

Kiind RESTful Partners API

Document Version 1.2.0 - July 2014

Changelog

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 Kiind 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

Overview

API Base URL

Production:

<https://www.kiind.me/papi/v1>

Test:

<https://testbed.kiind.me/papi/v1>

Summary of Resource Endpoints

/marketplace

/marketplace/{id}

/marketplace/regions

/marketplace/vendors

/marketplace/categories

/campaign

/campaign/{uuid}

/campaign/{id}

Example Calls

A section of example calls and responses can be found at the end of this document. Things in <BRACKETS> in the examples need to be replaced by your appropriate values. The example responses are for the purpose of showing response format; you should never use or hardcode any values in the example responses (such as gift ids and image URLs). Use the actual response values that come back for the given environment (Testbed or Production)

Authentication

The Kiind API uses token based authentication. Your secret token will be provided to you by Kiind (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 Kiind'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.

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.

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 Kiind. You will need to contact Kiind 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://www.kiind.me/papi/v1/>

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

Gift browsing criteria and getting the Id of a gift to send

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

Usecase 1: sending the same 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 Kiind's gift marketplace, but rather retrieve the Ids of the gifts you want manually, choose a valid and desired gift amount, and store them in your database or configuration.

GET <https://www.kiind.me/papi/v1/marketplace?limit=1000>

This will list all the gifts, and you can manually pick out the one(s) you want. Note: the Id values returned will differ between test and production environments, but will not change within an environment.

Usecase 2: providing an interactive marketplace

If you are providing a representation of Kiind'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://www.kiind.me/papi/v1/marketplace/regions>

GET - <https://www.kiind.me/papi/v1/marketplace/categories>

GET - <https://www.kiind.me/papi/v1/marketplace/vendors>

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

GET -

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

Both usescases: sending a gift

Once you have the Id(s) and values of the gift(s) you want to send, POST a JSON campaign object to the campaign endpoint. You can include more than one gift if you wish to allow the recipients to choose from one of them. Include one or more recipients to send the gifts too. You can optionally set some unique id object to reference the campaign by.

POST - <https://www.kiind.me/papi/v1/campaign> - {"subject": "This is the subject of the gift email", "message": "This is the greeting of the gift email.", "contacts": [{"firstname": "Graham", "lastname": "Baradoy", "email": "graham@kiind.me"}], "marketplace_gifts": [{"id": 230, "price_in_cents": 340}], "expiry": "2015-12-20", "id": "someGiftId"}

Important: including more than one gift gives each recipient a choice of the gifts, and they must choose one and only one. It does not give them multiple gifts.

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://www.kiind.me/papi/v1/campaign/someGiftId>

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 Kiind'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.
- **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 Kiind'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.
 - 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 Kiind’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 Kiind 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 Kiind'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 Kiind'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 Kiind. 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.
- **message**
 - **String.** A message that will be included in the gift email that is sent to the recipients.
- **subject**
 - **String.** The subject of the email that is sent to the recipients.
- **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 Kiind 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. The prices chosen for the gifts are often included in the list.
- **contacts**
 - **Contacts.** A list that 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 where 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 where 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 unsubscribe from receiving gifts from the client, 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 Kiind 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 Kiind's up front fee that is calculated as percentage of total gift value
 - BREAKAGE_PERCENTAGE (DEPRECATED)
 - Fee type representing the percentage of the unclaimed gift value that is kept by Kiind 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 Kiind 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
 - subject
 - 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 Kiind but there are insufficient funds in your Kiind 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 Kiind 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. .

Example Calls

Credentials Check

Command:

- `curl -X GET -H "Accept: application/json" -H "Authorization: Bearer <TOKEN>" https://www.kiind.me/papi/v1`

Response:

```
{
  "username": "partner@kiind.me",
  "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://www.kiind.me/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/staticimageskii
nd/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/b
liss.png"
    },
    "categories":[
        {
            "id":3,
            "name":"Dinner"
        }
    ],
    "regions":[
        {
            "id":5,
            "name":"Victoria",
            "image_url":"https://s3-us-west-2.amazonaws.com/staticimageskii
nd/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://www.kiind.mepapi/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://www.kiind.mepapi/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://www.kiind.mepapi/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",
    "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://www.kiind.mepapi/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@kiind.me"}, {"firstname": "Peter", "lastname": "Locke", "email": "peter@kiind.me"}], "marketplace_gifts": [{"id": "<FROM /marketplace QUERY>", "price_in_cents": 5000}], "expiry": "2015-12-20", "id": "GiftToGraham235", "quote": true}' <https://www.kiind.me/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@kiind.me",
        "firstname": "Graham",
        "added_successfully": true,
        "lastname": "Baradoy",
        "message": "Existing contact"
      },
      {
        "email": "peter@kiind.me",
        "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@kiind.me"}], "marketplace_gifts": [{"id": "<FROM /marketplace QUERY>", "price_in_cents": 2000}], "expiry": "2015-12-20", "id": "GiftToGraham236"}' https://www.kiind.me/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@kiind.me",
 "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://www.kiind.me/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@kiind.me",
 "firstname":"Graham",
 "lastname":"Baradoy"
 },
 {
 "email":"peter@kiind.me",
 "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@kiind.me",
 "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://www.kiind.me/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@kiind.me",
 "firstname": "Graham",
 "lastname": "Baradoy"
 },
 {
 "email": "peter@kiind.me",
 "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://www.kiind.me/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://www.kiind.me/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@kiind.me",
 "firstname": "Graham",
 "lastname": "Baradoy"
 },
 {
 "email": "peter@kiind.me",
 "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://www.kiind.me/api/v1/campaign/SomeOtherCampaignQuote`

### Response:

```

{
 "info":{
 "code":"INFO_CAMPAIGN_DELETED",

```

```
"name": "Campaign Deleted",
"message": "The campaign has been deleted."
```

```
}
```

```
}
```