

Fundraising Page API Document (JUL 2022)

Contents

Fundraising Page Creation	3
Overview	3
Minimum Requirements	3
Response Codes	3
201 – Created	3
202 – Accepted	4
409 – Conflict	4
Synchronous / Asynchronous Request	4
Beneficiary Selection.....	5
Beneficiary Selection Drop In UI	5
Creating your own Beneficiary Search Display	7
Deduplication	7
Accessibility.....	8
Page Type	9
Individuals	9
Teams	9
Team Members	10
Tagging	10
Custom Data.....	10
Page Configuration Settings.....	11
Web Tag	11
Page Title.....	12
Creator Name.....	12
Image Path	12
Targets	12
Notes	12
Display Settings	12
Page Expiry.....	12
Fundraising Enabled.....	13
User Profile	13
Example Response	13
Fundraising Page Create Status	15

Overview	15
Parameters.....	15
Response	15
Fundraising Page Search	16
Overview	16
Paging.....	16
Parameters.....	16
Response	17
Fundraising Page Retrieve	19
Overview	19
Parameters.....	19
Response	19
API Spec.....	20

Fundraising Page Creation

Overview

The Fundraising Page creation endpoint allows clients to create a fundraising page by sending through a request body with specified values. Required fields for this response include a valid `UserAccountId`, `BeneficiaryAccountId` and `EventCampaignId`. Pages can be created for new users by specifying `EmailAddress`, `FirstName`, `LastName` in replacement of `UserAccountId` so that a new account can be created on the GoFundraise system. Please note that if an existing account is found on the platform under the given `EmailAddress`, the page will be created under that account.

This endpoint requires Authorization via a Bearer Token

Url: <https://api.gofundraise.com.au/v1/pages>

Method: POST

Minimum Requirements

A page can be created with the following minimal settings:

```
{
  "EmailAddress": "string",
  "FirstName": "string",
  "LastName": "string",
  "BeneficiaryAccountId": "string",
  "EventCampaignId": "string"
}
```

For values not provided, the following default settings will be applied:

```
WebTag: Firstname Lastname
PageTitle: Firstname Lastname
CreatorName: Firstname Lastname
Message: Event Default
PageTypeId: S
ImagePath: Event Default
RaiseTarget: Event Default
MinimumRaiseTarget: Event Default
TeamTarget: Event Default
DisplayDonations: true
DisplayDonationAmount: true
DisplayFitnessData: true
DateOfBirth: NULL
ExpireDate: +1 Year
WaitForCompletion: True
```

All other values will be NULL

Response Codes

201 – Created

```
{
  "Id": "Integer",
  "Url": "String",
  "CreatorName": "String",
  "UserAccountId": "Integer",
```

```

    "BeneficiaryAccountId": "Integer",
    "EventCampaignId": "Integer",
    "PageTitle": "String",
    "ImagePath": "String",
    "Message": "String",
    "PageType": "Char",
    "TeamId": "Integer",
    "TeamTarget": "Decimal",
    "RaiseTarget": "Decimal",
    "MinimumRaiseTarget": "Decimal",
    "Total": "Decimal",
    "Notes": "String",
    "DisplayDonations": Boolean,
    "DisplayDonationAmount": Boolean,
    "DisplayFitnessData": Boolean,
    "PageTags": [
      {
        "TagId": Integer,
        "TagName": "String",
        "TagType": "String"
      }
    ],
    "Status": "Char",
    "CreateDate": "DateTime",
    "ExpireDate": "DateTime"
  }
}

```

202 – Accepted

```

{
  "requestId": Integer
}

```

409 – Conflict

```

{
  "Code": Integer,
  "PageId": "Integer",
  "Message": "String"
}

```

Synchronous / Asynchronous Request

When creating a fundraising page you can elect for the fundraising page to be created completely whilst your platform waits for the response. This is currently the default behaviour (`"WaitForCompletion" : true`). The mode will be set to `False` if no beneficiary id (`BeneficiaryAccountId`) is provided in the request.

The success response (201 Created) from a synchronous request will contain the entire fundraising page object that was created.

Alternatively, you can elect to have the page created asynchronously ("WaitForCompletion" : false).

The success response (202 Accepted) from an asynchronous request will contain the request id which can be requested in the future to retrieve details about the current creation progress of the page (CreateStatus).

