfront = new CheckfrontAPI(
        array(
                'host' => 'your-company.checkfront.com',
                'auth_type' => 'oauth2',
                'consumer_key' => '1d2d5cad60174b5972243693d082e4b4e54979bf',
                'consumer_secret' =>
↪'8e3751291c3d0fe090a7d5b18d964407baff96c1028e1e1afb1014d1db85b25a',
                'redirect_uri' => 'oob'
        )
);
```

Your **consumer_key** and **consumer_secret** should match the values provided when setting up your application. The **redirect_uri** should match the "Callback URL" value you define on the same page.

Staff Logins

The SDK includes a simple abstraction process to login staff using OAuth2, easing much of the more complex process of handling logins.

To review the process of handling staff credentials, your application would:

Handle previous logins

Load the previous **access_token** and **refresh_token** into your CheckfrontAPI object, likely from a database or a cookie.

To do this, you will at the very least want to write a custom CheckfrontAPI->store() function to save & load tokens. When handling multiple users with tokens stored in your database, remember to account for storing tokens for each.

Whenever a token is prepared for storage by the SDK, it will send an *array* to your **CheckfrontAPI->store($data)** function, containing the following fields for storage:

refresh_token	See *Refresh Token*.
access_token	See *Access Token*.
expire_token	The unix timestamp after which the access token will expire.
updated	The current time.

When **loading** tokens on initialization, the SDK will call the same *store()* function **without any parameters**. In this case, the function should **return** the previously stored values as an array.

If an **access_token** is expired, the SDK will automatically refresh it on startup. However, if a **refresh_token** expires, you will need to invalidate the stored tokens and begin the login process for the user. Based on store() input above, you can generally assume the expiry is **updated** + 14 days, but the tokens can be refreshed at any time before that point.

New logins

1. Make a call to **CheckfrontAPI->authorize(boolean $redirect)** to either return the login URL or set a redirect header to it. From here, the staff member grants permission to your application, and is redirected back to your specified callback URL.
2. On returning to your callback URL, the client will pass an authentication **code** in the query string (that's **$_GET['code']**), which your application then passes into the fetch_token function of the CheckfrontAPI object as follows:

```
$Checkfront->fetch_token($_GET['code']);
```

3. On completion, a new authenticated token will be returned, and as above, the token will be sent to your **store()** function.
4. You're ready to make calls to the API on behalf of the staff member.

2.2.3 Making Calls to the API

The SDK has simple abstractions to common HTTP request methods used with the API – in particular, **CheckfrontAPI->get($path, $data);** and **CheckfrontAPI->post($path, $data);**

Using these functions, your application would send the base path of the API element to be accessed in **$path**, and any options to be sent as an associative array in **$data**.

For example, instead of manually building a query string in a request such as:

```
GET /api/3.0/booking/index?start_date=today&status_id=PAID HTTP/1.1
Host: your-company.checkfront.com
```

You could instead use:

```php
$data = array(
        'start_date' => 'today',
        'status_id' => 'PAID'
);

$Checkfront->get('booking/index', $data);
```

2.3 How To...

2.3.1 List Items

You can query a list of available items based on search critera supplied in an API call to *item*.

When no dates are passed in the API call, a full list of enabled items in the inventory. When a date is passed, the API will return a "**rated**" response that includes pricing and availablity, as well as a "**SLIP**" that will be used to add the item to a session.

In order to build bookings, you'll need to first retrieve "**rated**" item details (by declaring the booking length and any additional parameters in an *item* call, see above), which will return a "**SLIP**" – effectively a token encoding the necessary information to book that item in that time period with the options you've selected.

Example rated request to a specific *item*:

```
GET /api/3.0/item/17?start_date=20141224&end_date=20150101&param[guests]=2 HTTP/1.1
Host: your-company.checkfront.com
```

2.3.2 Add Items to a Booking Session

Once you have the *SLIP* for your desired item (or items), this can be passed in to create a new *booking/session*. The session can be thought of as your "cart", and can be modified until you are ready to submit your booking.

If you have collected multiple items to be added, you can also add them in a batch by sending them as an array.

Example of adding a *SLIP* to a *new* session:

```
POST /api/3.0/booking/session HTTP/1.1
Host: your-company.checkfront.com

slip[]=17.20141224X8-guests.2&slip[]=18.20141224X8-guests.2
```

Note: For parent/child grouped items, the parent item functions as a container for the child items. As such, the request to obtain the SLIP must be made directly to the child item, rather than the parent.

Once the SLIP has been obtained, it can be added to the booking session as outlined above.

2.3.3 Alter Items in the Session

If you need to alter items in a *booking/session*, such as opting-in to upsell items, changing quantities, or removing the item, you can use the 'alter' query parameter of a *booking/session* call to an existing session.

To change the selected quantity of an added session item, pass the quantity in to the **alter** array entry for the selected line (which will be part of each session response). To opt-in or out of package items, find their line id and specify "optin" or "optout" instead of a quantity.

To remove an item from the session, use the alter array to specify that the line item should be removed from the order.

```
POST /api/3.0/booking/session HTTP/1.1
Host: your-company.checkfront.com

session_id=rtdv4osethqurlmqgi55mcrkm4&alter[3]=4&alter[2.1]=optin&alter[1]=remove
```

2.3.4 Create a New Booking

After you've added *SLIP*s to your session, your application should then capture the customer information needed to make a booking. The fields required for checkout on your account can be retrieved with a GET request to *booking/form*.

To submit a booking to the system, you'll then pass your `session_id` along with the required customer information to a *booking/create* call, which will return data relating to your booking, such as the booking/customer IDs (which could be recorded in your system) and an invoice/payment URL (if applicable).

2.4 Frequently Asked Questions

2.4.1 Understanding Errors

"The access token was not found."

When using OAuth2, this error will only appear if the server **did not receive** the *Authenticating with Token Pairs* from your application.

"The access token provided is invalid."

The server received an access token from you, but could not authorize it.

"Malformed auth header."

Your token pair authorization did not contain all of the required token information in the request header.

"You do not have access to this resource."

The token pair you attempted to use could not be authorized to use the API. Double check that you are using the correct credentials, and that you are using the correct authentication type. See *Connecting to the API*

CHAPTER 3

Reference

3.1 Webhooks

Your Checkfront account is capable of directly POSTing details of new or modified bookings, items, item events, and transactions to your SSL-secured server endpoint.

Notifications are sent over a standard **HTTPS POST** request as **JSON**, **XML**, or **x-www-form-urlencoded** data.

Your server **must** respond with an **HTTP 200** class status code immediately upon receipt. If your server does not indicate a successful response in a timely manner, after 5 consecutive failed delivery attempts, your webhook will be automatically disabled.

Webhook setup can be found in the **Manage > Developer** section of your Checkfront account under the "**Webhooks**" tab.

3.1.1 Booking Modified & Booking Status Change

The **Booking Modified** webhook event is triggered whenever a booking is created or modified, and the **Booking Status Change** webhook event is triggered whenever a booking status is altered. These events may occur simultaneously in some circumstances, and both are triggered whenever a booking status is changed. Using both of these events on the same webhook endpoint URL is discouraged.

A **JSON**, **XML**, or **x-www-form-urlencoded** object containing the following export fields can be found in directly in the **raw body** of the POST request to your server:

booking

Field	Type	Description
status	*string*	The current status code of the booking
code	*string*	A unique booking code used to refer to the booking
created_date	*timestamp*	The date on which the booking was created
staff_id	*integer*	Account ID of the staff account used to create the booking
source_ip	*string*	The IP address used to create the booking
start_date	*timestamp*	The start date/time of the booking, based on order items
end_date	*timestamp*	The end date/time of the booking, based on order items
customer	*array*	Customer details attached to the booking (See *booking.customer* below)
meta	*array*	A set of fields containing your custom parameters and other info
order	*array*	Details on booking items and payment. See *booking.order* below
tid	*string*	Details on the Tracking ID used for referral & campaign tracking

booking.customer

Field	Type	Description
name	*string*	The customer's full name
email	*string*	The customer's email address
address	*string*	The customer's street address
region	*string*	The customer's province or state
country	*string*	The customer's country of residence
postal_zip	*string*	The customer's postal or zip code
phone	*string*	The customer's phone number

booking.order

Field	Type	Description
sub_total	*float*	The sub-total of all charges added to the order
paid_total	*float*	The total amount the customer has paid on the order
total	*float*	The total of all charges and taxes added to the order
tax_total	*float*	The sum of all taxes applied to the order
taxes	*array*	Individual taxes that have been applied to the order, their names, and amounts
discount	*float*	The amount that has been discounted from the order total (if applicable)
items	*array*	Details on items included in the order (See *booking.order.items.item* below)

booking.order.items.item

An entry for *each* item in the booking exists will contain following fields:

Field	Type	Description
start_date	*timestamp*	The start date of the booking item
end_date	*timestamp*	The end date of the booking item
sku	*string*	The unique stock keeping unit of the item
slip	*string*	The booking slip attached to the item
package	*integer*	The package the item belongs to (if applicable)
status	*string*	The booking status of the item
total	*float*	The total price of the booking item
tax_total	*float*	The sum of taxes applied to this item
qty	*integer*	The quantity of this item in the booking

Sample Booking Notification

JSON


