

Giftbit RESTful Partners API

Document Version 1.5.1 - October 2015

Full changelog is at the bottom of the document.

Overview

API Base URL

Production:

<https://api.giftbit.com/papi/v1>

Test:

<https://testbedapp.giftbit.com/papi/v1>

Summary of Resource Endpoints

Marketplace: Giftbit's available gift card brands

[/marketplace](#)

[/marketplace/{id}](#)

[/marketplace/regions](#)

[/marketplace/vendors](#)

[/marketplace/categories](#)

Campaign: Send gift offers to your recipients or retrieve information on past campaigns.

[/campaign](#)

[/campaign/{giftbit campaign uuid or your supplied campaign id}](#)

Funds: Real time information on your account funding

[/funds](#)

Gifts: Detailed information on your sent gift offers

[/gifts](#)

[/gifts/{gift offer uuid}](#)

Links: Retrieve shortlinks for your link based gift offer campaigns

[/links/{giftbit campaign uuid or your supplied campaign id}](#)

Authentication

The Giftbit API uses token based authentication. Your secret token will be provided to you by Giftbit (be sure to keep it secret). You can authenticate yourself in one of two ways when making API calls (this follows OAUTH 2 specification):

1. Include your token prefixed by the word 'Bearer' (case sensitive) and a space in the "Authorization" HTTP header. The examples in this document use this format
2. Include the token in "access_token" as a url parameter. You will need to URL encode your token if using this method.

You can make a credentials check call (described below) to ensure that everything is working.

HTTP Status Codes:

200 OK - Everything looks good

400 Bad Request - The request cannot be fulfilled due to bad syntax

401 Unauthorized – Login credentials are not valid.

402 Request Failed – Parameters were valid but request failed.

403 Forbidden – Your credentials are not authorized to access this method

405 Method not allowed – The method used is not allowed for this resource.

500 Internal Server Error – Something on Giftbit's servers has gone wrong. Contact customer support if you continue to receive this error.

503 Service Unavailable - Planned maintenance or a known temporary condition resulting in downtime.

Common Responses - Info and Error objects

Many responses will include either an info or error object. An info object always be returned with an HTTP status of 200. An error object will always be returned with an HTTP status of 402. Both the info and error objects take the following form:

- code:
 - **String.** A code that corresponds to status of the request. The code will take the form of <INFO|ERROR>_<RESOURCE>_<NAME>.
- name:
 - **String.** A short description that can be displayed to the user.
- message:
 - **String.** A longer message that can be displayed to the user.

Typical Use Cases in Brief

Credentials check

To begin, get your token from Giftbit. You will need to contact Giftbit directly for your access token. The access token should be placed in "Authorization" header or (URL encoded) as the "access_token" url parameter as described in the 'Authentication' section . Check that your credentials are correct with the call:

GET - <https://api.giftbit.com/papi/v1/>

This can also serve as a Giftbit API healthcheck to build into your product.

Marketplace gift browsing criteria and getting the Id of a marketplace gift to send

In order to send a gift offer to one or more recipients, you need to get its Id and choose a value (marketplace gifts will either have one or more fixed values, or an open range to choose from). There are a couple common use-cases and recommended approaches:

Usecase 1: sending the same marketplace gift(s) all the time

A simple yet popular API usecase is to set up a certain gift or choice of gifts that is always used when sending campaigns via the API. In this usecase, it likely isn't necessary to build an interactive representation of Giftbit's gift marketplace, but rather retrieve the Ids of the marketplace gifts you want manually, choose a valid and desired gift amount, and store them in your database or configuration.

GET <https://api.giftbit.com/papi/v1/marketplace?limit=1000¤cyisocode=USD>

This will list all the available marketplace gifts, and you can manually pick out the one(s) you want. Note: the Id values returned will differ between test and production environments, so please ensure you keep environment specific configurations.

Usecase 2: providing an interactive marketplace

If you are providing a representation of Giftbit's gift marketplace that can be browsed, the API provides the capability to refine results by any combination of region, category, or vendor. The following API calls will return lists of valid refinements for each. It is recommended that the results be cached and updated at a daily (or less frequent) cadence as they change infrequently. Note: the Id values returned will differ between test and production environments, but will not change within an environment.

GET - <https://api.giftbit.com/papi/v1/marketplace/regions>

GET - <https://api.giftbit.com/papi/v1/marketplace/categories>

GET - <https://api.giftbit.com/papi/v1/marketplace/vendors>

Use the category, region and vendors returned (price range and currency parameters are also supported) to refine the list of marketplace gifts. Use the limit and offset properties to provide paging.

GET -

https://api.giftbit.com/papi/v1/marketplace?min_price_in_cents=1000&max_price_in_cents=5000®ion=2&category=5&vendor=7&limit=20&offset=20

Both use-cases: sending a gift offer

Once you have the Id(s) and values of the marketplace gift(s) you want to send, POST a JSON campaign object to the campaign endpoint. You can include more than one marketplace gift if you wish to allow the recipient(s) to choose from one of them (important: if you include more than one marketplace gift, recipients get a choice and may choose only one - they do not get all the gifts). You can optionally set some unique id that makes sense to your system to reference the campaign by - while optional, we recommend you do provide this value for idempotency (to avoid accidentally sending the same campaign more than once, as we will only allow a single campaign per supplied id). Many examples are included in the below sections.

Delivery types

Giftbit has two delivery types for your gift offers: GIFTBIT_EMAIL and SHORTLINK. Note that in each case the content Giftbit provides to the recipient is the same (the Giftbit email content is the same as the shortlink destination webpage) so all information around branding, templates and gift content applies to both delivery types. All reporting and statistics are available for both delivery methods. Examples of API calls for both delivery methods are provided below.

Giftbit Email Delivery

If you have email addresses for your recipients, email delivery is the recommended, fastest and most reliable delivery method. Giftbit specializes in email deliverability and trust best practices. We will deliver your branded gift offer to each supplied contact via an HTML email. Bounces, spam complaints, and other deliverability issues are made available to you so that you can keep your recipient lists clean and of high quality. You must include one or more recipients to send the offer to in your campaign creation request, and the `delivery_type` parameter may be omitted or set to GIFTBIT_EMAIL.

Example:

```
POST - https://api.giftbit.com/papi/v1/campaign - {"contacts": [{"firstname": "John", "lastname": "Doe", "email": "johndoe@somedomain.com"}, {"firstname": "Jane", "lastname": "Doe", "email": "janedoe@somedomain.com"}], "marketplace_gifts": [{"id": 14, "price_in_cents": 1000}], "expiry": "2015-12-20", "id": "someUniqueGiftId", "gift_template": "ABCDE", "delivery_type": "GIFTBIT_EMAIL"}
```

Note: There is no additional charge per gift offer for Giftbit Email Delivery.

Shortlink Delivery (for delivering the gift offer yourself)

In the cases where email based delivery is not appropriate or you do not have email addresses for your recipients, you may create a campaign where each gift offer is represented by a unique shortlink. To retrieve the links you poll the `/links` endpoint, and then deliver them yourself to your recipients physically or digitally as you see fit for ultimate flexibility. There are two ways to create shortlink campaigns:

1. Make the same POST call as per the Giftbit email delivery method, but change `delivery_type` to SHORTLINK. Instead of email delivery, a shortlink will be created for each contact you supply.

Example:

```
POST - https://api.giftbit.com/papi/v1/campaign - {"contacts": [{"firstname": "John", "lastname": "Doe", "email": "johndoe@somedomain.com"}, {"firstname": "Jane", "lastname": "Doe", "email": "janedoe@somedomain.com"}], "marketplace_gifts": [{"id": 14, "price_in_cents": 1000}], "expiry": "2015-12-20", "id": "someUniqueGiftId", "gift_template": "ABCDE", "delivery_type": "SHORTLINK"}
```

2. If you just want a certain number of links, you may supply a link_count number and omit the contacts.

Example:

```
POST - https://api.giftbit.com/papi/v1/campaign - {"marketplace_gifts": [{"id": 14,"price_in_cents": 1000}], "expiry": "2015-12-20", "id": "someUniqueGiftId", "gift_template": "ABCDE", "delivery_type": "SHORTLINK", "link_count": 10}
```

Note: when using delivery_type of SHORTLINK and providing contacts, Giftbit does NOT deliver an email to those contacts. The ability to provide contacts for shortlink campaigns is so that you can correlate your links to given recipients in our system (this correlation is visible anywhere we provide your gift offer history information including reports, the web application, and the /gifts endpoint) without needing to keep track of it in yours. It is up to you to ensure you deliver the correct link to the corresponding contact.

After successfully creating a campaign with a delivery_type of SHORTLINK, Giftbit provides an endpoint at /links to poll and retrieve the links. Generation time is usually within minutes, but is dependent on the number of gift offers being created.