Asynchronous requests should be used if

- there is a high volume of Fundraising pages that need to be created
- your app is sensitive to blocking and faster response times are required
- You do not know the beneficiary id (BeneficiaryAccountId) of the fundraising page prior to the page creation

Beneficiary Selection

In order to create a fundraising page, using the synchronous workflow, a beneficiary id (BeneficiaryAccountId) must be included in the request. If your event allows the user the ability to select from **multiple beneficiaries**, we suggest the best method is to use a **drop in UI component** which automatically connects to the GoFundraise event to provide seamless tiering and participating beneficiary updates. This is the same component we use on our UI so functionality / look and feel etc will remain identical between platforms and not require maintenance from your team for most upgrades.

Beneficiary Selection Drop In UI

Demo: <https://developer.gfnetworks.com/stable/gf-beneficiary-search/demo.html>

(You can inspect the source of the demo and copy that into your own HTML page to see it operating in your own Environment)

Development Instructions: Inject the following into the HEAD of the page you wish to load the widget on

```
<script src="https://unpkg.com/vue" charset="utf-8"></script>
<script
src="https://cdn.gofundraise.com.au/all/js/prod/GfBeneficiarySearch/
GfBeneficiarySearch.umd.min.js" charset="utf-8"></script>

<link rel="stylesheet"
href="https://cdn.gofundraise.com.au/all/js/prod/GfBeneficiarySearch/
GfBeneficiarySearch.css">
```

Inject the following into the body section. Place it where you want the search widget to appear

```
<div id="app">
  <gf-beneficiary-search
    event-campaign-id="10"
    region="au"
    api-domain="https://api.gofundraise.com"
```

```

        after-select-call-back="selectCallBack"
        after-reset-call-back="resetCallBack">
    </gf-beneficiary-search>
</div>

```

NOTE: If you want to preselect a beneficiary and you know the beneficiary id (BeneficiaryAccountId), you can pass that through on the `:beneficiary-id` property on the component as shown below (note the *colon* before the property)

```

<div id="app">
  <gf-beneficiary-search
    event-campaign-id="10"
    :beneficiary-id="104"
    region="au"
    api-domain="https://api.gofundraise.com"
    after-select-call-back="selectCallBack"
    after-reset-call-back="resetCallBack">
  </gf-beneficiary-search>
</div>

```

You can also ignore event tiering by using the following prop
`:ignore-event-tiering="true"`

And you can add additional tags to filter on by using the following prop
`tag-ids-as-csv="28,29,30"`

Add the script block in the body section that will trigger the loading of the component

```

<script type="text/javascript">
  Vue.component('gf-beneficiary-search', window.GfBeneficiarySearch);
  new Vue({ el: '#app' });
</script>

```

There are 2 callbacks from the component that should be handled

after-select-call-back

This gets called with a beneficiary object that is selected containing the following data:
 beneficiaryId, beneficiaryImagePath, beneficiaryName

```

beneficiary selected:
  Object
    beneficiaryId: "812781"
    beneficiaryImagePath: "https://cdn.g
    beneficiaryName: "0 Tolerance Inc"
    __proto__: Object

```

after-reset-call-back

This gets called when the user deselects a beneficiary. It will send an empty object on the callback.

The callbacks have to be in the global namespace (i.e. can be reached on the `window` object in the JS DOM). To add the callbacks you can add the following code example on the page in the body section

```
<script>
    // beneficiary selected success
    window.selectCallBack = function (data) {
        console.log("beneficiary selected: ");
        console.log(data);
    }
    // beneficiary reset
    window.resetCallBack = function (data) {
        console.log("beneficiary unselected");
    }
</script>
```

Creating your own Beneficiary Search Display

If you prefer to create your own custom UI display, you can do so by utilising the beneficiary search API:

<https://api.gofundraise.com/v1/beneficiaries/search?eventcampaignid=8474>

Deduplication

In order to prevent duplication of pages on the GoFundraise platform, we recommend passing a unique key through during the page creation call. GoFundraise will deduplicate on this key per event.

A successful page creation where there is NO duplicate fundraising page will return a 201 response. A page creation request that triggers the deduplication will return a 409 Conflict response with the duplicate entity id in the response. The "PageId" is the id of the Fundraising page id the has been created on our system under the deduplication key. If you need to pull more information about the page in addition to its id you will need to make another call to the pages search endpoint.

```
{
  "Code": "36001",
  "PageId": "938748",
  "Message": "The fundraising page you are trying to create already exists."
}
```

Deduplication does not take into account page type. If you pass in the same Id for a Team page and an individual page, it will trigger the deduplication and only the initial request will be honoured. If you use the same id's for these, add a pre/post fix to the id to ensure the team and individual pages remain unique.