```
{
    "@attributes": {
        "version": "3.0",
        "host": "your-company.checkfront.com"
    },
    "booking": {
        "@attributes": {
            "booking_id": "157"
        },
        "status": "HOLD",
        "code": "KMVQ-060314",
        "created_date": "1394156618",
        "staff_id": "1",
        "source_ip": "66.228.55.142",
        "start_date": "1394128800",
        "end_date": "1394128800",
        "customer": {
            "name": "Your Customer",
            "email": "support@checkfront.com",
            "region": "BC",
            "address": "170-422 Richards Street",
            "city": "Vancouver",
            "country": "CA",
            "phone": "1 (800) 559-0985",
            "postal_zip": "V6B 2Z4"
        },
        "meta": {
            "note": "Free icecream!",
            "mobile_device": {}
        },
        "order": {
            "@attributes": {
                "currency_id": "CAD"
            },
            "sub_total": "46.00",
            "tax_total": "2.30",
            "paid_total": "0.00",
            "total": "48.30",
            "taxes": {
```

```
                "tax": {
                    "@attributes": {
                        "tax_id": "6"
                    },
                    "name": "GST",
                    "amount": "2.30"
                }
            },
            "items": {
                "item": [
                    {
                        "@attributes": {
                            "line_id": "1",
                            "item_id": "6"
                        },
                        "start_date": "1394128800",
                        "end_date": "1394128800",
                        "sku": "paddle",
                        "slip": {},
                        "package_id": "1",
                        "status": "HOLD",
                        "total": "6.30",
                        "tax_total": "0.30",
                        "qty": "2"
                    },
                    {
                        "@attributes": {
                            "line_id": "2",
                            "item_id": "8"
                        },
                        "start_date": "1394092800",
                        "end_date": "1394092800",
                        "sku": "paddleboard",
                        "slip": {},
                        "package_id": "1",
                        "status": "HOLD",
                        "total": "42.00",
                        "tax_total": "2.00",
                        "qty": "2"
                    }
                ]
            }
        }
    }
}
```

XML

```
<?xml version="1.0" ?>
<checkfront-notify version="3.0" host="your-company.checkfront.com">
    <booking booking_id="157">
        <status>HOLD</status>
        <code>KMVQ-060314</code>
        <created_date>1394156618</created_date>
        <staff_id>1</staff_id>
        <source_ip>66.228.55.142</source_ip>
```

```
        <start_date>1394128800</start_date>
        <end_date>1394128800</end_date>
        <customer>
            <name>Your Customer</name>
            <email>support@checkfront.com</email>
            <region>BC</region>
            <address>170-422 Richards Street</address>
            <city>Vancouver</city>
            <country>CA</country>
            <phone>1 (800) 559-0985</phone>
            <postal_zip>V6B 2Z4</postal_zip>
        </customer>
        <meta>
            <note>Free icecream!</note>
            <mobile_device></mobile_device>
        </meta>
        <order currency_id="EUR">
            <sub_total>46.00</sub_total>
            <tax_total>2.30</tax_total>
            <paid_total>0.00</paid_total>
            <total>48.30</total>
            <taxes>
                <tax tax_id="6">
                    <name>GST</name>
                    <amount>2.30</amount>
                </tax>
            </taxes>
            <discount>0.00</discount>
            <items>
                <item line_id="1" item_id="6">
                    <start_date>1394128800</start_date>
                    <end_date>1394128800</end_date>
                    <sku>paddle</sku>
                    <slip/>
                    <package_id>1</package_id>
                    <status>HOLD</status>
                    <total>6.30</total>
                    <tax_total>0.30</tax_total>
                    <qty>2</qty>
                </item>
                <item line_id="2" item_id="8">
                    <start_date>1394092800</start_date>
                    <end_date>1394092800</end_date>
                    <sku>paddleboard</sku>
                    <slip/>
                    <package_id>1</package_id>
                    <status>HOLD</status>
                    <total>42.00</total>
                    <tax_total>2.00</tax_total>
                    <qty>2</qty>
                </item>
            </items>
        </order>
    </booking>
</checkfront-notify>
```

3.1.2 Item Calendar Edit

3.1.3 Item Event/Discount Edit

This webhook is triggered whenever any item events or discounts are altered on your account. Because of their similarities, **Item Events and Discounts are both called "rates"**. When fields have specific meanings for either Item Events or Discounts, those differences are defined in the Description column.

A **JSON**, **XML**, or **x-www-form-urlencoded** object containing the following export fields can be found in directly in the **raw body** of the POST request to your server:

rate

Field	Type	Description
rate_id	*integer*	A unique code used to refer to this rate
name	*string*	The name of this rate
type	*string*	Item Events: "SP" for "Special" events, "SE" for "Seasonal" events. Discounts: "DC" for "Discount"
start_date	*integer*	The date *(yyyymmdd)* when this rate starts to apply
end_date	*integer*	The date *(yyyymmdd)* when this rate stops applying (0 if the rate has no end date)
span	*integer*	The "Force item length to the above start and end dates" option (1 if selected, 0 if not selected)
price_src	*string*	If applicable, the type of price modification for this rate ("Base Price" is "B", "Create new Price Point" is "N", "Dynamic" is "D", "Yield" is "Yield")
dynamic_rate	*decimal*	If this event is a "Dynamic" price rate (*price_src* is "D"), this value holds the rate's percent change or fixed amount price.
dynamic_type	*string*	Dynamic Price rates: "P" for percent, "F" for fixed amount
status	*string*	Item Events: whether this is an available ("A") or unavailable ("U") event. Discounts: always available ("A")
repeat_id	*string*	If the rate is not recurring, this value is blank. Weekly recurring events are "W", "Always" recurring events are "*".
enabled	*integer*	If the rate is enabled, 1, otherwise 0 when disabled.
rule_set_id	*integer*	The ruleset id applying to this event. The default ruleset id is 1.
vouchers	*integer*	Discounts: the number of vouchers attached to this discount.
details	*object*	See *rate.details* below.
repeat	*object*	See *rate.repeat* below.
apply_to	*object*	See *rate.apply_to* below.

rate.details

Field	Type	Description
thresholds	*object*	If this is rate has a "Yield" **price_src**, this element will contain a list of objects containing a level and rate pair for each threshold. See *rate.details.thresholds* below

rate.details.thresholds

Field	Type	Description
threshold	*object*	This contains a pair of level and rate values for a threshold. See *rate.details.thresholds.threshold* below

rate.details.thresholds.threshold

Field	Type	Description
level	*integer*	The inventory threshold at which this Yield priced rate applies
rate	*decimal*	The percentage price multiplier for this threshold

rate.repeat

Field	Type	Description
days	*object*	If set, this rate repeats on specific days of the week. See *rate.repeat.days* below.

rate.repeat.days

Field	Type	Description
<day of week abbreviation> (e.g., mon, tue,...)	*integer*	The rate repeats for this day of the week.

rate.apply_to

Field	Type	Description
items / categories	*object*	This object contains a list of the items or categories which this rate applies to. See *rate.apply_to.items* below.

rate.apply_to.items

Field	Type	Description
id	*integer*	An item_id which this rate applies to

rate.apply_to.categories

Field	Type	Description
id	*integer*	This contains a category_id which this rate has completely selected. **This rate will apply to any new items that are added to this category.**

Sample Rate Notification

JSON

```
{
    "version": "3.11.0.0",
    "host": "test.checkfront.com",
    "type": "rate",
    "action": "update",
    "rate": {
        "rate_id": 8,
        "name": "Max 3 Days",
        "start_date": 20161221,
        "end_date": 0,
        "span": 0,
        "price_src": "Yield",
        "dynamic_rate": 10,
        "status": "A",
        "type": "SP",
        "repeat_id": "W",
        "enabled": 1,
        "details": {
            "thresholds": [
                {
                    "threshold": {
                        "level": 5,
                        "rate": 150.5
                    }
                },
                {
                    "threshold": {
                        "level": 2,
                        "rate": 200
                    }
                }
            ]
        },
        "repeat": {
            "days": {
                "tue": 1,
                "sat": 1
            }
        },
        "rule_set_id": 1,
        "dynamic_type": "P",
        "apply_to": {
            "items": [
                {
                    "id": 14
```