GET - https://api.giftbit.com/papi/v1/links/myCampaignId

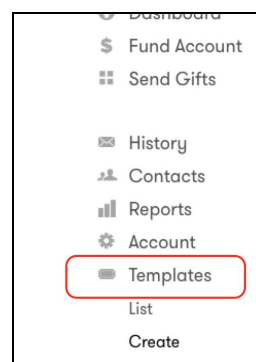
If the links are ready, they will be returned. If not, an appropriate not-yet-ready response will be returned. Full formats for the response JSON and examples are provided later in the document. We recommend polling every 5 minutes after the successful campaign creation request.

Note: We use a different short domain in testbed that resolves to the testbed platform, so you can fully test your link generation and recipient experience of following the shortlink before you get to production.

Using templates to pre-create your gift content and use rich formatting in your gift messages

Templates allow your gift messages to be richly formatted, and provide nice separation of responsibility between content creators and developers by allowing the gift content (subject, message with formatting and rich content, and company brand image) to be maintained through the web interface and WYSIWYG editor, without requiring any changes to the API calls themselves when the content changes. To use templates:

1. If you have not already done so, have your content maintainer (or for testbed, likely the developer themselves) access the templates area in the Giftbit web application (see image), and create one or more templates.



Each template will be auto-assigned a unique uppercase short identifier shown in the web interface - this is how you will reference that template in your API call.

2. Make a POST request to the /campaign endpoint with the given template identifier . Omit 'message' and 'subject' unless you want to override what is in the template.

POST - `https://api.giftbit.com/papi/v1/campaign - {"contacts": [{"firstname": "Graham", "lastname": "Baradoy", "email": "graham@giftbit.com"}], "marketplace_gifts": [{"id": 14, "price_in_cents": 1000}], "expiry": "2015-12-20", "id": "someUniqueGiftId", gift_template: 'ABCDE'}`

3. Optionally, supply a subject or message if you wish to override either or both specified in the template for that particular campaign.

POST - `https://api.giftbit.com/papi/v1/campaign - {"subject": "This subject will override the one from the template", "contacts": [{"firstname": "Graham", "lastname": "Baradoy", "email": "graham@giftbit.com"}], "marketplace_gifts": [{"id": 14, "price_in_cents": 1000}], "expiry": "2015-12-20", "id": "someUniqueGiftId", gift_template: 'ABCDE'}`

Including your subject and message in the API call when not using templates.

If you simply wish to do everything via the API call and not use templates, you must supply a subject and a message in your API call:

POST - `https://api.giftbit.com/papi/v1/campaign - {"subject": "This is the subject of the gift email", "message": "This is the greeting message of the gift email.", "contacts": [{"firstname": "Graham", "lastname": "Baradoy", "email": "graham@giftbit.com"}], "marketplace_gifts": [{"id": 230, "price_in_cents": 340}], "expiry": "2015-12-20", "id": "someUniqueGiftId"}`

Note that unlike with templates, we will automatically add a personalized greeting to your recipient's name (or a generic greeting if you only specify an email address) above your message.

Quoting

If you do not wish to send the campaign immediately, use "quote":true in the campaign JSON. This allows you to:

1. See how much a campaign will cost before you send it.
2. Set up an internal approval pipeline. This can be desirable if you have built automatic gifting into your product, but still want a human to OK the campaigns before sending them as a separate step.

To take an existing quote and send it, make a PUT call to the campaign endpoint with the Id you set for the campaign, or the uuid returned from the call.

PUT - <https://api.giftbit.com/papi/v1/campaign/someGiftId>

Resending or Canceling a gift

After you have created a campaign, a number of gifts are created equal to the number of recipients. From time to time, you may find that you wish to resend the original gift message, or cancel a gift that has already been sent out. To cancel or resend a gift, you will first need the gift's uuid. To get the uuid of a gift, you will need the uuid of the campaign, or the id assigned to the campaign in the campaign creation step. The uuid of the campaign is returned when creating a campaign, or from querying the /campaign endpoint. To get the uuid of the gift, use the uuid of the campaign.

GET - https://api.giftbit.com/papi/v1/gifts?campaign_uuid=<UUID>

This will return all of the gifts from the campaign. You can also search for the gifts sent to a particular user with the recipient_email search parameter.

GET - https://api.giftbit.com/papi/v1/gifts?recipient_email=graham@giftbit.com

This will return all of the gifts sent to that email address. Using the uuid of one of the returned gifts, you can resend it by passing it a json object with resend true.

PUT - <https://api.giftbit.com/papi/v1/gifts/<UUID>> - { "resend":true }

Likewise, you can cancel the gift with its uuid.

DELETE - <https://api.giftbit.com/papi/v1/gifts/<UUID>>

Once a gift is canceled, the recipient will no longer be able to redeem it.

Checking your account balance

Gifts in a campaign will only be sent if there are enough available funds for the campaign. The funds endpoint will return your available, pending and outstanding gift offers balances.

GET - <https://api.giftbit.com/papi/v1/funds>

API Request and Response attributes

Base:

Credentials Check

- **Overview**

- A HTTP 200 response code of 200 confirms your credentials are valid. A HTTP 403 response code will be returned if the provided credentials are not authorized to use the API. A HTTP 401 response code will be returned if the credentials provided cannot be used to login suggesting a bad username or password.

- **Parameters**

- Method: GET
- Location: /

- **Response Attributes**

- **info:**
 - INFO_CREDENTIALS - Credentials are valid - The credentials used are valid and login attempt was successful.
- **username:**
 - **String**. The username associated with the credentials provided.
- **displayname:**
 - **String**. The display name associated with the credentials provided.

Gifts In the Marketplace

Common Marketplace Objects

MarketplaceGift. A Marketplace Gift represents an item in the Giftbit's marketplace that can be given. Some gifts in the marketplace have a set of prices that you can choose from while others have an open range of prices.

- **id:**
 - **Long**. A numeric identifier for the gift. This identifier can be used with the campaign object to send gifts.
- **name:**
 - **String**. The name of the gift.
- **description:**
 - **String**. A description of the gift.
- **redemption_instructions:**
 - **String**. The redemption instructions that will be presented to the recipient.
- **terms_and_conditions:**
 - **String**. Terms and conditions associated with the gift.
- **image_url:**
 - **String**. The full url of the gift's image.
- **tax:**
 - **String**. The tax code associated with the gift. **Possible values:** HST, GST, NOTAX, and GSTANDPST.
- **vendor:**

- **Vendor. Optional.** The vendor associated with the marketplace gift.
- regions:
 - **List<Region>. Optional.** A list of regions that the marketplace gift is available in.
- categories:
 - **List<Category>. Optional.** A list of categories that the marketplace gift fits into.
- prices_in_cents:
 - **List<Integer>. Optional.** A list of values, in cents, that can be chosen when giving this gift. Either prices_in_cents or max_price_in_cents will be defined but not both.
- currencyisocode: **String** the three digit uppercase currency code that gives the currency of this gift
- max_price_in_cents:
 - **Integer. Optional.** The maximum value that can be chosen when giving this gift. max_price_in_cents will always be accompanied by min_price_in_cents.
- min_price_in_cents:
 - **Integer. Optional.** The minimum value that can be chosen when giving this gift. min_price_in_cents will always be accompanied by max_price_in_cents.

Vendor. A Vendor represents a gift vendor in Giftbit's marketplace. For example "Rdio" or "Zambri's".

- id:
 - **Long.** The id that can be used to reference a Vendor.
- name:
 - **String.** The name of the Vendor.
- image_url:
 - **String.** The full url of a logo for the Vendor.

Category. A Category such as "Cafe" or "Apparel".

- id:
 - **Long.** The id that can be used to reference a Category.
- name:
 - **String.** The name of the Category.

Region. A Region, which can be a country such as "Canada" or a city like "Victoria".

- id:
 - **Long.** The id that can be used to reference a Category.
- name:
 - **String.** The name of the Region.
- image_url:
 - **String. Optional.** The full url of an image, usually a skyline, that can be presented with the name.
- parent_id:
 - **Long. Optional.** If the region is inside of another region, such as Victoria being inside of Canada, the parent_id of the region will reference the id of the region that encapsulates it.

List Marketplace

- **Overview**
 - Get a list of the items in the marketplace

- **Parameters**

- Method: GET
- Location: /marketplace
- vendor: **Long. Optional**. Limits the returned marketplace gifts to those that have the vendor with given vendor id.
- region: **Long. Optional**. Limits the returned marketplace gifts to those that are in
- category: **Long. Optional**. Limits the returned marketplace gifts to those that have a category with the corresponding id.
- max_price_in_cents: **Integer. Optional**. Limits the returned marketplace gifts to those that can be sent for a value less than max_price_in_cents.
- min_price_in_cents: **Integer. Optional**. Limits the returned marketplace gifts to those that can be sent for a value more than min_price_in_cents.
- currencyisocode: **String. Optional**. Limits the response to only those gifts matching the provided currency. Current allowed values are CAD or USD
- limit: **Integer. Optional. Default 20**. The maximum number of marketplace gifts to return at once.
- offset: **Integer. Optional. Default 0**. The offset of the first marketplace gift in the total results.
- long_format: **Boolean. Optional. Default false**. Indicate if Vendor, Category and Region should be returned as part of the marketplace gift.