To add a deduplication field when creating the event, add the following field to your request:
OriginEntryKey

A minimal request including this key would be:

```
{
  "EmailAddress": "string",
  "FirstName": "string",
  "LastName": "string",
  "BeneficiaryAccountId": "string",
  "EventCampaignId": "string",
  "OriginEntryKey" : "string"
}
```

Accessibility

It is possible to add certain settings to the page to decrease the likelihood that a page can be found online without direct linking to the page. This can be done through a String Array when creating the page.

```
"AccessibilityOptions": [ ],
```

There are currently 3 options that can be supplied here:

- NotSearchableOnSite
This prevents a Fundraising page from being populated in GoFundraise search requests and public search API endpoint.
- NotSearchableBySearchEngine
This adds a noindex meta tag to the head of the page to ask search engines not to index the page. Please note: search engines SHOULD adhere to this, however, we do not have control of how external search engines index pages outside of this standard.
- NotVisibleOnSite
This will prevent the page from loading through our site. It does not prevent the page from being accessible through the API or leaderboard. This should be used in conjunction with other accessibility tags.

These options can be used individual or together. A minimal example with both options passed through would be:

```
{
  "EmailAddress": "string",
  "FirstName": "string",
  "LastName": "string",
  "BeneficiaryAccountId": "string",
  "EventCampaignId": "string",
  "AccessibilityOptions": [
    "NotSearchableOnSite",
    "NotSearchableBySearchEngine"
  ]
}
```



```
]
}
```

Page Type

There are 2 Page types available to use, however, Individual Pages can be nested underneath Team pages to create a team member relationship.

These page types are:

Individual – S

Team – T

Individuals

A request will default to setting the **PageTypeId** as 'S', however, it is recommended that you have this set in your platform to ensure consistency when also using teams.

A minimal request when explicitly specifying an individual page would be:

```
{
  "EmailAddress": "string",
  "FirstName": "string",
  "LastName": "string",
  "BeneficiaryAccountId": "string",
  "EventCampaignId": "string",
  "PageTypeId": "S"
}
```

Teams

To create a team page, specify the **PageTypeId** as 'T'. A Team must be created prior to its team members. The id of the Team Page returned in the success response can be used by team members to nest under the team. It is not required that a team have any team members.

When creating a team, the `PageTitle` should be used to set the Team Name

If team members have been created prior to the team page, the currently must login to their GoFundraise account and manually join the team from their edit page screen. This cannot be done through the API at present.

Team Configuration Settings

In the circumstance of a team page, the `Target` field is specific to donations made directly to the team page and not team members. The `TeamTarget` field is an additional field which should be used to set the overall Team and Team Members fundraising target.

A minimal request when specifying a Team Page should be:

```
{
  "EmailAddress": "string",
  "FirstName": "string",
  "LastName": "string",
  "BeneficiaryAccountId": "string",
  "EventCampaignId": "string",
  "PageTypeId": "T",
  "PageTitle": "string",
  "TeamTarget": 0
}
```

Team Members

Team members are individual pages which have a team Id (`TeamId`) set to that of a team page. The `TeamId` can be found either through the id captured during team creation, or through the pages search api.

A minimal request when creating a Team Member would be:

```
{
  "EmailAddress": "string",
  "FirstName": "string",
  "LastName": "string",
  "BeneficiaryAccountId": "string",
  "EventCampaignId": "string",
  "PageTypeId": "S",
  "TeamId": "string"
}
```

Tagging

Allow the ability to search, group, and report on Fundraising pages simply by allowing tags to be added to them. For example, pages participating in a particular race or activity rather than grouping solely by the event.

Tags can be both pre-set or created on the fly. Any tags created on the fly will inherit default details which can be modified by the event owner on the GoFundraise platform after they have been created.

Applying tags to a Fundraising page can be done through a string array:

```
"PageTags": [ ]
```

A minimal example of this would be:

```
{
  "EmailAddress": "string",
  "FirstName": "string",
  "LastName": "string",
  "BeneficiaryAccountId": "string",
  "EventCampaignId": "string",
  "PageTags": [
    "2020 ROUND 2: Tom Price",
    "2020 ROUND 3: Wickham"
  ]
}
```

Tags are expected to be extensively utilised for better reporting and user functionality and as such should be considered for any page creation API request.