```
                    },
                    {
                        "id": 16
                    },
                    {
                        "id": 35
                    },
                    {
                        "id": 36
                    },
                    {
                        "id": 37
                    }
                ],
                "categories": [
                    {
                        "id": 3
                    },
                    {
                        "id": 5
                    }
                ]
            }
        }
}
```

XML

```
<?xml version="1.0" encoding="utf-8"?>
<checkfront-notify version="3.11.0.0" host="test.checkfront.com">
    <type>rate</type>
    <action>update</action>
    <rate>
        <rate_id>8</rate_id>
        <name>Max 3 Days</name>
        <start_date>20161221</start_date>
        <end_date>0</end_date>
        <span>0</span>
        <price_src>Yield</price_src>
        <dynamic_rate>10</dynamic_rate>
        <status>A</status>
        <type>SP</type>
        <repeat_id>W</repeat_id>
        <enabled>1</enabled>
        <details>
            <thresholds>
                <threshold>
                    <level>5</level>
                    <rate>150.5</rate>
                </threshold>
                <threshold>
                    <level>2</level>
                    <rate>200</rate>
                </threshold>
            </thresholds>
        </details>
```

```
        <repeat>
            <days>
                <tue>1</tue>
                <sat>1</sat>
            </days>
        </repeat>
        <rule_set_id>1</rule_set_id>
        <dynamic_type>P</dynamic_type>
        <apply_to>
            <items>
                <id>14</id>
                <id>16</id>
                <id>35</id>
                <id>36</id>
                <id>37</id>
            </items>
            <categories>
                <id>3</id>
                <id>5</id>
            </categories>
        </apply_to>
    </rate>
</checkfront-notify>
```

3.1.4 Item Update

3.1.5 New Payment/Refund

This webhook is triggered whenever any transactions are created or refunded on your account.

A **JSON**, **XML**, or **x-www-form-urlencoded** object containing the following export fields can be found in directly in the **raw body** of the POST request to your server:

payment

Field	Type	Description
transaction_id	*string*	A unique code used to refer to this transaction
date	*timestamp*	When this transaction was created
status	*string*	Identifies whether the transaction was a payment (PAID) or a refund (REFUND)
amount	*decimal*	The transaction's monetary amount
gateway_id	*string*	Identifies which payment gateway made the transaction
payment_mask	*integer*	The last four digits of the card, or (POS) for a POS transaction
payment_type	*string*	The type of card used, or the type of POS transaction
payment_customer	*string*	The payer's name
booking_id	*string*	The code of the booking which received this transaction (See *booking*)

Sample Payment Notification

JSON

```
{
    "version": "3.11.0.0",
    "host": "your-company.checkfront.com",
    "type": "transaction",
    "action": "insert",
    "transaction": {
        "transaction_id": "TEST1482283550",
        "date": 1582283550,
        "status": "PAID",
        "amount": "27.00",
        "gateway_id": "TestPayment",
        "payment_mask": "1111",
        "payment_type": "Visa",
        "payment_customer": "Testy McTestface",
        "booking_id": "LULZ-080916"
    }
}
```

XML

```
<?xml version="1.0" encoding="utf-8"?>
<checkfront-notify version="3.11.0.0" host="your-website.checkfront.com">
    <type>transaction</type>
    <action>insert</action>
    <transaction>
        <transaction_id>TEST1482284175</transaction_id>
        <date>1482284175</date>
        <status>PAID</status>
        <amount>27.00</amount>
        <gateway_id>TestPayment</gateway_id>
        <payment_mask>1111</payment_mask>
        <payment_type>Visa</payment_type>
        <payment_customer>Testy McTestface</payment_customer>
        <booking_id>LULZ-080916</booking_id>
    </transaction>
</checkfront-notify>
```

3.2 API 3.0 Endpoints

3.2.1 account

GET /api/3.0/account
Retrieve a list detailing all staff accounts in your system.

```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
```

```
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "status": "OK",
        "resource": "account",
        "records": 1,
        "limit": 0,
        "page": 1,
        "pages": 1,
        "time": 0.0013,
        "method": "get"
    },
    "account": [
        {
            "account_id": 1,
            "first_name": "Johnson",
            "last_name": "Smith",
            "email": "support@checkfront.com",
            "mobile_phone": "",
            "nickname": "John",
            "login_id": "johnsmith",
            "enabled": 1,
            "date_created": 1490573063,
            "date_last_login": 1490573064,
            "ip_last_login": "0.0.0.0",
            "feed_token":
↪"354629ce3117f34e0d7208d4847137f371deae824625ee1635009b7b88ed8804",
            "partner": 0
        }
    ]
}
```


3.2.2 booking

A **booking** in the system represents an order and all associated information. It's possible to list, retrieve, create, and modify bookings in the system through use of the API.

booking

GET /api/3.0/booking

Retrieve a listing of bookings in the system.

Query timestamps may be Date strings or unix timestamps, and can be prefixed with '<' or '>' to match before or after a date.

Query Parameters

- **status_id** (*string*) – The current status of a booking.
- **customer_id** (*integer*) – The customer id associated with the booking.
- **customer_email** (*string*) – The customer email associated with the booking.

- **start_date** (*string/timestamp*) – The date the booking starts on (i.e. check-in).
- **end_date** (*string/timestamp*) – The date the booking ends on (i.e. check-out).
- **last_modified** (*string/timestamp*) – The date the booking was last changed. Useful for getting bookings added or changed since your last call.
- **limit** (*integer*) – The limit of bookings to return per page (default: 100).
- **page** (*integer*) – The page of results to return.

Response JSON Array of Objects

- **booking_id** (*integer*) – The internal index of a booking.
- **code** (*string*) – The unique booking reference code.
- **status_id** (*string*) – The status code of the booking.
- **status_name** (*string*) – The display name of the booking status.
- **created_date** (*timestamp*) – The date/time the booking was created.
- **total** (*float*) – The full value of the booking.
- **tax_total** (*float*) – The sum of the taxes applied to the booking.
- **paid_total** (*float*) – The amount the customer has paid on the booking.
- **customer_name** (*string*) – The full name of the booking contact.
- **customer_email** (*string*) – The email address of the booking contact.
- **summary** (*string*) – A brief description of the contents of the booking.
- **date_desc** (*string*) – A string describing the booked dates/times.
- **token** (*string*) – A customer token for the booking, can be used to build links to customer portions of the reservation system.