- **Response Attributes**

- **info**
 - INFO_MARKETPLACE_RETRIEVED - Marketplace Gifts Retrieved - A list of marketplace gifts has been retrieved.
- **marketplace_gifts**
 - **List<MarketplaceGift>**. The list of marketplace gifts in Giftbit's marketplace. The list is filtered by the supplied parameters.
- **limit**
 - **Integer**. The maximum number of marketplace gifts that can be returned.
- **offset**
 - **Integer**. The offset of the first marketplace gift in the total results.
- **total_count**
 - **Integer**. The total number of marketplace gifts that can be returned by the query.

List Vendors

- **Overview**

- Get a list of the vendors in the marketplace. The id of a vendor can be used to lookup gifts that are from that vendor.

- **Parameters**

- Method: GET
- Location: /campaign/vendors

- **Response Attributes**

- **info**

- INFO_MARKETPLACE_RETRIEVED_VENDORS - Marketplace Vendors Retrieved - A list of marketplace vendors has been retrieved.
- **vendors**
 - **List<Vendors>**. The list of Vendors in the Giftbit marketplace.

List Regions

- **Overview**
 - Get a list of the regions in the marketplace. The id of a region can be used to lookup gifts that are in that region.
- **Parameters**
 - Method: GET
 - Location: /campaign/regions
- **Response Attributes**
 - **info**
 - INFO_MARKETPLACE_RETRIEVED_REGIONS - Marketplace Regions Retrieved - A list of marketplace regions has been retrieved.
 - **regions**
 - **List<Regions>**. The list of marketplace regions in Giftbit's marketplace.

List Categories

- **Overview**
 - Get a list of the categories in the marketplace. The id of a category can be used to lookup gifts that are in that category.
- **Parameters**
 - Method: GET
 - Location: /campaign/categories
- **Response Attributes**
 - **info**
 - INFO_MARKETPLACE_RETRIEVED_CATEGORIES - Marketplace Categories Retrieved - A list of marketplace categories has been retrieved.
 - **categories**
 - **List<Category>**. The list of marketplace categories in Giftbit's marketplace.

Campaign:

Common Campaign Objects

Campaign. A Campaign represents a gift offer to one or more recipients. The campaign can include one marketplace gift, or multiple marketplace gifts that the each recipient can choose from. Included in that campaign object is a list of gifts, a list of recipients, the date that the gift offer expires, the status of the campaign, and the fees associated with the campaign. It is possible to create a campaign object as a quote that can be sent or deleted later.

A campaign can be sent to the api to create a new campaign (See the Create a Campaign section) and can

be returned as a response to an api call (See the List Campaigns and Get Campaign sections).

- id:
 - **String. Optional.** An identifier for the campaign assigned by the client. This identifier can be used to send or delete a quote. This identifier will always be returned in a response object if it has been supplied by the client.
- uuid
 - **String. Response Only.** An identifier for the campaign assigned by Giftbit. This identifier can be used to send or delete. This is only a response and should not be sent as part of a campaign object in any requests.
- gift_template:
 - **String. Optional.** The template id used for this campaign.
- delivery_type:
 - **String. Optional.** Determines if Giftbit will deliver the gift offers via email or create shortlinks for you to deliver yourself. Possible values:
 - GIFTBIT_EMAIL: Default if omitted. Must be used in conjunction with Contacts.
 - SHORTLINK: a shortlink will be created for each gift offer. May be used in conjunction with the link_count parameter or Contacts.
- link_count:
 - **Integer. Optional.** Only for use with delivery_type of SHORTLINK. Sets how many links to create. Cannot be used in conjunction with Contacts
- message
 - **String. Optional.** The message that will be included in the gift email that is sent to the recipients. (not present if a template was used and a message override was not provided in the request).
- subject
 - **String. Optional.** The subject of the email that is sent to the recipients. (not present if a template was used and a subject override was not provided in the request). Not relevant if using the SHORTLINK delivery_type.
- status
 - **String.** The status representing where campaign object is in its life cycle. Possible values:
 - QUOTE - The campaign is a quote and has not been sent to any recipients and no charges have occurred.
 - API_CREATING - The campaign is in the process of being created.
 - CAMPAIGN_CREATING @Deprecated
 - AWAITING_FUNDS - The campaign has not yet been delivered to the recipient(s) because of insufficient funds in your Giftbit account. Note you do NOT need to resend the campaign - as soon as you add funds to your account or your pending funds clear, the campaign will be automatically delivered to your recipients.
 - CAMPAIGN_CREATED - The campaign has been created and the emails are sent or in the process of sending.
- fees
 - **Fees.** The fees object as described below.
- expiry

- **Date. yyyy-MM-dd.** The date that gift offer expires.
- marketplace_gifts
 - **List<MarketplaceGiftReference>.** A list of references to Marketplace Gifts.
- contacts
 - **Contacts.** A list of the recipients of the campaign. Described in the contacts section below.
- contacts_success_count
 - **Integer. Optional.** The contacts success count is only returned when a campaign object is first created. It represents how many of the contacts provided were successfully added as recipients to the campaign.
- contacts_failure_count
 - **Integer. Optional.** The contacts failure count is only returned when a campaign object is first created. It represents how many of the contacts provided were not successfully added as recipients to the campaign. Contacts can fail to be added to the campaign due to poorly formed email addresses, if they have unsubscribed or reported spam from receiving gifts from your giver account, or if they are known to be a non-reachable email address.

Fees. The fees object, via its child list of cost_entries, represents the complete monetary cost associated with creating this campaign. This includes the required funds that will be used to pay for gifts as they are redeemed, and any additional fees that Giftbit charges that are associated with the campaign. Each type of cost is represented via an entry in the cost_entries list.

- subtotal_in_cents - DEPRECATED
 - **Integer. Cents.** The before tax cost of the campaign.
 - Cost of campaign without commission.
- tax_in_cents - DEPRECATED - consider the tax_in_cents in each cost_entries entry.
 - **Integer. Cents.** The tax for the campaign.
 - Tax for campaign commission.
- tax_type - DEPRECATED - consider the tax_type in each cost_entries entry.
 - **String.** The tax type for the campaign. Possible values include HST, GST, NOTAX, and GSTANDPST.
- total_in_cents - DEPRECATED - consider amount_in_cents in all cost_entries
 - **Integer. Cents.** The after tax cost of sending the campaign.
 - Cost of campaign with commission.
- cost_entries
 - **List<CostEntry>.** A list of cost entry objects described below.

CostEntry. A cost entry details a particular part of the monetary cost for this campaign. Note that fee types that do not apply to a particular campaign will not be included.

- amount_in_cents
 - **Integer.** The total amount, in cents, for this fee_type for the entire campaign. Zero for fee_type BREAKAGE_PERCENTAGE until the campaign expiry date has passed.
- percentage
 - **Float.** A value between 1-100 for fee_type BREAKAGE_PERCENTAGE or UPFRONT_COMMISSION_PERCENTAGE. Zero otherwise.
- tax_in_cents

- **Integer.** The total amount, in cents, for the tax on this fee_type. Zero for tax_type NOTAX, and for fee_type BREAKAGE_PERCENTAGE until the campaign expiry date has passed.
- tax_type
 - **String.** The tax type for this cost entry. Possible values include HST, GST, NOTAX, and GSTANDPST.
- number_of_gifts - DEPRECATED - cost_entries apply to all recipients
 - The number of recipients for the campaign that are included in this fee structure.
- fee_per_gift_in_cents - DEPRECATED - see amount_in_cents for UPFRONT_PER_GIFT_FEE fee_type
 - The fee per recipient associated with this campaign
- currency
 - **String.** The ISO 4217 three letter uppercase currency code of this cost entry, for example USD.
- fee_type
 - **String.** Will be one of:
 - INCLUDED_IN_SUBSCRIPTION (DEPRECATED)
 - OVER_SUBSCRIPTION_LIMIT (DEPRECATED)
 - UPFRONT_COMMISSION_PERCENTAGE
 - Fee type representing Giftbit's up front fee that is calculated as percentage of total gift value
 - BREAKAGE_PERCENTAGE
 - Fee type representing the percentage of the unclaimed gift value that is kept by Giftbit after the campaign's expiration date has been reached.
 - UPFRONT_PER_GIFT_FEE
 - Fee type representing a per-gift cost.
 - PREFUND_GIFT_COST
 - The amount of funds that are needed to cover gift redemptions.