Custom Data

To help our clients, relate data within reporting, we have added some fields which can be displayed on the fundraising report alongside other fundraising specific data. The custom data option is expected to be expanded in the future to provide additional functionality by utilising other Name and Source handler types.

In addition to reporting, this data can be used in **Campaign Monitor, Autopilot, and Salesforce sync** connectors from GoFundraise.

Currently the only Source handler type available is "Static". Static data a string with a sample use case being the ability to capture registration data from an external form and associate with a Fundraising Page.

The Name field can be any string, however, only the following fields are currently available through reports, or the Salesforce connector:

```
CustomFormData-1. CustomFormData-2. CustomFormData-3,  
CustomFormData-4, CustomFormData-5
```

It is suggested to only use the above fields by default, if you require additional fields in excess of these, please contact GoFundraise Customer Care and we can discuss methods to utilise additional fields.

The **custom data** is sent in the format of

```
"CustomData": [  
  {  
    "Name": "CustomFormData-1",  
    "SourceType": "Static",  
    "Source": "String"  
  },  
  {  
    "Name": "CustomFormData-5",  
    "SourceType": "Static",  
    "Source": "String"  
  }  
]
```

A minimal example would be

```
{  
  "EmailAddress": "string",  
  "FirstName": "string",  
  "LastName": "string",  
  "BeneficiaryAccountId": "string",  
  "EventCampaignId": "string",  
  "CustomData": [  
    {  
      "Name": "CustomFormData-1",  
      "SourceType": "Static",  
      "Source": "String"  
    }  
  ]  
}
```

Page Configuration Settings

Web Tag

The **WebTag** field will set the URL of the Fundraising page. If this URL is already in use, this field will automatically add additional data to the end of the webtag. Please utilise the response to save the final webtag of the page if required.

```
"WebTag": "string"
```

Page Title

The `PageTitle` field sets the Fundraising page title which is displayed on the Fundraising page, in the Browser window, and on leaderboards

```
"PageTitle": "string"
```

Creator Name

The `CreatorName` field allows the user to utilise a pseudonym to publicly identify themselves as the page creator.

```
"CreatorName": "string"
```

Image Path

The image path field allows for a publicly available image to be downloaded and utilised as the fundraising page image. This image only is required to be publicly available temporarily during page creation for our system to download it. Pages created through an async process (no `Beneficiary Id` OR `WaitForCompletion=False`) currently cannot be passed through an image path and it will be ignored.

```
"ImagePath": "string"
```

Targets

The `RaiseTarget` and `MinimumRaiseTarget` fields allow you to set goals for the user to reach during fundraising. A minimum raise target is unable to be altered by the user, however, the raise target may be altered by the fundraiser.

```
"RaiseTarget": 0  
"MinimumRaiseTarget": 0
```

Notes

The `notes` field allow you to enter an ASCII compatible string which is not available publicly but is available through reporting.

```
"Notes": "string"
```

Display Settings

The following settings allow you to display or hide data from a public facing fundraising page rendered on the GoFundraise platform. This data is still available through the API.

```
"DisplayDonations": true,  
"DisplayDonationAmount": true,  
"DisplayFitnessData": true,
```

Page Expiry

The page expiry date (`ExpireDate`) should be set to coincide with completed fundraising of an event. Please note: in general it is recommended to allow fundraising to continue beyond the event date to raise more funds. By default the `ExpireDate` is set to 1 Year from the page creation date.

```
"ExpireDate": "2020-08-12T05:58:29.440Z"
```

Fundraising Enabled

The fundraising enabled (`FundraisingEnabled`) field allows pages to be created for reporting purposes where the user has decided not to participate in fundraising. This is also particularly useful for creating Team Pages where the team owner has declined to fundraise but team members may still choose to fundraise. It is expected that this tag would be used in conjunction with the `NotVisibleOnSite`, `NotSearchableOnSite` accessibility flags for individual pages.

```
"FundraisingEnabled": true
```

User Profile

The following fields are available to populate a portion of the user profile which will be attached to the Fundraising Page created. The user profile is deduplicated based on the Email address field unless a user id (`UserAccountId`) is specifically provided.

```
"UserAccountId": "string",  
"EmailAddress": "string",  
"FirstName": "string",  
"LastName": "string",  
"DateOfBirth": "YYYY-MM-DD",  
"MobileNumber": "string",
```

If multiple registrants are expected to use the same email address, for secondary registrations you should not pass through the `DateOfBirth` or `MobileNumber` fields as these will override the primary registration user.