```
{
    "booking/index": {
        "3149": {
            "booking_id": 3149,
            "code": "PVCA-050314",
            "status_id": "PAID",
            "status_name": "Paid",
            "created_date": 1394079749,
            "total": "141.90",
            "tax_total": "12.90",
            "paid_total": "141.90",
            "customer_name": "Your Customer",
            "customer_email": "support@checkfront.com",
            "summary": "Islands Snorkeling Tour",
            "date_desc": "Wed Mar 5, 2014",
            "token":
↪"57c2ac0a58dc02a72df50a8841b1c8a190627ff69a2435060a3ed3d469819477"
        }
    }
}
```

Booking endpoints

booking/[booking_id]

GET /api/3.0/booking/{booking_id}
Retrieve extended information on a specific booking.

Parameters

- **booking_id** (*string*) – The unique booking code identifying a booking in the system.

booking/[booking_id]/invoice

GET /api/3.0/booking/{booking_id}/invoice
Return a pre-formatted invoice to display to the customer.

Parameters

- **booking_id** (*string*) – The unique booking code identifying a booking in the system.

booking/[booking_id]/checkin

You can check-in and check-out a booking. By default, a note is created under the name of the account when a booking is either checked in or checked out. VOID, and CANCELLED bookings cannot be checked-in or out.

POST /api/3.0/booking/{booking_id}/checkin

Parameters

- **booking_id** (*string*) – The unique booking code identifying a booking in the system.

booking/[booking_id]/checkout

POST /api/3.0/booking/{booking_id}/checkout

Parameters

- **booking_id** (*string*) – The unique booking code identifying a booking in the system.

booking/[booking_id]/update

The state of an existing booking can be modified using calls to booking/update.

Specifying the `status_id` is **required**, but any other update fields are optional.

POST /api/3.0/booking/{booking_id}/update

Parameters

- **booking_id** (*string*) – The unique booking code identifying a booking in the system.

Query Parameters

- **status_id** (*string*) – The status that this booking should be set to. See **Manage / Layout / Statuses** in your account for a list of all available statuses. The default available statuses are: **PEND, HOLD, PART, PAID, WAIT, STOP,** and **VOID**

- **notify** (*boolean*) – Toggle whether to trigger notifications when this booking is changed. (default: 1)
- **set_paid** (*boolean*) – When set to **1** (true) on an *unpaid* booking, and the requested status_id is '**PAID**', attempt to create a POS transaction covering the remaining cost of the booking (cannot be used with other input).

booking/[booking_id]/bookmark

Bookmarks are made available in the Checkfront mobile apps, and are listed under bookings while logged into the platform. You can add or remove a bookmark to a specific booking for the account you are acting on behalf of.

POST /api/3.0/booking/{booking_id}/bookmark

Parameters

- **booking_id** (*string*) – The unique booking code identifying a booking in the system.

Query Parameters

- **mark** (*boolean*) – Enable or disable the bookmark.

booking/[booking_id]/note

Notes can be added to bookings for reference. These will be logged in the booking as being made by the account you are acting on behalf of, so can identify comments made by individual staff members.

POST /api/3.0/booking/{booking_id}/note

Parameters

- **booking_id** (*string*) – The unique booking code identifying a booking in the system.

Query Parameters

- **body** (*string*) – The text to include in your booking note. Up to 3000 chars.

booking/create

A call to booking/create must pass in a **session_id** for a *booking/session* containing at least one *SLIP* in addition to customer input entered to fields from *booking/form*

POST /api/3.0/booking/create

Attempt to create a booking from an existing session, with customer form input sent in the request in the "form" parameter.

Form Parameters

- **form** – An array of fields containing customer details matching the required and optional booking fields (e.g. `form[customer_name]="John Smith"`)
- **session_id** – The session ID containing the booking items to be committed. *Can also be sent as a cookie.*

booking/form

This call can easily be cached if your configuration does not often change, but depending on your setup, a number of the fields returned in this request may be required to complete a booking.

You will receive an error indicating missing information if you attempt to call *booking/create* without the required fields, either those required by staff if you're acting on behalf of a staff member, or by a customer otherwise.

GET /api/3.0/booking/form
Retrieve the customer details input form for the booking. Some fields may be required to call booking/create.

Response JSON Object

- **booking_form_ui** – An array of form field definitions


```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "_en",
        "currency": "CAD"
    },
     "request": {
        "status": ""
    },
    "booking_form_ui": {
        "msg": [],
        "errors": 0,
        "mode": "html",
        "_cnf": [],
        "customer_name": {
            "value": "",
            "define": {
                "required": 1,
                "layout": {
                    "staff": {
                        "invoice": 1,
                        "form": 1,
                        "required": 1
                    },
                    "customer": {
                        "invoice": 1,
                        "form": 1,
                        "required": 1
                    },
                    "sys": 0,
                    "lbl": "Name",
                    "type": "text"
                }
            }
        },
        "booking_policy":{
            "body":"This is your <strong style='color: firebrick'>sample booking
↪policy<\/strong>.  It's included in customer e-mails, your booking receipt page
↪and in PDF print outs.\n\nYou can edit this policy by logging into your
↪Checkfront account, and navigating to Manage \/ Layout \/ Invoice.",
```

```
            "fields":{
                "customer_tos_agree":{
                    "value":0,
                    "define":{
                        "required":1,
                        "layout":{
                            "staff":{
                                "invoice":1,
                                "form":1,
                                "required":0
                            },
                            "customer":{
                                "invoice":1,
                                "form":1,
                                "required":1
                            },
                            "lbl":"I have read and agreed to the Terms of Service
↪",
                            "type":"checkbox"
                        }
                    }
                }
            },
            "customer_email": {
                "value": "",
                "define": {
                    "mask": "email",
                    "layout": {
                        "staff": {
                            "invoice": 1,
                            "form": 1,
                            "required": 0
                        },
                        "customer": {
                            "invoice": 1,
                            "form": 1,
                            "required": 1
                        },
                        "sys": 0,
                        "lbl": "E-mail",
                        "type": "text"
                    },
                    "required": 0
                }
            },
            "customer_phone": {
                "value": "",
                "define": {
                    "mask": "simple",
                    "layout": {
                        "staff": {
                            "invoice": 1,
                            "form": 1,
                            "required": 0
                        },
                        "customer": {
                            "invoice": 1,
```

```
                        "form": 1,
                        "required": 0
                    },
                    "sys": 0,
                    "lbl": "Phone",
                    "type": "text",
                    "customer_address": true
                },
                "required": 0
            }
        },
        "customer_address": {
            "value": "",
            "define": {
                "layout": {
                    "staff": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "customer": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "sys": 0,
                    "lbl": "Address",
                    "type": "text",
                    "customer_address": true
                },
                "required": 0
            }
        },
        "customer_city": {
            "value": "",
            "define": {
                "layout": {
                    "staff": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "customer": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "sys": 0,
                    "lbl": "City",
                    "type": "text",
                    "customer_address": true
                },
                "required": 0
            }
        },
        "customer_country": {
            "value": "CA",
            "define": {
```

```
                "layout": {
                    "staff": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "customer": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "sys": 0,
                    "lbl": "Country",
                    "type": "select",
                    "customer_address": true,
                    "options": {
                        "AF": "Afghanistan",
                        "AL": "Albania",
                        "DZ": "Algeria",
                        "AS": "American Samoa",
                        "AD": "Andorra",
                        "AO": "Angola",
                        "AG": "Antigua And Barbuda",
                        "AR": "Argentina",
                        "AM": "Armenia",
                        "AU": "Australia",
                        "AT": "Austria",
                        "AZ": "Azerbaijan",
                        "BS": "Bahamas",
                        "BH": "Bahrain",
                        "BD": "Bangladesh",
                        "BB": "Barbados",
                        "BY": "Belarus",
                        "BE": "Belgium",
                        "BZ": "Belize",
                        "BJ": "Benin",
                        "BM": "Bermuda",
                        "BT": "Bhutan",
                        "BO": "Bolivia",
                        "BA": "Bosnia And Herzegovina",
                        "BW": "Botswana",
                        "BR": "Brazil",
                        "BN": "Brunei Darussalam",
                        "BG": "Bulgaria",
                        "BF": "Burkina Faso",
                        "BI": "Burundi",
                        "KH": "Cambodia",
                        "CM": "Cameroon",
                        "CA": "Canada",
                        "CV": "Cape Verde",
                        "KY": "Cayman Islands",
                        "CF": "Central African Republic",
                        "TD": "Chad",
                        "CL": "Chile",
                        "CN": "China",
                        "CO": "Colombia",
                        "KM": "Comoros",
                        "CD": "Congo (DC)",
```

```
                "CG": "Congo (R)",
                "CK": "Cook Islands",
                "CR": "Costa Rica",
                "HR": "Croatia",
                "CU": "Cuba",
                "CY": "Cyprus",
                "CZ": "Czech Republic",
                "DK": "Denmark",
                "DJ": "Djibouti",
                "DM": "Dominica",
                "DO": "Dominican Republic",
                "TP": "East Timor",
                "EC": "Ecuador",
                "EG": "Egypt",
                "SV": "El Salvador",
                "GQ": "Equatorial Guinea",
                "ER": "Eritrea",
                "EE": "Estonia",
                "ET": "Ethiopia",
                "FK": "Falkland Islands",
                "FO": "Faroe Islands",
                "FJ": "Fiji",
                "FI": "Finland",
                "FR": "France",
                "PF": "French Polynesia",
                "GA": "Gabon",
                "GM": "Gambia",
                "GE": "Georgia",
                "DE": "Germany",
                "GH": "Ghana",
                "GI": "Gibraltar",
                "GR": "Greece",
                "GL": "Greenland",
                "GD": "Grenada",
                "GP": "Guadeloupe",
                "GU": "Guam",
                "GT": "Guatemala",
                "GN": "Guinea",
                "GW": "Guinea-bissau",
                "HT": "Haiti",
                "VA": "Holy See",
                "HN": "Honduras",
                "HK": "Hong Kong",
                "HU": "Hungary",
                "IS": "Iceland",
                "IN": "India",
                "ID": "Indonesia",
                "IR": "Iran",
                "IQ": "Iraq",
                "IE": "Ireland",
                "IL": "Israel",
                "IT": "Italy",
                "CI": "Ivory coast",
                "JM": "Jamaica",
                "JP": "Japan",
                "JO": "Jordan",
                "KZ": "Kazakhstan",
                "KE": "Kenya",
```