Marketplace Gift Reference. A reference to a marketplace gift. For marketplace gifts that have multiple prices or a range of prices, this must contain a price. If a price is defined, it must either match a price on the gift or be within the price range of the gift.

- id
 - **Long.** A reference to the id of a Marketplace Gift.
- price_in_cents
 - Cents. Optional. The price of the gift. This is required if the gift has multiple price options or has a price range. The value must either match a value in the Marketplace Gift's prices_in_cents list or be within the range defined by the Marketplace Gift's max_price_in_cents and min_price_in_cents.

Contacts. The recipients of the campaign. There must be at least one recipient for the campaign to be valid.

- firstname
 - **String. Optional.** The first name associated with the contact. This name will be used to address recipients in gift emails.
- lastname
 - **String. Optional.** The last name associated with the contact. It is useful for differentiating

contacts in the Giftbit system, but is not used in the gift emails.

- email
 - **String.** The recipients email and a unique identifier for the recipient.
- added_successfully
 - **Boolean. Response Only. Optional.** If the contact was successfully added to the campaign. This will only be returned during campaign or quote creation.
- message
 - **String. Response Only. Optional.** The status message of adding the contact to the campaign. The message will be “New contact”, “Existing contact”, or an error message stating why the contact could not be added. This will only be returned during campaign or quote creation.

List Campaigns

- **Overview**
 - Returns a list of existing campaigns.
- **Parameters**
 - Method: GET
 - Location: /campaign
- **Response Attributes**
 - **info**
 - INFO_CAMPAIGN_RETRIEVED - Campaign Retrieved - A campaign or list of campaigns has been retrieved.
 - **campaigns**
 - A list of campaign objects that the client has created.

Get Campaign

- **Overview**
 - Get the details of a campaign by id you have assigned it or by the UUID associated with it.
- **Parameters**
 - Method: GET
 - Location: /campaign/<id|UUID>
- **Response Attributes**
 - **info**
 - INFO_CAMPAIGN_RETRIEVED - Campaign Retrieved - A campaign or list of campaigns has been retrieved.
 - **campaign**
 - The campaign object associated with the id or UUID.

Create Campaign

- **Overview**
 - Create a campaign or get a quote for a campaign.
- **Parameters**
 - Method: POST

- Location: /campaign
- json: a json Campaign object including
 - message (required if not using templates)
 - subject (required if not using templates)
 - gift_template (only if using templates)
 - delivery_type
 - link_count (only if delivery_type is SHORTLINK and not supplying contacts)
 - contacts (list)
 - firstname
 - lastname
 - email
 - marketplace_gifts (list)
 - id
 - price_in_cents (optional)
 - expiry (yyyy-MM-dd)
 - id
 - quote (optional)
- **Response Attributes**
 - **info (one of the following)**
 - INFO_CAMPAIGN_CREATED - Campaign Created - Campaign creation has begun successfully
 - INFO_CAMPAIGN_CREATED_FUNDS_REQUIRED - The campaign creation request was successfully received by Giftbit but there are insufficient funds in your Giftbit account to cover the cost of the campaign. Note you do NOT need to resend the campaign - as soon as you add funds to your account, the campaign will be automatically delivered to your recipients.
 - INFO_CAMPAIGN_CREATED_FUNDS_PENDING - The campaign creation request was successfully received by Giftbit but pending funds (such as a cheque or wire transfer that you sent) have not yet cleared. Note you do NOT need to resend the campaign - as soon as your pending funds clear and become available, the campaign will be automatically delivered to your recipients.
 -
 - **campaign**
 - The created campaign.

Delete Campaign Quote

- **Overview**
 - Delete a campaign quote by uuid or id
- **Parameters**
 - Method: DELETE
 - Location: /campaign/<id|UUID>
- **Response Attributes**
 - **info**

- INFO_CAMPAIGN_DELETED - Campaign Deleted - The campaign has been deleted.

Send a Quote

- **Overview**
 - Send a previously created quote by uuid or id.
- **Parameters**
 - Method: PUT
 - Location: /campaign/<id|UUID>
- **Response Attributes**
 - **info**
 - INFO_CAMPAIGN_CREATED - Campaign Created - Campaign creation has begun.
 - **campaign**
 - The created campaign. .

Funds:

Show Funds

- **Overview**
 - Retrieve a current funds.
- **Parameters**
 - Method: GET
 - Location: /funds
- **Response Attributes**
 - **info**
 - INFO_FUNDS - Fund information retrieved - Your current fund information was retrieved.
 - **fundsbycurrency**
 - **Map.** A map of currencies, with USD and CAD. Each currency contains:
 - **available_in_cents. Integer.** Value, in cents of available funds in your account. This is your purchasing power for new gift campaigns
 - **pending_in_cents. Integer.** Value, in cents, of funds for your account that have been sent by you but yet fully processed by Giftbit. These will move to available balance once processed.
 - **reserved_in_cents. Integer.** Value, in cents, of funds that are held to cover the cost of outstanding gift offers that have not yet been claimed and are still within the claim before date.

Links:

Retrieve Links

- **Overview**
 - Retrieve shortlinks
- **Parameters**
 - Method: GET
 - Location: /links/{
 - offset: where in the result set to start
 - limit: default is to return 50 links per request. Maximum is 100. For large link campaigns, use limit and offset to page through and save all the links
- **Response Attributes**
 - **info**
 - INFO_LINKS - Your gift offers are ready and the links are returned
 - INFO_LINKS_GENERATION_IN_PROGRESS - Giftbit is still preparing your gift offers. Try again in 5 minutes
 - **shortlinks**
 - **List.** A list of your links
 -
 - **number_of_results**
 - Integer, the number of links returned from this request.
 - **limit**
 - Integer, the maximum returned for this request
 - **offset**
 - Where in the result set this response is.
 - **total_count**
 - Integer. The total number of links in this campaign
 -

Gifts

Common Gifts Objects

Gift. A gift represents an individual gift offer to a recipient. A gift is created by Giftbit for every recipient of a campaign; as such gifts are response-only objects in the API (to send gift offers to your recipients, see the /campaign endpoint)

- **uuid**
 - **String.** A unique identifier for the gift assigned by Giftbit. This identifier can be used to cancel or resend the gift.
- **campaign_uuid**
 - **String.** An identifier for the campaign assigned by Giftbit. This identifier is returned from calls to the campaign endpoint. This identifier can be used to send or delete campaigns.
- **campaign_id**
 - **String. Nullable.** An identifier for the campaign assigned by the client during campaign creation. This identifier can be used to send or delete a quote.
- **price_in_cents**

- **String. Nullable.** The price of the gift in cents. For a gift of choice, this may be null if the recipient has not selected a gift.
- marketplacegift_id
 - **Long. Nullable.** The id of the marketplace gift. For a gift of choice, this may be null if the recipient has not selected a gift.
- recipient_name
 - **String. Nullable.** The name of the recipient. Not present if no recipient name was supplied.
- recipient_email
 - **Email. Nullable** The email address of the recipient and the address the gift offer was sent too. Not present for a shortlink Campaign that did not include contacts
- shortlink
 - **URI. Nullable** The fully qualified URI that should be given to the recipient for this gift.
- status
 - **String.** The status of the gift. Possible values:
 - SENT_AND_REDEEMABLE - The gift has been sent or is waiting to be sent.
 - REDEEMED - The recipient has redeemed the gift.
 - TO_CHARITY - The recipient has donated the gift to charity.
 - GIVER_CANCELLED - The gift has been canceled by the giver.
 - EXPIRED - The gift was not redeemed before the user before date and is no longer redeemable.
- delivery_status
 - **String.** The delivery status of the gift. Possible values:
 - UNSENT - The gift has not yet been sent.
 - DELIVERED - The gift has been sent, and no bounce or other problem notifications have been received. This is a success status
 - UNDELIVERABLE - There has been a fatal issue with delivering the gift. This is often caused by attempting to send a gift to an email address that does not exist. We suggest double checking the recipient address and changing or removing it for future campaigns.
 - TEMPORARILY_UNDELIVERABLE - There has been a temporary email deliverability issue to this recipient. This can be caused by issues like a full mailbox for their email address. While they may eventually get their gift, It is not guaranteed so we suggest following up with this recipient and resending their gift if necessary via the web interface or the /gifts API endpoint.
 - UNSUBSCRIBED - Your recipient unsubscribed from the unsubscribe link in this or a previous gift offer email. You should remove them from all future campaigns as Giftbit will return an error if you attempt to send to them again.
 - COMPLAINT - Your recipient marked the gift message as spam. You should remove them from all future campaigns as Giftbit will return an error if you attempt to send to them again.
 -
- created_date

- **Date. yyyy-MM-dd HH:mm:ss.** The date the gift was created. This may not match the campaign creation date as the gifts are created after a campaign is funded and approved.
- **delivery_date**
 - **Date. yyyy-MM-dd HH:mm:ss.** The date the gift was delivered. This may not match the `created_date` if the campaign was scheduled to be sent at a later date.
- **redelivery_count**
 - **Integer. Nullable.** The number of times a gift has been redelivered to a recipient. This may be null if the gift has not been redelivered.
- **redeemed_date**
 - **Date. yyyy-MM-dd HH:mm:ss. Nullable.** The date that a gift was redeemed. This will be null if the gift has not been redeemed.
- **management_dashboard_link**
 - **url.** The url of the gift that the sender can use to view the gift in Giftbit's dashboard. The sender must login to view this page.