A minimal example would be

```
{  
  "EmailAddress": "string",  
  "FirstName": "string",  
  "LastName": "string",  
  "DateOfBirth": "YYYY-MM-DD",  
  "MobileNumber": "string",  
  "BeneficiaryAccountId": "string",  
  "EventCampaignId": "string"  
}
```

Example Response

```
{  
  "Id": "938749",  
  "Url": "https://APITestEvent01.au.awstest.gfrnetworks.com/page/stringstring-99496131",  
  "CreatorName": "string string",  
  "UserAccountId": "1114786",
```

```
"BeneficiaryAccountId": "104",
"EventCampaignId": "6039",
"PageTitle": "string string",
"ImagePath": "https://APITestEvent01.au.awstest.gfnetworks.com/Upload/Fundraising/6039/Page131526722589918664.jpg",
"Message": "<p>Hi there!</p>\n\n<p>&nbsp;</p>\n\n<p>I'm making a difference by fundraising for a cause that's close to our hearts. Please support my efforts by making a secure online donation and by posting a message of support.</p>\n\n<p>&nbsp;</p>\n\n<p>Donations made through this platform are secure and will be remitted directly through to my charity of choice. Thanks so much for your support!</p>\n\n<p>&nbsp;</p>",
"PageType": "S",
"RaiseTarget": "700.0",
"MinimumRaiseTarget": "0.0",
"Total": "0.0",
"DisplayDonations": true,
"DisplayDonationAmount": true,
"DisplayFitnessData": true,
"Status": "A",
"CreateDate": "2020-09-18T03:24:56.131498Z",
"ExpireDate": "2021-09-18T03:24:56.131498Z"
}
```

Fundraising Page Create Status

Overview

The fundraising page create status is used to retrieve the status of an asynchronous invocation of the Fundraise Page Create endpoint

This endpoint requires Authorization via a Bearer Token

Url: <https://api.gofundraise.com.au/v1/pages/createstatus>

Method: GET

Parameters

requestids

- CSV list of request Ids received from asynchronous page creation requests

Response

```
{
  "Requests": [
    {
      "RequestId": Integer,
      "IsComplete": Boolean,
      "EventCampaignId": Integer,
      "FundraisingPages": [
        {
          "FundraisingPageId": Integer,
          "PageType": "Char",
          "IsActivated": Boolean,
          "ActivationId": "String",
          "Created": "DateTime",
          "HasError": Boolean,
          "Message": "String"
        }
      ]
    }
  ]
}
```

Fundraising Page Search

Overview

This allows clients to retrieve a list of active Fundraising Pages on the GoFundraise platform.

This is a public endpoint

Url: <https://api.gofundraise.com.au/v1/pages/search>

Method: GET

Paging

This endpoint is paged, with a maximum of 1000 results per page.

pagesize

- Number of results to return per query (Max 1000)

pageindex

- Which page in the query should be returned (Default 1)

sortby

- Sort results by either `CreatorName`, `PageTitle`, `CreateDate`, or `Total`

`sortorder`

- Sort results as ascending 'ASC' or descending 'DESC'

Parameters

pageid

- Search for a specific fundraising page ID

Includeteammembers

- Required "PageId". Include team members when searching for a team page by PageId

eventcampaignid

- Filter by the Event Campaign ID which corresponds to the Event Campaign on the GoFundraise platform (e.g. event campaign/appeal)

beneficiaryaccountid

- Filter by Beneficiary ID which corresponds to the Beneficiary Account (e.g. charity, school) on the GoFundraise platform

region

- Filter by region - US, AU, NZ

query

- String search for fundraising page title or creator name

pagetype

- Type of fundraising page, Individual or Team (S/T)

tags

- Filter pages by public tag names

tagids

- Filter pages by public tag internal ids if known

excludepageswithnofundraising

- Show or hide fundraising pages where fundraising is disabled in search results
excludepageswithrestrictedcreation
- Show or hide fundraising pages which are part of an event which has restricted fundraising page creation.
externalid
- Requires "ExternalIdType". Search by specific id of the fundraising beneficiary from an external platform if known
externalidtype
- Search by an external platform of the fundraising beneficiary