```
                "KI": "Kiribati",
                "KW": "Kuwait",
                "KG": "Kyrgyzstan",
                "LA": "Laos",
                "LV": "Latvia",
                "LB": "Lebanon",
                "LS": "Lesotho",
                "LR": "Liberia",
                "LI": "Liechtenstein",
                "LT": "Lithuania",
                "LU": "Luxembourg",
                "LY": "Lybia",
                "MO": "Macao",
                "MK": "Macedonia",
                "MG": "Madagascar",
                "MW": "Malawi",
                "MY": "Malaysia",
                "MV": "Maldives",
                "ML": "Mali",
                "MT": "Malta",
                "MQ": "Martinique",
                "MR": "Mauritania",
                "MU": "Mauritius",
                "MX": "Mexico",
                "MD": "Moldova",
                "MC": "Monaco",
                "MN": "Mongolia",
                "MS": "Montserrat",
                "MA": "Morocco",
                "MZ": "Mozambique",
                "MM": "Myanmar",
                "NA": "Namibia",
                "NR": "Nauru",
                "NP": "Nepal",
                "NL": "Netherlands",
                "AN": "Netherlands Antilles",
                "NC": "New Caledonia",
                "NZ": "New Zealand",
                "NI": "Nicaragua",
                "NE": "Niger",
                "NG": "Nigeria",
                "KP": "North Korea",
                "MP": "Northern Mariana Islands",
                "NO": "Norway",
                "OM": "Oman",
                "PK": "Pakistan",
                "PW": "Palau",
                "PS": "Palestine",
                "PA": "Panama",
                "PG": "Papua New Guinea",
                "PY": "Paraguay",
                "PE": "Peru",
                "PH": "Philippines",
                "PL": "Poland",
                "PT": "Portugal",
                "PR": "Puerto Rico",
                "QA": "Qatar",
                "RE": "Reunion",
```

```
                        "RO": "Romania",
                        "RU": "Russian Federation",
                        "RW": "Rwanda",
                        "SPM": "Saint-Pierre et Miquelon",
                        "WS": "Samoa",
                        "SM": "San Marino",
                        "ST": "Sao Tome And Principe",
                        "SA": "Saudi Arabia",
                        "SN": "Senegal",
                        "YU": "Serbia and Montenegro",
                        "SC": "Seychelles",
                        "SL": "Sierra Leone",
                        "SG": "Singapore",
                        "SK": "Slovakia",
                        "SI": "Slovenia",
                        "SB": "Solomon Islands",
                        "SO": "Somalia",
                        "ZA": "South Africa",
                        "KR": "South Korea",
                        "ES": "Spain",
                        "LK": "Sri Lanka",
                        "VC": "St Vincent And The Grenadines",
                        "SD": "Sudan",
                        "SR": "Suriname",
                        "SZ": "Swaziland",
                        "SE": "Sweden",
                        "CH": "Switzerland",
                        "SY": "Syria",
                        "TW": "Taiwan",
                        "TJ": "Tajikistan",
                        "TZ": "Tanzania",
                        "TH": "Thailand",
                        "TG": "Togo",
                        "TO": "Tonga",
                        "TT": "Trinidad And Tobago",
                        "TN": "Tunisia",
                        "TR": "Turkey",
                        "TM": "Turkmenistan",
                        "TV": "Tuvalu",
                        "UG": "Uganda",
                        "UA": "Ukraine",
                        "AE": "United Arab Emirates",
                        "GB": "United Kingdom",
                        "US": "United States",
                        "UY": "Uruguay",
                        "UZ": "Uzbekistan",
                        "VU": "Vanuatu",
                        "VE": "Venezuela",
                        "VN": "Vietnam",
                        "VG": "Virgin Islands, British",
                        "VI": "Virgin Islands, U.S.",
                        "YE": "Yemen",
                        "ZM": "Zambia",
                        "ZW": "Zimbabwe"
                    }
                },
                "required": 0
            }
```

```
        },
        "customer_region": {
            "value": "BC",
            "define": {
                "layout": {
                    "staff": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "customer": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "sys": 0,
                    "lbl": "Province",
                    "type": "select",
                    "customer_address": true,
                    "options": {
                        "AB": "Alberta",
                        "BC": "British Columbia",
                        "MB": "Manitoba",
                        "NB": "New Brunswick",
                        "NF": "Newfoundland",
                        "NWT": "Northwest Territories",
                        "NS": "Nova Scotia",
                        "NU": "Nunavut",
                        "ON": "Ontario",
                        "PE": "Prince-Edward Island",
                        "QC": "Quebec",
                        "SK": "Saskatchewan",
                        "YK": "Yukon"
                    }
                },
                "required": 0
            }
        },
        "customer_postal_zip": {
            "value": "",
            "define": {
                "mask": "simple",
                "layout": {
                    "staff": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "customer": {
                        "invoice": 1,
                        "form": 1,
                        "required": 0
                    },
                    "sys": 0,
                    "lbl": "Postal code",
                    "type": "text",
                    "customer_address": true
                },
```

```
                    "required": 0
                }
            },
            "note": {
                "value": "",
                "define": {
                    "layout": {
                        "staff": {
                            "invoice": 0,
                            "form": 1,
                            "required": 0
                        },
                        "customer": {
                            "invoice": 0,
                            "form": 1,
                            "required": 0
                        },
                        "sys": 1,
                        "lbl": "Note",
                        "type": "textarea"
                    },
                    "required": 0
                }
            }
        }
}
```

booking/index

GET /api/3.0/booking/index

Retrieve a listing of bookings in the system.

Query timestamps may be Date strings or unix timestamps, and can be prefixed with '<' or '>' to match before or after a date.

Query Parameters

- **status_id** (*string*) – The current status of a booking.
- **customer_id** (*integer*) – The customer id associated with the booking.
- **customer_email** (*string*) – The customer email associated with the booking.
- **start_date** (*string/timestamp*) – The date the booking starts on (i.e. check-in).
- **end_date** (*string/timestamp*) – The date the booking ends on (i.e. check-out).
- **last_modified** (*string/timestamp*) – The date the booking was last changed. Useful for getting bookings added or changed since your last call.
- **limit** (*integer*) – The limit of bookings to return per page (default: 100).
- **page** (*integer*) – The page of results to return.

Response JSON Array of Objects

- **booking_id** (*integer*) – The internal index of a booking.
- **code** (*string*) – The unique booking reference code.
- **status_id** (*string*) – The status code of the booking.

- **status_name** (*string*) – The display name of the booking status.
- **created_date** (*timestamp*) – The date/time the booking was created.
- **total** (*float*) – The full value of the booking.
- **tax_total** (*float*) – The sum of the taxes applied to the booking.
- **paid_total** (*float*) – The amount the customer has paid on the booking.
- **customer_name** (*string*) – The full name of the booking contact.
- **customer_email** (*string*) – The email address of the booking contact.
- **summary** (*string*) – A brief description of the contents of the booking.
- **date_desc** (*string*) – A string describing the booked dates/times.
- **token** (*string*) – A customer token for the booking, can be used to build links to customer portions of the reservation system.


```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "path": [
            "booking",
            "index"
        ],
        "id": "index",
        "resource": "booking",
        "records": 100,
        "total_records": 11985,
        "status": "OK",
        "limit": 100,
        "page": 1,
        "pages": 120,
        "time": 0.0501,
        "method": "get"
    },
    "booking/index": {
        "2": {
            "booking_id": 2,
            "code": "CFPP-290317",
            "status_id": "PEND",
            "status_name": "Pending",
            "created_date": 1490854462,
            "total": "80.40",
            "tax_total": "13.40",
            "paid_total": "0.00",
            "customer_id": 3,
            "customer_name": "Bridgette Hammes",
            "customer_email": "bridgette@example.com",
            "summary": "Walking Tour with a Taste of London, The London␣
↪Experience Hat, The London Experience Mini-Bus",
```

```
            "date_desc": "Sun Dec 24, 2017",
            "tid": "",
            "token":
↪"b93eb6589f3ac8bdfca07f26dbe1a73ff6354620a51d8609a54a6697e686567f"
        }
    }
}
```

booking/session

While creating a booking, the details of the proposed booking are stored in an API session. This allows you to add multiple items to a booking, remove items and inform the server of your intent to book the item(s) in order to prevent over-bookings.

Each session call will return the **session_id** of the active session, and any attached session information. Your application **must** use the **session_id** in order to further modify the session or submit the booking beyond this point, and can pass it back to the API either as a **cookie** or a request parameter.

To start a new session, you simply pass in the booking *SLIP*s returned when from earlier "**rated**" inventory *item* queries.

`POST /api/3.0/booking/session`

Create or return information about and stored in the current server session. Once initiated, you can fetch the details of a proposed booking by accessing the **session** again with the **session_id** found in the response. The item details for the session will be returned with your request.

Form Parameters

- **`string session_id`** – The session ID to be loaded or written to. *Can also be sent as a cookie.*
- **`string/array slip`** – A *SLIP*, or multiple *SLIP*s to be **added** to a session. If need to add or remove more of the *same* item to a session, use **alter** below.
- **`array alter`** – Alterations to be made on the current session, based on the **line_id** of items in the session and actions to be taken. To change the **qty** of an item in the session, send an alter for the line_id of that item with the integer value you need to set (e.g alter[3]=5 sets item 3 to a qty of 5). To **remove** an item, use alter 'remove'; and to opt-in or out of listed package items use 'optin' or 'optout' respectively.

`GET /api/3.0/booking/session`

Query Parameters

- **`session_id`** (*`string`*) – The session ID to read information from. *Can also be sent as a cookie.*