List Gifts

- **Overview**

- Retrieve a list of gift offers you have sent through Giftbit.

- **Parameters**

- Method: GET
- Location: /gifts
- uuid: **String. Optional.** Return the gift with matching uuid.
- campaign_uuid: **String. Optional.** Filter gifts by the campaign with the matching uuid.
- campaign_id: **String. Optional.** Filter gifts by the campaign with the matching campaign id.
- price_in_cents<[_greater_than|_less_than]: **Integer. Optional.** Filter gift by their price in cents. Gifts from a gift of choice campaign may not have this value set.
price_in_cents_greater_than and/or price_in_cents_less_than can be used to filter gifts with prices in cents greater than or less than the supplied value respectively.
- marketplacegift_id: **Long. Optional.** Filter gifts by the marketplace gift id. Gifts from a gift of choice campaign may not have this value set.
- recipient_name: **String. Optional.** Filter gifts by the name, or part of the name of the recipient. The search string must be at least 3 characters long.
- recipient_email: **String. Optional.** Filter gifts by the email address, or part of the email of the recipient. The search string must be at least 3 characters long.
- delivery_status: **String. Optional.** Filter gifts by their delivery status. Delivery status can be one of UNSENT, DELIVERED, UNDELIVERABLE, TEMPORARILY_UNDELIVERABLE, UNSUBSCRIBED, COMPLAINT.
- status: **String. Optional.** Filter gifts by their status. Status can be one of SENT_AND_REDEEMABLE, REDEEMED, TO_CHARITY, GIVER_CANCELLED, or EXPIRED.
- created_date<[_greater_than|_less_than]: **Date. Optional.** Filter the gifts by their created date. Date format is yyyy-MM-dd HH:mm:ss. Use created_date_greater_than for dates that are greater than or equal to the given date. Use created_date_lesser_than for dates that are lesser than or equal to the given date.
- delivery_date<[_greater_than|_less_than]: **Date. Optional.** Filter the gifts by their delivery date. Use delivery_date_greater_than for dates that are greater than or equal to the given date. Use delivery_date_lesser_than for dates that are lesser than or equal to the given date.
- redelivery_count<[_greater_than|_less_than]: **Integer. Optional.** Filter gifts by their redelivery count. Use redelivery_count_greater_than for delivery counts greater than or equal to the given value. Use redelivery_count_less_than for delivery counts less than or equal to the given value.
- redeemed_date<[_greater_than|_less_than]: **Date. Optional.** Filter gifts by their redeemed date. Date format is yyyy-MM-dd HH:mm:ss. Use redeemed_date_greater_than for dates that are greater than or equal to the given date. Use redeemed_date_lesser_than for dates that are lesser than or equal to the given date.
- limit: **Integer. Optional. Default 20.** The maximum number of gifts to return at once.
- offset: **Integer. Optional. Default 0.** The offset of the first gift in the total results.

- sort: **String. Optional. Default created_date.** The field to sort results by. Must be one of campaign_id, price_in_cents, marketplacegift_id, recipient_name, recipient_email, delivery_status, or status.
- order: **String. Optional. Default desc.** The order to sort by. Must be one of asc or desc.
- **Response Attributes**
 - **info**
 - INFO_GIFTS - Gift information retrieved - Returned gifts matching your search parameters.
 - **gifts**
 - **List<Gift>**. The list of your gifts sent through Giftbit. The list is filtered by the supplied parameters.
 - **limit**
 - **Integer.** The maximum number of gifts that can be returned.
 - **offset**
 - **Integer.** The offset of the first gift in the total results.
 - **total_count**
 - **Integer.** The total number of gifts that can be returned by the query.

Resend Gift

- **Overview**
 - Resend a gift. Given the gift's uuid, resend the gift offer email to the recipients email address. The gift's uuid can be found through the /gifts endpoint.
- **Parameters**
 - Method: PUT
 - Location: /gifts/<UUID>
 - json: a json object with a resend parameter.
 - { 'resend':true }
- **Response Attributes**
 - **info**
 - INFO_GIFTS_RESENT - Gift Resent - The gift has been resent.
 - **gift**
 - **Gift.** The gift that was resent.

Cancel Gift

- **Overview**
 - Cancel a gift. Given the gift's uuid, cancel the gift offer. The gift's uuid can be found through the /gifts endpoint.
- **Parameters**
 - Method: DELETE
 - Location: /gifts/<UUID>
- **Response Attributes**
 - **info**

- INFO_GIFTS_CANCELED - Gift canceled - The gift has been canceled.
 - gift
 - Gift. The gift that was canceled.

Retrieve Gift

- Overview
 - Retrieve a gift. Given the gift's uuid, retrieve an object that represents the gift. The gift's uuid can be found through the /gifts endpoint.
- Parameters
 - Method: GET
 - Location: /gifts/<UUID>
- Response Attributes
 - info
 - INFO_GIFTS - Gift information retrieved - Returned gifts matching your search parameters.
 - gift
 - Gift. The gift that matches the uuid.

Example Calls

Credentials Check

Command:

- curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/papi/v1

Response:

```
{
  "username": "partner@giftbit.com",
  "displayname": "Integration Partner Inc.",
  "info": {
    "code": "INFO_CREDNTIALS",
    "name": "Credentials are valid",
    "message": "The credentials used are valid and login attempt was successful."
  }
}
```

List Marketplace

Command:

- curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/papi/v1/marketplace

Response:

```
{
  "marketplace_gifts": [
    {
      "id": 1,
```

```

    "name": "Gift Certificate for Ulla Restraunt",
    "description": "Description of Gift Certificate for Ulla Restraunt",
    "redemption_instructions": "redemptionInstruction for Gift Certificate for Ulla Restraunt",
    "terms_and_conditions": "Terms",
    "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/20/ulla.png",
    "tax": "GSTANDPST",
    "vendor": {
      "id": 6,
      "name": "Ulla Restraunt",
      "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/20/ulla.png"
    },
    "categories": [
      {
        "id": 1,
        "name": "Golf"
      }
    ],
    "regions": [
      {
        "id": 5,
        "name": "Victoria",
        "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/card-images/victoria.jpg",
        "parent_id": 4
      }
    ],
    "max_price_in_cents": 100000,
    "min_price_in_cents": 500
  },
  {
    "id": 2,
    "name": "Gift Certificate for Bliss Boutique",
    "description": "Description of Gift Certificate for Bliss Boutique",
    "redemption_instructions": "redemptionInstruction for Gift Certificate for Bliss Boutique",
    "terms_and_conditions": "Terms",
    "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/21/bliss.png",
    "tax": "GSTANDPST",
    "vendor": {
      "id": 7,
      "name": "Bliss Boutique",
      "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/21/bliss.png"
    },

```



```

    "categories":[
      {
        "id":3,
        "name":"Dinner"
      }
    ],
    "regions":[
      {
        "id":5,
        "name":"Victoria",
        "image_url":"https://s3-us-west-2.amazonaws.com/staticimageskiind/card-images/victoria.jpg",
        "parent_id":4
      }
    ],
    "max_price_in_cents":100000,
    "min_price_in_cents":500
  },
  {
    "id":3,
    "name":"Gift Certificate for Bliss Boutique",
    "description":"Description of Gift Certificate for Bliss Boutique",
    "redemption_instructions":"redemptionInstruction for Gift Certificate for Bliss Boutique",
    "terms_and_conditions":"Terms",
    "image_url":"https://s3-us-west-2.amazonaws.com/staticimageskiind/21/bliss.png",
    "tax":"GSTANDPST",
    "vendor":{
      "id":7,
      "name":"Bliss Boutique",
      "image_url":"https://s3-us-west-2.amazonaws.com/staticimageskiind/21/bliss.png"
    },
    "categories":[
      {
        "id":3,
        "name":"Dinner"
      }
    ],
    "regions":[
      {
        "id":5,
        "name":"Victoria",
        "image_url":"https://s3-us-west-2.amazonaws.com/staticimageskiind/card-images/victoria.jpg",
        "parent_id":4
      }
    ]
  }
}