Response

```

{
  "Pages" : [
    {
      "Id": Integer,
      "Url": "String",
      "FundraisingEnabled": Boolean,
      "DisplayDonations": Boolean,
      "DisplayDonationAmount": Boolean,
      "DonationUrl": "String",
      "CreatorName": "String",
      "Beneficiary": {
        "BeneficiaryAccountId": Integer,
        "BeneficiaryName": "String",
        "BeneficiaryImagePath": "String",
        "BeneficiaryURL": "String",
        "BeneficiaryExternalIds": [
          {
            "Type": "String",
            "Id": "String"
          }
        ],
        "BeneficiaryTheme": {
          "DriverButtonBackgroundColour": "String",
          "DriverButtonFontColour": "String",
          "SecondaryButtonColour": "String",
          "SecondaryButtonFontColour": "String",
          "BannerBackgroundColour": "String"
        }
      },
      "Event": {
        "EventCampaignId": Integer,
        "EventName": "String",
        "EventImagePath": "String",
        "EventUrl": "String",
        "EventCreatePageUrl": "String",
        "EventTheme": {
          "DriverButtonBackgroundColour": "String",
          "DriverButtonFontColour": "String",
          "SecondaryButtonColour": "String",
    
```

```

        "SecondaryButtonFontColour" : "String",
        "BannerBackgroundColour": "String"
    },
    "defaults": {
        "currency": {
            "symbol": "String",
            "name": "String",
            "code": "String",
            "isoLanguageCode": "String"
        },
        "measurements": [
            {
                "unitShortName": "String",
                "unitName": "String",
                "type": "String",
                "unitMultiplier": "String"
            }
        ]
    },
    "ActivityData": {
        "TotalStepInNumber": "String",
        "TotalDistanceInMeters": "String",
        "TotalTimeInMinutes": "String",
        "TotalEnergyInCalories": "String"
    },
    "Team": {
        "TeamId": Integer,
        "TeamTarget": Decimal,
        "TeamTotal": Decimal,
        "JoinTeamUrl": "String"
    },
    "PageTags": [
        {
            "TagId": Integer,
            "TagName": "String",
            "TagType": "String"
        }
    ],
    "Meta": [
        {
            "MetaId": Integer,
            "MetaType": "String",
            "MetaName": "String",
            "MetaValueType": "String",
            "MetaValue": "String",
            "MetaDescription": "String",
            "MetaLastModified": "DateTimeOffset",
            "MetaLastModifiedBy": Integer
        }
    ],
    "PageTitle": "String",
    "ImagePath": "String",
    "Message": "String",
    "PageType": "Char",
    "RaiseTarget": Decimal,

```

```

        "Total": Decimal
    }
],
"ResultsReturned": Integer,
"PageNumber": Integer,
"TotalPages": Integer
}

```

Fundraising Page Retrieve

Overview

This allows clients to retrieve a Fundraising Pages on the GoFundraise platform. This endpoint will return non public information associated with the Fundraising page.

This endpoint requires Authorization via a Bearer Token

Url: <https://api.gofundraise.com.au/v1/pages/{pageid}>

Method: GET

Parameters

Pageid

- Retrieve a specific fundraising page by ID

Response

```

{
  "Id": Integer,
  "Url": "String",
  "DonationUrl": "String",
  "CreatorName": "String",
  "UserAccountId": "Integer",
  "PageTitle": "String",
  "ImagePath": "String",
  "Message": "String",
  "PageType": "Char",
  "RaiseTarget": Decimal,
  "Total": Decimal,
  "Team": {
    "TeamId": Integer,
    "TeamTarget": Decimal,
    "TeamImagePath": "String",
    "TeamTitle": "String",
    "TeamUrl": "String",
    "TeamDonationUrl": "String"
  },
  "Beneficiary": {
    "BeneficiaryAccountId": Integer,
    "BeneficiaryImagePath": "String",
    "BeneficiaryName": "String",
    "BeneficiaryUrl": "String",
    "BeneficiaryExternalIds": [
      {
        "Type": "String",
        "Id": "String"
      }
    ]
  }
}

```

```
        }
    ]
},
"Event": {
    "EventCampaignId": Integer,
    "EventImagePath": "String",
    "EventName": "String",
    "EventUrl": "String"
},
"Notes": "String",
"MinimumRaiseTarget": Decimal,
"DisplayDonations": Boolean,
"DisplayDonationAmount": Boolean,
"DisplayFitnessData": Boolean,
"Status": "Char",
"CreateDate": "DateTime",
"ExpireDate": "DateTime"
}
```

API Spec

The API spec document can be found here and will be updated when new functionality is released into production.

<https://support.gofundraise.com/hc/en-us/articles/235631127-API-Documentation>