```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
```

```
        "status": "OK",
        "resource": "booking",
        "id": "session",
        "records": 4,
        "limit": 0,
        "page": 1,
        "pages": 1,
        "time": 0.0132,
        "method": "get"
    },
    "booking": {
        "session": {
            "id": "60rq7eme0tmh1mpkeg2fb2p287",
            "age": 0,
            "summary": "Bungee Jumping from the Top of the O2 Arena",
            "start_date": 1507186800,
            "end_date": 1507186800,
            "date_desc": "Thu Oct 5, 2017",
            "time_desc": "",
            "account_id": 1,
            "partner_id": 0,
            "currency_id": "CAD",
            "sub_total": "130.00",
            "tax_total": "0.00",
            "tax_inc_total": "0.00",
            "discount": "0.00",
            "total": "130.00",
            "due": "130.00",
            "paid_total": "0.00",
            "flat_discount": {
                "total": 0,
                "total_pretax": 0
            },
            "deposit": null,
            "tax": [],
            "qty": 1,
            "item": {
                "1": {
                    "sku": "bungeejumpingfromthetopoftheo2arena",
                    "item_id": 18,
                    "name": "Bungee Jumping from the Top of the O2 Arena",
                    "unit": "D",
                    "rate": {
                        "total": "130.00",
                        "item_total": "130.00",
                        "qty": 1,
                        "summary": "<strong title="Adults">Adults:</strong> 1 Day @
↪$130.00"
                    },
                    "date": {
                        "summary": "Thu Oct 5, 2017",
                        "start_date": 1507186800,
                        "end_date": 1507186800
                    },
                    "slip": "18.20171005X1-adults.1-children.0",
                    "available": 9
                },
                "1.1": {
```

```
                    "sku": "memorabiliaphoto",
                    "item_id": 21,
                    "name": "Memorabilia - Photo Package",
                    "unit": "D",
                    "rate": {
                        "total": "18.00",
                        "item_total": "0.00",
                        "qty": 1,
                        "summary": "$18.00"
                    },
                    "date": {
                        "summary": "Thu Oct 5, 2017",
                        "start_date": 1507186800,
                        "end_date": 1507186800
                    },
                    "slip": "21.20171005X1-1",
                    "available": 1000000000,
                    "optin": "O",
                    "opt": "out"
                },
                "1.2": {
                    "sku": "thelondonexperiencet-hat",
                    "item_id": 20,
                    "name": "The London Experience Hat",
                    "unit": "D",
                    "rate": {
                        "total": "15.00",
                        "item_total": "0.00",
                        "qty": 1,
                        "summary": "$15.00"
                    },
                    "date": {
                        "summary": "Thu Oct 5, 2017",
                        "start_date": 1507186800,
                        "end_date": 1507186800
                    },
                    "slip": "20.20171005X1-1",
                    "available": 1000000000,
                    "optin": "O",
                    "opt": "out"
                },
                "1.3": {
                    "sku": "thelondonexperiencet-shirt",
                    "item_id": 19,
                    "name": "The London Experience Mini-Bus",
                    "unit": "D",
                    "rate": {
                        "total": "25.00",
                        "item_total": "0.00",
                        "qty": 1,
                        "summary": "$25.00"
                    },
                    "date": {
                        "summary": "Thu Oct 5, 2017",
                        "start_date": 1507186800,
                        "end_date": 1507186800
                    },
                    "slip": "19.20171005X1-1",
```

```
                    "available": 1000000000,
                    "optin": "O",
                    "opt": "out"
                }
            }
        }
    }
}
```

POST /api/3.0/booking/session/clear

POST /api/3.0/booking/session/end

Query Parameters

- **session_id** (*string*) – The session ID to empty or close. *Can also be sent as a cookie.*

3.2.3 category

GET /api/3.0/category

Retrieve a list of the available categories in the system.

Response JSON Array of Objects

- **category_id** (*integer*) – The unique ID of the category.
- **name** (*string*) – The display name of the category.
- **pos** (*integer*) – The order in which the category will display.


```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "status": "OK",
        "resource": "category",
        "records": 6,
        "limit": 0,
        "page": 1,
        "pages": 1,
        "time": 0.0028,
        "method": "get"
    },
    "query": [],
    "category": {
        "2": {
            "category_id": 2,
            "name": "Tours",
            "description": "From Walking, to Bicycles, to Double Decker buses, we have the perfect tour for you and your family.",
            "pos": 20,
            "image": "/media/Lcat-2.jpg",
```

```
            "qty": 4
        },
        "3": {
            "category_id": 3,
            "name": "Events",
            "description": "Special Events brought to you by The London Experience. Follow us on Twitter to keep up to date!",
            "pos": 18,
            "image": "/media/Lcat-3.jpg",
            "qty": 2
        },
        "4": {
            "category_id": 4,
            "name": "Accommodations",
            "description": "Book a Double or Single room for your overnight stay in Manchester.",
            "pos": 17,
            "image": "/media/Lcat-4.jpg",
            "qty": 4
        },
        "5": {
            "category_id": 5,
            "name": "Gift Certificates",
            "description": "Give the Gift of The London Experience.",
            "pos": 16,
            "image": "/media/Lcat-5.jpg",
            "qty": 4
        },
        "6": {
            "category_id": 6,
            "name": "Activities",
            "description": "",
            "pos": 19,
            "image": "/media/Lcat-6.jpg",
            "qty": 3
        },
        "7": {
            "category_id": 7,
            "name": "Merchandise",
            "description": "",
            "pos": 14,
            "image": "/media/Lcat-7.jpg",
            "qty": 3
        }
    }
}
```


3.2.4 company

GET /api/3.0/company

Retrieve company configuration information of the system you are connecting to.

Response JSON Object

- **url** (*string*) – The Checkfront URL of the account.
- **name** (*string*) – The display name of the company.

- **plan_id** (*integer*) – The Checkfront subscription plan ID of the account.
- **email** (*string*) – The contact email address of the company.
- **address** (*string*) – The street address of the company.
- **city** (*string*) – The city where the company is located.
- **postal_zip** (*string*) – The postal or zip code of the company.
- **region** (*string*) – The state/province of the company.
- **region_id** (*string*) – The state/province of the company.
- **country_id** (*string*) – The country where the company is based.
- **currency_id** (*string*) – The currency type accepted by the company.
- **timezone** (*string*) – The timezone of the company.
- **locale_id** (*string*) – The locale string set for the company.
- **lang_id** (*string*) – The ISO 639-1 language code of the company.
- **date_format** (*string*) – The date format set for the company.
- **time_format** (*string*) – The time format set for the company.


```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_GB",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "status": "OK",
        "resource": "company",
        "records": 19,
        "limit": 0,
        "page": 1,
        "pages": 1,
        "time": 0.0016,
        "method": "get"
    },
    "company": {
        "url": "your-company.checkfront.com",
        "name": "Your Company",
        "plan_id": 1,
        "email": "hello@checkfront.com",
        "address": "",
        "city": "",
        "postal_zip": "",
        "region": "LND",
        "country_id": "CA",
        "currency_id": "CAD",
        "timezone": "Canada/Pacific",
        "locale_id": "en_GB",
        "lang_id": "en",
        "date_format": "%m/%d/%y",
```

```
        "time_format": "%I:%M %p",
        "phone": "",
        "utc_offset": -7,
        "currency": {
            "symbol": "$",
            "symbol_space": 0,
            "symbol_precedes": 1,
            "decimals": 2,
            "decimal_separator": ".",
            "thousands_separator": ","
        }
    },
    "hours": false
}
```

3.2.5 customer

The customer object provides access to customer details stored in the system. Customer information is automatically stored in the system when a new booking is made.

To retrieve information on a specific customer, you should ideally use the customer ID in a GET call to **customer/[customer_id]**. In this case, the customer information will be returned as "customer" in the JSON response.

If you do not have the customer ID stored, you can also look up customers based on indexed fields, including email address, name, and phone number. These must be an exact match, but can return multiple results on non-unique fields such as the customer name. The results will be stored in the "customers" object in the JSON response.

GET /api/3.0/customer
Retrieve a list of customers.

Query Parameters

- **customer_id** (*string*) – The unique customer code identifying a customer in the system.
- **customer_email** (*string*) – A unique customer e-mail address to search.
- **customer_name** (*string*) – The full name of the customer to search.
- **customer_phone** (*string*) – The (exact) customer phone number to search.

GET /api/3.0/customer/{customer_id}
Retrieve contact details for a customer.

Parameters

- **customer_id** (*string*) – The unique customer code identifying a customer in the system.