```

```

    ],
    "prices_in_cents":[
        5000,
        10000,
        15000
    ]
},
"info":{
    "code":"INFO_MARKETPLACE_RETRIEVED",
    "name":"Marketplace Gifts Retrieved",
    "message":"A list of marketplace gifts has been retrieved."
}
}

```

Command:

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/papi/v1/marketplace?vendor=6`

Response:

```

{
    "marketplace_gifts":[
        {
            "id":1,
            "name":"Gift Certificate for Ulla Restraunt",
            "description":"Description of Gift Certificate for Ulla Restraunt",
            "redemption_instructions":"redemptionInstruction for Gift Certificate for Ulla Restraunt",
            "terms_and_conditions":"Terms",
            "image_url":"https://s3-us-west-2.amazonaws.com/staticimageskiind/20/ulla.png",
            "tax":"GSTANDPST",
            "vendor":{
                "id":6,
                "name":"Ulla Restraunt",
                "image_url":"https://s3-us-west-2.amazonaws.com/staticimageskiind/20/ulla.png"
            },
            "categories":[
                {
                    "id":1,
                    "name":"Golf"
                }
            ],
            "regions":[
                {
                    "id":5,
                    "name":"Victoria",

```

```

        "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/card-images/victoria.jpg",
        "parent_id": 4
    },
    ],
    "max_price_in_cents": 100000,
    "min_price_in_cents": 500
},
"info": {
    "code": "INFO_MARKETPLACE_RETRIEVED",
    "name": "Marketplace Gifts Retrieved",
    "message": "A list of marketplace gifts has been retrieved."
}
}

```

List Vendors

Command

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/papi/v1/marketplace/vendors`

Response:

```

{
    "vendors": [
        {
            "id": 6,
            "name": "Ulla Restraunt",
            "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/20/ulla.png"
        },
        {
            "id": 7,
            "name": "Bliss Boutique",
            "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/21/bliss.png"
        },
        {
            "id": 8,
            "name": "Bear Mountain",
            "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/22/bearmountain.png"
        },
        {
            "id": 9,
            "name": "Simon Says Roses",
            "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/23/flowerssimonrose.png"
        }
    ]
}

```

```

    },
    "info": {
      "code": "INFO_MARKETPLACE_RETRIEVED_VENDORS",
      "name": "Marketplace Vendors Retrieved",
      "message": "A list of marketplace vendors has been retrieved."
    }
  }
}

```

List Regions

Command:

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/papi/v1/marketplace/regions`

Response:

```

{
  "regions": [
    {
      "id": 1,
      "name": "North America"
    },
    {
      "id": 2,
      "name": "U.S.A",
      "parent_id": 1
    },
    {
      "id": 3,
      "name": "New England",
      "parent_id": 2
    },
    {
      "id": 4,
      "name": "Canada",
      "parent_id": 1
    },
    {
      "id": 5,
      "name": "Victoria",
      "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/card-images/victoria.jpg",
      "parent_id": 4
    },
    {
      "id": 6,
      "name": "Vancouver",
      "image_url": "https://s3-us-west-2.amazonaws.com/staticimageskiind/card-images/vancouver.jpg"
    }
  ]
}

```

```

        es/vancouver.jpg",
        "parent_id":4
    },
    ],
    "info":{
        "code":"INFO_MARKETPLACE_RETRIEVED_REGIONS",
        "name":"Marketplace Regions Retrieved",
        "message":"A list of marketplace regions has been retrieved."
    }
}

```

List Categories

Command:

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/papi/v1/marketplace/categories`

Response:

```

{
    "categories":[
        {
            "id":1,
            "name":"Golf"
        },
        {
            "id":2,
            "name":"Coffee"
        },
        {
            "id":3,
            "name":"Dinner"
        },
        {
            "id":4,
            "name":"Flowers"
        },
        {
            "id":5,
            "name":"Clothing"
        },
        {
            "id":6,
            "name":"Health & Wellness"
        },
        {
            "id":7,
            "name":"Art"
        },
    ],
}

```

```

    {
      "id":8,
      "name":"Amazon"
    }
  ],
  "info":{
    "code":"INFO_MARKETPLACE_RETRIEVED_CATEGORIES",
    "name":"Marketplace Categories Retrieved",
    "message":"A list of marketplace categories has been retrieved."
  }
}

```

Create Campaign Quote

Command:

- `curl -k -X POST -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -H "Content-Type: application/json" -d '{"message": "I am sorry that I shot you in the eye with a Nerf gun. Enjoy something from Amazon.com on me. I think I will send myself a gift at the same time, why not!", "subject": "Sorry", "contacts": [{"firstname": "Graham", "lastname": "Baradoy", "email": "graham@giftbit.com"}, {"firstname": "Peter", "lastname": "Locke", "email": "peter@giftbit.com"}], "marketplace_gifts": [{"id": "<FROM /marketplace QUERY>", "price_in_cents": 5000}], "expiry": "2015-12-20", "id": "GiftToGraham235", "quote": true}' https://api.giftbit.com/papi/v1/campaign`

Response:

```

{
  "info":{
    "code":"INFO_CAMPAIGN_QUOTE",
    "name":"Campaign Quote",
    "message":"A quote for the campaign can be found in the campaign fees object."
  },
  "campaign":{
    "message":"I am sorry that I shot you in the eye with a Nerf gun. Enjoy something from Amazon.com on me. I think I will send myself a gift at the same time, why not!",
    "subject":"Sorry",
    "contacts":[
      {
        "email":"graham@giftbit.com",
        "firstname":"Graham",
        "added_successfully":true,
        "lastname":"Baradoy",
        "message":"Existing contact"
      },
      {
        "email":"peter@giftbit.com",
        "firstname":"Peter",
        "added_successfully":true,
        "lastname":"Locke",

```

```

        "message": "New contact"
    },
    ],
    "marketplace_gifts": [
        {
            "id": 27,
            "price": 5000
        }
    ],
    "status": "QUOTE",
    "uuid": "35e430691445424bbd4932ebcaafdd0d",
    "expiry": "2015-12-20",
    "contacts_success_count": 2,
    "contacts_failure_count": 0,
    "fees": {
        "cost_entries": [
            {
                "percentage": 0,
                "fee_type": "PREFUND_GIFT_COST",
                "amount_in_cents": 10000,
                "currency": "USD",
                "tax_type": "NOTAX",
                "tax_in_cents": 0,
                "number_of_gifts": 2,
                "fee_per_gift_in_cents": 0
            },
            {
                "percentage": 50,
                "fee_type": "BREAKAGE_PERCENTAGE",
                "amount_in_cents": 0,
                "currency": "USD",
                "tax_type": "NOTAX",
                "tax_in_cents": 0,
                "number_of_gifts": 0,
                "fee_per_gift_in_cents": 0
            }
        ],
        "subtotal_in_cents": 10000,
        "tax_in_cents": 0,
        "tax_type": "NOTAX",
        "total_in_cents": 10000
    },
    "id": "GiftToGraham235"
}

```

Create Campaign Immediately

Command:

- `curl -k -X POST -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -H "Content-Type: application/json" -d '{"message": "Thanks for all of the hard work on the API this week. Have lunch at Bliss on me.", "subject": "Thanks For the Hard Work", "contacts": [{"firstname": "Graham", "lastname": "Baradoy", "email": "graham@giftbit.com"}], "marketplace_gifts": [{"id": "<FROM /marketplace QUERY>", "price_in_cents": 2000}], "expiry": "2015-12-20", "id": "GiftToGraham236"}' https://api.giftbit.com/papi/v1/campaign`

Response:

```
{
  "info": {
    "code": "INFO_CAMPAIGN_CREATED",
    "name": "Campaign Created",
    "message": "Campaign creation has begun."
  },
  "status": 200,
  "campaign": {
    "message": "Thanks for all of the hard work on the API this week. Have lunch at Bliss on me.",
    "subject": "Thanks For the Hard Work",
    "contacts": [
      {
        "email": "graham@giftbit.com",
        "firstname": "Graham",
        "added_successfully": true,
        "lastname": "Baradoy",
        "message": "Existing contact"
      }
    ],
    "marketplace_gifts": [
      {
        "id": 1,
        "price": 2000
      }
    ],
    "status": "API_CREATING",
    "uuid": "cc33649d6932403783614ad1a8f6632e",
    "expiry": "2015-12-20",
    "contacts_success_count": 1,
    "contacts_failure_count": 0,
    "fees": {
      "cost_entries": [
        {
          "percentage": 0,
          "fee_type": "PREFUND_GIFT_COST",
          "amount_in_cents": 2000,
          "currency": "USD",
          "tax_type": "NOTAX",