```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_CA",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
```

```
        "status": ""
    },
    "customer": {
        "customer_id": 25,
        "meta_id": null,
        "code": "NQ2-039-421",
        "status_id": "ACTIVE",
        "last_booking_date": null,
        "customer_name": "John Smith",
        "customer_email": "support@checkfront.com",
        "customer_email_optin": null,
        "customer_region": "BC",
        "customer_city": "Victoria",
        "customer_address": "386 Mews St",
        "customer_country": "0",
        "customer_postal_zip": "12345",
        "customer_phone": "0",
        "customer_crm_id": null,
        "notes": null,
        "meta": {
            "source": false,
            "diet": false,
            "diet_allergies": false
        },
        "referer": null,
        "bookings": {
            "LZTM-290317": {
                "created_date": "2017-03-29T23:54:18-07:00"
            },
            "XJZQ-020417": {
                "created_date": "2017-04-02T23:19:29-07:00"
            }
        }
    }
}
```


3.2.6 event

Events are used to modify the price or availability of your inventory items based on date. The **event** endpoint allows you to retrieve information on events in the system, and see which items they are applied to.

GET /api/3.0/event
Retrieve a list of all events in the system.

Response JSON Array of Objects

- **events** (*array*) – A list of all events in the system. See *Event Object Structure* below.

GET /api/3.0/event/{id}
Retrieve data on a specific event.

Response JSON Array of Objects

- **event** (*object*) – Information about the requested event. See *Event Object Structure* below.

Event Object Structure

Field	Type	Description
rate_id	*integer*	A unique code used to refer to this event
name	*string*	The name of this event
type	*string*	Item Events: "SP" for "Special" events, "SE" for "Seasonal" events. Discounts: "DC" for "Discount"
start_date	*integer*	The date *(yyyymmdd)* when this event starts to apply
end_date	*integer*	The date *(yyyymmdd)* when this event stops applying (0 if the event has no end date)
span	*integer*	The "Force item length to the above start and end dates" option (1 if selected, 0 if not selected)
price_src	*string*	If applicable, the type of price modification for this event ("Base Price" is "B", "Create new Price Point" is "N", "Dynamic" is "D", "Yield" is "Yield")
dynamic_rate	*decimal*	If this event is a "Dynamic" price event (*price_src* is "D"), this value holds the event's percent change or fixed amount price.
dynamic_type	*string*	Dynamic Price events: "P" for percent, "F" for fixed amount
status	*string*	Item Events: whether this is an available ("A") or unavailable ("U") event. Discounts: always available ("A")
repeat_id	*string*	If the event is not recurring, this value is blank. Weekly recurring events are "W", "Always" recurring events are "*".
enabled	*integer*	If the event is enabled, 1, otherwise 0 when disabled.
rule_set_id	*integer*	The ruleset id applying to this event. The default ruleset id is 1.
vouchers	*integer*	Discounts: the number of vouchers attached to this discount.
details	*array*	If this is event has a "Yield" **price_src**, this element will contain a list of objects containing a level and rate pair for each threshold. See *event.details objects* below
repeat	*array*	A list of days (mon, tue, . . .) for which this event repeats.
apply_to	*object*	See *event.apply_to* below.

event.details objects

Field	Type	Description
level	*integer*	The inventory threshold at which this Yield priced event applies
rate	*decimal*	The percentage price multiplier for this threshold

event.apply_to

Field	Type	Description
item_id	*array*	This array contains the list of item IDs which this event applies to.
category_id	*array*	This array contains the list of category IDs which this event applies to. **This event will apply to any new items that are added to this category.**

Sample Event Object


```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "path": [
            "event"
        ],
        "resource": "event",
        "records": 6,
        "total_records": 6,
        "status": "OK",
        "limit": 100,
        "page": 1,
        "pages": 1,
        "time": 0.0028,
        "method": "get"
    },
    "events": {
        "1": {
            "rate_id": 1,
            "name": "Max 3 Days",
            "type": "SP",
            "start_date": 20161221,
            "end_date": 0,
            "span": 0,
            "price_src": "Yield",
            "dynamic_rate": 10,
            "dynamic_type": "P",
            "status": "A",
            "repeat_id": "W",
            "enabled": 1,
            "rule_set_id": 1,
            "details": [
                {
                    "level": 5,
                    "rate": 150.5
                },
                {
```

```
                    "level": 2,
                    "rate": 200
                }
            ],
            "repeat": [ "tue", "sat" ],
            "apply_to": {
                "item_id": [ 14, 16, 35, 36, 37 ],
                "category_id": [ 3, 5 ]
            }
        },
        "2": {
            "rate_id": 2,
            "name": "Easter Closure",
            "type": "SE",
            "start_date": "20170417",
            "end_date": "20170417",
            "span": 0,
            "price_src": "",
            "dynamic_rate": "0.0000",
            "dynamic_type": "P",
            "status": "U",
            "repeat_id": "",
            "enabled": 1,
            "rule_set_id": 1,
            "details": [],
            "repeat": [],
            "apply_to": {
                "item_id": [
                    "all"
                ],
                "category_id": []
            }
        }
    }
}
```


3.2.7 item

Your Checkfront inventory is made up of '**items**' that are added to bookings, and have complex availability controls and rules defining their stock and prices.

When no dates are passed in the API call, a full list of enabled items in the inventory is displayed, without specific availability and pricing.

However, if you pass in the relevant booking dates (and optionally parameters, discount codes, etc), the API will return a "**rated**" response that includes pricing and availablity, as well as a "*SLIP*" that will be used to add the item to a session.

SLIP

The item SLIP is an encoded string returned when making a rated query to a specific item. The slip contains the information needed to make a booking. Don't attempt to reverse engineer this, as the format changes. It must be optioned via a rated API call.

Booking Parameters

Booking parameters are defined globally in your system, and can be configured on a per item basis. Your parameters specify the number of items to book, for example Child and Adult. These are completely configurable in your Checkfront account under **Manage / Settings / Configure**.

To query *specific* pricing and availability, you need to pass the appropriate parameters in your API call (using the parameter IDs as specified in your configuration).

Say for instance you have 2 parameters configured: Adults (id: adults), and Children (id: children). To get pricing for 2 adults and one child you would pass: **param[adults]=2&param[children]=1** in your API call.

Booking parameters have many options, and can be configured to control inventory in very specific ways. See the Checkfront support centre or ask us for more information.

Request

GET /api/3.0/item
Retrieve a list of the enabled items in the system.

Query Parameters

- **start_date** (*date*) – (rated) Booking start date.
- **end_date** (*date*) – (rated) Booking end date.
- **date** (*date*) – (rated) Alias of **start date** (for same-day bookings).
- **start_time** (*date*) – (rated) Start time (used in hourly bookings).
- **end_time** (*date*) – (rated) End time (used in hourly bookings).
- **category_id** (*integer*) – Filter items by category.
- **packages** (*boolean*) – Show package options.
- **available** (*integer*) – (rated) Filter to items with at least this many left in stock.
- **keyword** (*string*) – Filter to items with a name containing this keyword.
- **item_id** (*string*) – A comma-seperated list of items to filter to.
- **discount_code** (*string*) – (rated) The discount code to apply to the price.
- **rules** (*string*) – 'soft' will prevent triggering date based rule errors, or 'off' will disable rule checking.
- **param** (*array*) – (rated) See *Booking Parameters*.

GET /api/3.0/item/{item_id}
Retrieve details for a single item. Shares same params as above for rated requests.

Parameters

- **item_id** (*string*) – The unique item_id of the item to query, as found in a response or your Checkfront panel.

GET /api/3.0/item/{item_id}/cal
Retrieve calendar availability for a single item.

Parameters

- **item_id** (*string*) – The unique item_id of the item to query.

Query Parameters

- **start_date** (*date*) – Availability range start date.
- **end_date** (*date*) – Availability range start date.

GET /api/3.0/item/cal

Retrieve calendar availability for a set of items.

Query Parameters

- **item_id** (*array|integer*) – The unique item_id of the item to query, or an array of item IDs
- **category_id** (*integer*) – A category of items to filter to.
- **start_date** (*date*) – Availability range start date.
- **end_date** (*date*) – Availability range start date.

Unrated Response


```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "status": "OK",
        "resource": "item",
        "id": "18",
        "records": 0,
        "limit": 0,
        "page": 1,
        "pages": 1,
        "time": 0.0027,
        "method": "get"
    },
    "query": [],
    "item": {
        "unit": "D",
        "item_id": 18,
        "sku": "bungeejumpingfromthetopoftheo2arena",
        "url": "",
        "lock": 1,
        "visibility": "*",
        "name": "Bungee Jumping from the Top of the O2 Arena",
        "pos": 0,
        "summary": "<p>Take a trip to the top of London's most famous concert venue,
↪the O2 Arena, and right back down again. This exhilarating chance of a lifetime
↪starts with a tour through the Memorabilia Halls of the O2, and finishes with a 160
↪ft bungee jump. Days and space are limited, so book today!</p>",
        "details": "",
        "meta": "{\"display_mode\":\"dropdown\",\"item_package_rate\":\"\",\"delay\
↪":0}",
        "stock": 15,
```

```
        "unlimited": 0,
        "video": {
            "id": "B6MwTLgmFMM",
            "start": 0
        },
        "image": {
            "1": {
                "title": "",
                "src": "18-1--3942",
                "path": "/media/M18-1--3942.jpg"
            },
            "2": {
                "title": "",
                "src": "18-2--9804",
                "path": "/media/M18-2--9804.jpg"
            }
        },
        "category_id": 6,
        "rated": 1,
        "product_group_type": "",
        "product_group_children": "",
        "type": "I",
        "status": "A",
        "alias_id": 0,
        "len": 1,
        "rules": "{\"param\":{\"adults\":{\"MIN\":\"0\",\"MAX\":\"0\"},\"children\":{\
↪"MIN\":\"0\",\"MAX\":\"0\"}},\"fixed\":{\"start_time\":\"\"}}",
        "category": "Activities"
    }
}
```


Rated Response


```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "status": "OK",
        "resource": "item",
        "id": "18",
        "records": 0,
        "limit": 0,
        "page": 1,
        "pages": 1,
        "time": 0.0101,
        "method": "get"
    },
    "query": {
        "start_date": "20171005",
```

```
        "end_date": "20171005",
        "param": {
            "qty": "1",
            "rooms": "1",
            "adults": "1",
            "children": "0",
            "guests": "1",
            "familyticket2ad": "1",
            "familyticket": "4"
        },
        "local_end_date": "10/05/17",
        "local_start_date": "10/05/17"
    },
    "item": {
        "unit": "D",
        "item_id": 18,
        "sku": "bungeejumpingfromthetopoftheo2arena",
        "url": "",
        "lock": 1,
        "visibility": "*",
        "name": "Bungee Jumping from the Top of the O2 Arena",
        "pos": 0,
        "summary": "<p>Take a trip to the top of London's most famous concert venue,␣
↪the O2 Arena, and right back down again. This exhilarating chance of a lifetime␣
↪starts with a tour through the Memorabilia Halls of the O2, and finishes with a 160␣
↪ft bungee jump. Days and space are limited, so book today!</p>",
        "details": "",
        "meta": {
            "display_mode": "dropdown",
            "item_package_rate": "",
            "delay": 0,
            "default_len": 1
        },
        "stock": 15,
        "unlimited": 0,
        "video": {
            "id": "B6MwTLgmFMM",
            "start": 0
        },
        "image": {
            "1": {
                "title": "",
                "src": "18-1--3942",
                "path": "/media/M18-1--3942.jpg"
            },
            "2": {
                "title": "",
                "src": "18-2--9804",
                "path": "/media/M18-2--9804.jpg"
            }
        },
        "category_id": 6,
        "rated": 1,
        "type": "I",
        "status": "A",
        "alias_id": 0,
        "len": 1,
        "rules": "{\"param\":{\"adults\":{\"MIN\":\"0\",\"MAX\":\"0\"},\"children\":{\
↪"MIN\":\"0\",\"MAX\":\"0\"}},\"fixed\":{\"start_time\":\"\"}}",
```

```
        "category": "Activities",
        "package": [
            {
                "package_id": 21,
                "optin": "O",
                "select_package_params": 1
            },
            {
                "package_id": 20,
                "optin": "O",
                "select_package_params": 1
            },
            {
                "package_id": 19,
                "optin": "O",
                "select_package_params": 1
            }
        ],
        "param": {
            "adults": {
                "price": 1,
                "lbl": "Adults",
                "lock": 1,
                "qty": 1,
                "hide": 0,
                "customer_hide": 0,
                "req": 1,
                "range": 1,
                "def": 1,
                "MIN": 0,
                "MAX": 0
            },
            "children": {
                "price": 1,
                "lbl": "Children",
                "lock": 1,
                "qty": 0,
                "hide": 0,
                "customer_hide": 0,
                "req": 0,
                "range": 1,
                "def": 0,
                "MIN": 0,
                "MAX": 0
            }
        },
        "qty": 1,
        "rate": {
            "status": "AVAILABLE",
            "available": 9,
            "slip": "18.20171005X1-adults.1-children.0",
            "summary": {
                "title": "9 Available",
                "details": "<strong title='Adults'>Adults:</strong> 1 Day @ $130.00",
                "price": {
                    "total": "$130.00",
                    "title": "$110.00 - $130.00",
                    "unit": "per day",
```

```
                    "param": {
                        "adults": "$130.00",
                        "children": "$110.00"
                    }
                },
                "date": "Thu Oct 5, 2017"
            },
            "sub_total": "130.00",
            "event": [
                {
                    "adults": {
                        "amount": 130,
                        "days": 1,
                        "type": "q",
                        "range": "1-5"
                    },
                    "adults~1-5": {
                        "amount": 130,
                        "type": "q"
                    },
                    "adults~6-15": {
                        "amount": 115,
                        "type": "q"
                    },
                    "children": {
                        "amount": 110,
                        "days": 1,
                        "type": "q"
                    },
                    "children~1-5": {
                        "amount": 110,
                        "type": "q"
                    },
                    "children~6-15": {
                        "amount": 100,
                        "type": "q"
                    }
                }
            ],
            "start_date": "20171005",
            "end_date": "20171005",
            "dates": {
                "20171005": {
                    "status": "A",
                    "price": {
                        "adults": 130,
                        "children": 110
                    },
                    "dow": "4",
                    "stock": {
                        "T": 15,
                        "B": 6,
                        "A": 9,
                        "MinA": 9
                    },
                    "rates": [],
                    "discount": [],
                    "sub_total": 130
```

```
                    }
                }
            },
            "days": 1,
            "discount": {
                "amount": 0
            },
            "gprice": {
                "adults": {
                    "1-5": "q",
                    "6-15": "q"
                },
                "children": {
                    "1-5": "q",
                    "6-15": "q"
                }
            },
            "local_start_date": "10/05/17",
            "local_end_date": "10/05/17"
        },
        "events": {
            "9": {
                "name": "Group Booking",
                "start_date": "",
                "end_date": ""
            }
        }
}
```


3.2.8 help

GET /api/3.0/help

Retrieve a list of API endpoints.


```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "status": "OK",
        "resource": "help",
        "records": 15,
        "limit": 0,
        "page": 1,
        "pages": 1,
        "time": 0.001,
        "method": "get"
    },
    "help": {
        "resources": [
```

```
            "ping",
            "help",
            "company",
            "account",
            "status",
            "mobile",
            "customer",
            "ui",
            "item",
            "stock",
            "category",
            "booking",
            "import",
            "discount",
            "event"
        ],
        "see": "http://api.checkfront.com/"
    }
}
```

3.2.9 ping

GET /api/3.0/ping
Show API connection details and latency.

```
{
    "version": "3.0",
    "account_id": 1,
    "host_id": "your-company.checkfront.com",
    "name": "Your Company",
    "locale": {
        "id": "en_US",
        "lang": "en",
        "currency": "CAD"
    },
    "request": {
        "status": "OK",
        "resource": "ping",
        "records": 1,
        "limit": 0,
        "page": 1,
        "pages": 1,
        "time": 0.0009,
        "method": "get"
    },
    "pong": {
        "remote_addr": "0.0.0.0",
        "time": 1507249065,
        "timezone": "Canada/Pacific",
        "provider": "API Console"
    }
}
```

HTTP Routing Table

/api