```



```

        "tax_in_cents":0,
        "number_of_gifts":1,
        "fee_per_gift_in_cents":2000
    },
    {
        "percentage":50.00,
        "fee_type":"BREAKAGE_PERCENTAGE",
        "amount_in_cents":0,
        "currency":"USD",
        "tax_type":"NOTAX",
        "tax_in_cents":0,
        "number_of_gifts":0,
        "fee_per_gift_in_cents":0
    }
],
"subtotal_in_cents":2000,
"tax_in_cents":0,
"tax_type":"NOTAX",
"total_in_cents":2000
},
"id":"GiftToGraham236"
}
}

```

List Campaigns

Command:

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/campaign`

Response:

```

{
  "info":{
    "code":"INFO_CAMPAIGN_RETRIEVED",
    "name":"Campaign Retrieved",
    "message":"A campaign or list of campaigns has been retrieved."
  },
  "campaigns":[
    {
      "message":"I am sorry that I shot you in the eye with a Nerf gun. Enjoy something from Amazon.com on me. I think I will send myself a gift at the same time, why not!",
      "subject":"Sorry",
      "contacts":[
        {
          "email":"graham@giftbit.com",
          "firstname":"Graham",
          "lastname":"Baradoy"
        }
      ]
    }
  ]
}

```

```

    },
    {
        "email": "peter@giftbit.com",
        "firstname": "Peter",
        "lastname": "Locke"
    }
],
"marketplace_gifts": [
    {
        "id": 27,
        "price": 5000
    }
],
"status": "QUOTE",
"uuid": "a0622d68d8604b03b1f0ca70ea119b96",
"expiry": "2015-12-20",
"fees": {
    "cost_entries": [
        {
            "percentage": 0,
            "fee_type": "PREFUND_GIFT_COST",
            "amount_in_cents": 10000,
            "currency": "USD",
            "tax_type": "NOTAX",
            "tax_in_cents": 0,
            "number_of_gifts": 2,
            "fee_per_gift_in_cents": 5000
        },
        {
            "percentage": 50.00,
            "fee_type": "BREAKAGE_PERCENTAGE",
            "amount_in_cents": 0,
            "currency": "USD",
            "tax_type": "NOTAX",
            "tax_in_cents": 0,
            "number_of_gifts": 2,
            "fee_per_gift_in_cents": 0
        }
    ],
    "subtotal_in_cents": 10000,
    "tax_in_cents": 0,
    "tax_type": "NOTAX",
    "total_in_cents": 10000
},
"id": "GiftToGraham235"
},
{
    "message": "Thanks for all of the hard work on the API this week. Have lunch at

```

```

Bliss on me.",
"subject":"Thanks For the Hard Work",
"contacts":[
    {
        "email":"graham@giftbit.com",
        "firstname":"Graham",
        "lastname":"Baradoy"
    }
],
"marketplace_gifts":[
    {
        "id":2,
        "price":2000
    }
],
"status":"API_CREATING",
"uuid":"b9a6b9ec29d8416689c3bbafd21e43a0",
"expiry":"2015-12-20",
"fees":{
    "cost_entries":[
        {
            "percentage":0,
            "fee_type":"PREFUND_GIFT_COST",
            "amount_in_cents":2000,
            "currency":"USD",
            "tax_type":"NOTAX",
            "tax_in_cents":0,
            "number_of_gifts":1,
            "fee_per_gift_in_cents":2000
        },
        {
            "percentage":50.00,
            "fee_type":"BREAKAGE_PERCENTAGE",
            "amount_in_cents":0,
            "currency":"USD",
            "tax_type":"NOTAX",
            "tax_in_cents":0,
            "number_of_gifts":1,
            "fee_per_gift_in_cents":0
        }
    ],
    "subtotal_in_cents":2000,
    "tax_in_cents":0,
    "tax_type":"NOTAX",
    "total_in_cents":2000
},
"id":"GiftToGraham236"
}

```

```
]
}
```

Get Campaign By UUID

Command:

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/campaign/45c6e59a5d334fcf822bf4921ad45fed`

Response:

```
{
  "info":{
    "code":"INFO_CAMPAIGN_RETRIEVED",
    "name":"Campaign Retrieved",
    "message":"A campaign or list of campaigns has been retrieved."
  },
  "campaign":{
    "message":"I am sorry that I shot you in the eye with a Nerf gun. Enjoy something from Amazon.com on me. I think I will send myself a gift at the same time, why not!",
    "subject":"Sorry",
    "contacts":[
      {
        "email":"graham@giftbit.com",
        "firstname":"Graham",
        "lastname":"Baradoy"
      },
      {
        "email":"peter@giftbit.com",
        "firstname":"Peter",
        "lastname":"Locke"
      }
    ],
    "marketplace_gifts":[
      {
        "id":26,
        "price":5000
      }
    ],
    "status":"QUOTE",
    "uuid":"bf56fdec0734fb18561632e021eb735",
    "expiry":"2015-12-20",
    "fees":{
      "cost_entries":[
        {
          "percentage":0,
          "fee_type":"PREFUND_GIFT_COST",
          "amount_in_cents":10000,

```

```

        "currency": "USD",
        "tax_type": "NOTAX",
        "tax_in_cents": 0,
        "number_of_gifts": 2,
        "fee_per_gift_in_cents": 5000
    },
    {
        "percentage": 50.00,
        "fee_type": "BREAKAGE_PERCENTAGE",
        "amount_in_cents": 0,
        "currency": "USD",
        "tax_type": "NOTAX",
        "tax_in_cents": 0,
        "number_of_gifts": 2,
        "fee_per_gift_in_cents": 0
    }
],
"subtotal_in_cents": 10000,
"tax_in_cents": 0,
"tax_type": "NOTAX",
"total_in_cents": 10000
},
"id": "GiftToGraham235"
}
}

```

Get Campaign By Id

Command:

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/campaign/GiftToGraham235`

Response:

(Same as get campaign by UUID)

Send a Quote

Command:

- `curl -X PUT -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/campaign/GiftToGraham235`

Response:

```

{
  "info": {
    "code": "INFO_CAMPAIGN_CREATED",
    "name": "Campaign Created",
    "message": "Campaign creation has begun."
  },

```

```

"status":200,
"campaign":{
  "message":"I am sorry that I shot you in the eye with a Nerf gun. Enjoy something
from Amazon.com on me. I think I will send myself a gift at the same time, why
not!",
  "subject":"Sorry",
  "contacts":[
    {
      "email":"graham@giftbit.com",
      "firstname":"Graham",
      "lastname":"Baradoy"
    },
    {
      "email":"peter@giftbit.com",
      "firstname":"Peter",
      "lastname":"Locke"
    }
  ],
  "marketplace_gifts":[
    {
      "id":27,
      "price":5000
    }
  ],
  "status":"API_CREATING",
  "uuid":"35e430691445424bbd4932ebcaafdd0d",
  "expiry":"2015-12-20",
  "fees":{
    "cost_entries":[
      {
        "percentage":0,
        "fee_type":"PREFUND_GIFT_COST",
        "amount_in_cents":10000,
        "currency":"USD",
        "tax_type":"NOTAX",
        "tax_in_cents":0,
        "number_of_gifts":2,
        "fee_per_gift_in_cents":0
      },
      {
        "percentage":50,
        "fee_type":"BREAKAGE_PERCENTAGE",
        "amount_in_cents":0,
        "currency":"USD",
        "tax_type":"NOTAX",
        "tax_in_cents":0,
        "number_of_gifts":2,
        "fee_per_gift_in_cents":0
      }
    ]
  }
}

```

```

    },
    "subtotal_in_cents":10000,
    "tax_in_cents":0,
    "tax_type":"NOTAX",
    "total_in_cents":10000
  },
  "id":"GiftToGraham235"
}
}

```

Delete A Quote

Command:

- `curl -X DELETE -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/campaign/SomeOtherCampaignQuote`

Response:

```

{
  "info":{
    "code":"INFO_CAMPAIGN_DELETED",
    "name":"Campaign Deleted",
    "message":"The campaign has been deleted."
  }
}

```

Find A Gift

- (see [definition](#) for full list of available query parameters)

Command:

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/gifts?created_date_greater_than=2014-12-10+14%3A40%3A33&created_date_less_than=2014-12-12+00%3A00%3A00`

Response:

```

{
  gifts:[
    {
      campaign_uuid: "5e1e9fa81306475f9e2142ec1049ab04",
      uuid: "cb52ff27126d42638dff37ab65fcc473",
      recipient_email: "tim@giftbit.com",
      delivery_status: "DELIVERED",
      status: "REDEEMED",
      management_dashboard_link:
        "https://app.giftbit.com/campaign/campaignInfo?uuid=5e1e9fa81306475f9e214

```

[2ec1049ab04&gift_uuid=cb52ff27126d42638dff37ab65fcc473",](#)

redelivery_count: 0,
recipient_name: "Tim",
price_in_cents: 5000,
marketplacegift_id: "2",
created_date: "2014-12-11 09:14:16",
delivery_date: "2014-12-11 09:14:26",
redeemed_date: "2014-12-11 09:54:26"

},
{

campaign_uuid: "5e1e9fa81306475f9e2142ec1049ab04",
uuid: "97043e9022df4b4c850696ec3ca09a82",
recipient_email: "graham@giftbit.com",
delivery_status: "DELIVERED",
status: "SENT_AND_REDEEMABLE",
management_dashboard_link:

["https://app.giftbit.com/campaign/campaignInfo?uuid=5e1e9fa81306475f9e2142ec1049ab04&gift_uuid=97043e9022df4b4c850696ec3ca09a82",](#)

redelivery_count: 0,
recipient_name: "Graham",
price_in_cents: 5000,
marketplacegift_id: "2",
created_date: "2014-12-11 09:14:16",
delivery_date: "2014-12-11 09:14:24"

}

],

limit: 20,

offset: 0,

total_count: 2,

info:{

code: "INFO_GIFTS",

name: "Gift information retrieved",

message: "Returned gifts matching your search parameters."

},

status: 200

}

Resend A Gift

Command:

- `curl -X PUT -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" -H "Content-Type: application/json" -d '{ "resend":true }' https://api.giftbit.com/api/v1/gifts/97043e9022df4b4c850696ec3ca09a82`

Response:

```
{
  gift:{
    uuid: "97043e9022df4b4c850696ec3ca09a82",
    campaign_uuid: "5e1e9fa81306475f9e2142ec1049ab04",
    recipient_email: "graham@giftbit.com",
    delivery_status: "DELIVERED",
    status: "SENT_AND_REDEEMABLE",
    management_dashboard_link:
      "https://app.giftbit.com/campaign/campaignInfo?uuid=5e1e9fa81306475f9e2142ec1049ab04&gift\_uuid=97043e9022df4b4c850696ec3ca09a82",
    redelivery_count: 1,
    recipient_name: "Graham",
    price_in_cents: 5000,
    marketplacegift_id: "2",
    created_date: "2014-12-11 09:14:16",
    delivery_date: "2014-12-11 09:14:24"
  },
  info:{
    code: "INFO_GIFTS_RESENT",
    name: "Gift resent",
    message: "The gift has been resent."
  },
  status: 200
}
```

Cancel A Gift

Command:

- `curl -X DELETE -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/gifts/97043e9022df4b4c850696ec3ca09a82`

Response:

```

{
  gift:{
    uuid: "97043e9022df4b4c850696ec3ca09a82",
    campaign_uuid: "5e1e9fa81306475f9e2142ec1049ab04",
    recipient_email: "graham@giftbit.com",
    delivery_status: "DELIVERED",
    status: "GIVER_CANCELLED",
    management_dashboard_link:
"https://app.giftbit.com/campaign/campaignInfo?uuid=5e1e9fa81306475f9e2142ec1049ab04&gift\_uuid=97043e9022df4b4c850696ec3ca09a82",
    redelivery_count: 1,
    recipient_name: "Graham",
    price_in_cents: 5000,
    marketplacegift_id: "2",
    created_date: "2014-12-11 09:14:16",
    delivery_date: "2014-12-11 09:14:24"
  },
  info:{
    code: "INFO_GIFTS_CANCELED",
    name: "Gift canceled",
    message: "The gift has been canceled."
  },
  status: 200
}

```

Show Funds

Command:

- `curl -X Get -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/funds`

Response:

```

{
  info:{
    code: "INFO_FUNDS",
    name: "Fund information retrieved",
    message: "Your current fund information was retrieved."
  },

```

```
fundsbycurrency:{
  USD:{
    available_in_cents: 500,
    pending_in_cents: 1000000,
    reserved_in_cents: 350500
  },
  CAD:{
    available_in_cents: 0,
    pending_in_cents: 0,
    reserved_in_cents: 0
  }
}
```

Retrieve Links, generation not complete

Command:

- `curl -X Get -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/links/myCampaignId`

Response:

```
{
  info:{
    code: "INFO_LINKS_GENERATION_IN_PROGRESS",
    name: "Links have not yet completed generating; request again in 5 minutes",
    message: "Link generation is not complete. Large campaigns will take some time to process and may be held for manual review; use the /campaigns endpoint for more detail on your campaign"
  }
}
```

Retrieve Links, generation complete

Command:

- `curl -X Get -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://api.giftbit.com/api/v1/links/myCampaignId`

Response:

```
{
  info:{
```

```

    code: "INFO_LINKS",
    name: "Link information retrieved",
    message: "Returned links for requested Campaign"
  },
  shortlinks: [
    {
      number: 1,
      shortlink: "http://gft.link/uniquecode1"
    },
    {
      number: 2,
      shortlink: "http://gft.link/uniquecode2"
    }
  ],
  number_of_results: 2,
  limit: 50,
  offset: 0,
  total_count: 2
}

```

Changelog

v1.5.1 - October 2015

- Documentation only update to reflect Giftbit's move to a free pricing and elimination of tiers. All API functionality is available to all users.

v1.5 - June 2015

- Added support for option of generating shortlinks for gift offers instead having Giftbit deliver them via email.

v1.4 - April 2015

- Added support for sending gifts via predefined templates created through the web interface.

v1.3.1 - March 2015

- Added the ability to filter the /marketplace endpoint results by gift currency by supplying 'currencyisocode' parameter.
- Updated documentation on the /gifts endpoint to reflect additional delivery_status values

v1.3.0 - January 2015

- Added /funds and /gifts endpoints

- /gifts endpoint is a major addition to the API that allows you to programmatically search and retrieve detailed information on the individual gifts that you have sent, and also resend or cancel them. Requires Corporate or higher subscription
- /funds endpoint allows you to see the current status of your account funding including available, reserved, and pending funds.

Full specifications and examples for the new endpoints are below in the document.

- **Bugfix:** optional user specified id was sometimes being enforced as required in campaign creation
- **Endpoints and branding:** updated api endpoints and general wording to reflect Giftbit branding. Note: kiind.me domain endpoints will continue to function for those already built in and will continue to receive the same functionality updates as the Giftbit endpoints.

v1.2.0 - July 2014 - improved response for campaign creation when account does not have enough funds to cover the cost of the campaign

- The 'info' object, prior to this update, always returned a code of 'INFO_CAMPAIGN_CREATED' even if the Giftbit account had insufficient funds. It now returns one of three possible values:
 - INFO_CAMPAIGN_CREATED - account is sufficiently funded and is being delivered (no change).
 - INFO_CAMPAIGN_CREATED_FUNDS_REQUIRED - account needs additional funds before the campaign can be delivered.
 - INFO_CAMPAIGN_CREATED_FUNDS_PENDING - we've received funds or notifications of funds (wire transfer or cheque) but they have not yet cleared.

Important Note: in all cases, it is still a status 200 successful response and you do not need to resend the create campaign API call. As soon as appropriate funds are available in your account, the campaign will automatically be delivered.

v1.1.0 - May 2014 - support for pre-funded campaign cost and status response

- Changes to potential values of 'status' field value of the 'campaign' object:
 - AWAITING_FUNDS now a possible status value.
 - CAMPAIGN_CREATING is deprecated to avoid confusion with 'API_CREATING'.
- The 'fees' object and its child 'cost_entries' list, which details fees for the given campaign, have a number of changes given the new campaign funding model.
 - New fields added to CostEntry object to reflect deprecation of top level 'fees' fields:
 - amount_in_cents, percentage, currency, tax_in_cents, tax_type
 - New possible values for fee_type in cost_entries list item
 - UPFRONT_COMMISSION_PERCENTAGE, BREAKAGE_PERCENTAGE, UPFRONT_PER_GIFT_FEE, PREFUND_GIFT_COST
 - All top level attributes of 'fees' object are deprecated and will be removed with the next major version update. Individual cost_entries should be considered instead. Deprecated fields of 'fees':
 - subtotal_in_cents
 - tax_in_cents
 - tax_type

- total_in_cents

- Documentation change to reflect more common use case in the example queries section

v1.0.0 - Initial release

Versioning Strategy

MajorVersion.MinorVersion.PointRelease

MajorVersion

The Major Version will be incremented in the case of major changes which deeply impact backwards compatibility, or other API architecture changes provided using a new /vX/ version set of end-points. Clients will need to migrate to the new /vX/ endpoint to use the functionality introduced in the major version.

MinorVersion

The Minor Version will be incremented when new element, attribute or endpoints are added that have little to no impact on backwards compatibility and should not require clients to change their code unless they want to make use of the newly introduced functionality. Changes introduced in a minor version update will be applied automatically to the Major /vX/ endpoint.

PointRelease

The Point Release will be incremented for trivial changes such as bug fixes and document changes/clarifications that will not structurally affect the API request or response format.

