GiftingNetwork REST API Overview

Draft v0.1

Contents

Document Information

Created by	Alkesh Kumar Singh
Reviewed By	Ruchir Gupta, Anurag Gaur and team
Contact	Eric Swerdlin <eric@giftingnetwork.com>

1   Introduction

This document provides an overview of the GiftingNetwork REST APIs to the downstream clients. Once connected and authenticated by our REST API Server, the downstream clients can use these APIs to fetch information of Contacts (Donors, Grant Seekers, etc.), Funds, Recommendations, Gifts and Grants of Donors, Organizations and more.

1.1   Purpose

The purpose of this document is to provide an overview of the GiftingNetwork REST APIs to the downstream clients. This document describes the requirements to connect, authorize and fetch information from the GiftingNetwork REST API Server.

1.2   Target Audience

The primary audience of this document includes API users, and various stack holders, product owner and development team.

1.3   Scope

Apart from the client authentication, these APIs allow access to the following business objects to downstream clients - Contacts, Funds, Fund Statements, Fund Recommendations, Gifts, Grants and Organizations.

1.4   Definitions

The following abbreviations are used in this document

S. No.	Abbreviation	Description
1.	API	Application Programming Interface
2.	HTTPS	Hypertext Transfer Protocol Secure
3.	REST	Representational State Transfer
4.	GN	GiftingNetwork

1.5   References

The following references are used while creating this document.

1.   GiftingNetwork REST API Reference document

2   Get Started

2.1   Basic Requirements

To access the GN REST APIs, one should have the following:

- o   An active API Username and Password
- o   Client Id
- o   GN REST API Server URL
- o   GN REST API Overview document (this document)
- o   GN REST API Reference document

If you are a valid API user, and if you do not have any of the above information or if you are facing an issue in accessing the APIs, please contact us.

2.2   API Security and Access-Token

All requests must be secure, i.e. https, not http.

In order to access any data API, you need an Access-Token. The Access-Token grants access to data APIs. For security reasons, the Access-Token automatically expires after certain time of inactivity (usually 15min). Please make sure that your Access-Token is never shared with anyone or compromised.

You can acquire an access token by using 'Login' API URL.

2.3   Server Base URL and API URL

The following is an example of server base URL:

SERVER_BASE_URL https://api.giftingnetwork.com/v1

The server base URL consists of two parts – domain URL (https://api.giftingnetwork.com/) and API version (v1).

Server Base URL should be added to the 'Endpoints' to form the API URL. For example, Login API URL is: API_URL https://api.giftingnetwork.com/v1/auth/login

Where,

https://api.giftingnetwork.com/v1 is Server Base URL and '/auth/login' is 'Login' Endpoint.

2.4   Accessing API Endpoints

All endpoints return either a JSON object or XML. The default Content-Type of all endpoints is 'application/json'. Please refer to 'GN REST API Reference document' for XML responses.

General steps to use GN REST APIs are:

1.   Make sure you have all information mentioned above in 'Basic Requirements' section.

2. Use 'Login' API to authenticate yourself. You will get an Access-Token in response. Keep it as it will be required while accessing other API Endpoints.
3. To call any data API, pass the Access-Token as Bearer Token in the header of the HTTP request along with other required parameters.

An overview of the API request and response format, along with endpoints and general information is provided in the next selection.

3   Overview

3.1   API End Points

This section provides an overview of the GN REST APIs without going into details of API request and response format. Please refer to GN API Reference document for details of each endpoint.

The following endpoints are available.

S. No.	Method	Endpoints	Description
1	POST	/auth/login	Returns Access-Token
2	POST	/auth/logout	Revokes Access-Token
3	GET	/contacts/search?	Returns a list of Contacts
4	GET	/contacts/{Contact-Id}	Returns a single Contact
5	GET	/funds/search?	Returns a list of donor Funds
6	GET	/funds/{Fund-Id}	Returns a single donor Fund
7	GET	/statements/{Fund-Id}	Returns a single Fund Statement
8	GET	/recommendations/search?	Returns a list of Recommendations
9	GET	/recommendations/{Fund-Id}	Returns a single Recommendation
10	GET	/gifts/search?	Returns a list of Gifts
11	GET	/grants/search?	Returns a list of Grants
12	GET	/organizations/search?	Returns a list of Organizations
13	GET	/organizations/{organization-Id}	Returns a single Organization
14			

The above API end-points are grouped into sub-sections for understanding only. The API groups do not have any other meaning.

3.1.1   Authentication

3.1.1.1   Login

Authenticates a user and return Access-Token.

Endpoint: /auth/login

This Access-Token must be passed as Bearer Token when calling any other APIs.

3.1.1.2   Logout

Revokes the Access-Token.

Endpoint: /auth/logout

The Access-Token cannot be used after revocation.

3.1.2   Contacts

3.1.2.1   Search Contacts

Returns a list of Contacts matching with the search criterion.

Endpoint: /contacts/search?{Query-Parameters}

This API supports the following query parameters:

- Contact Name, Email, Phone, etc.
- Contact Type (Donor, Grant Seeker, Students, etc.)

3.1.2.2   Get Contact by Id

Returns a single Contact matching with the received Contact-Id.

Endpoint: /contacts/{Contact-Id}

3.1.3   Funds
3.1.3.1   Search Funds

Returns a list of Funds matching with the search criterion.

Endpoint: /funds/search?{Query-Parameters}

This API supports the following query parameters:

- Fund Name, etc.
- Contact-Id

3.1.3.2   Get Fund by Id

Returns a single Fund matching with the received Fund-Id.

Endpoint: /funds/{Fund-Id}

3.1.4   Fund Statement
3.1.4.1   Get Fund Statement by Id

Returns the latest Fund Statement matching with the received Fund-Id.

Endpoint: /statements/{Fund -Id}

3.1.5   Grant Recommendations
3.1.5.1   Search Grant Recommendations

Returns a list of Grant Recommendations matching with the search criterion.

Endpoint: /recommendations/search?{Query-Parameters}

This API supports the following query parameters:

- Abc?, etc.
- Contact-Id
- Fund-Id

- Organization-Id

3.1.5.2   Get Recommendation by Id

Returns a single Recommendation matching with the received Recommendation-Id.

Endpoint: /recommendations/{Recommendation-Id}

3.1.6   Gift/Contribution History

3.1.6.1   Search Gift History

Returns a list of Gift History matching with the search criterion.

Endpoint: /gifts/search?{Query-Parameters}

This API supports the following query parameters:

- Gift date?, etc.
- Contact-Id
- Fund-Id

3.1.7   Grant/Disbursement History

3.1.7.1   Search Grant History

Returns a list of Grant History matching with the search criterion.

Endpoint: /grants/search?{Query-Parameters}

This API supports the following query parameters:

- Grant date?, etc.
- Contact-Id
- Fund-Id
- Organization-Id

3.1.8   Organizations

3.1.8.1   Search Organizations

Returns a list of Organizations matching with the search criterion.

Endpoint: /organizations/search?{Query-Parameters}

This API supports the following query parameters:

- Organization Name, etc.
- Contact-Id/Grant-Seeker-Id

3.1.8.2   Get Organization by Id

Returns a single Organization matching with the received Organization-Id.

Endpoint: /organizations/{Organization-Id}

3.2   API Response Templates

3.2.1   API Response – Single Object

```
Response Code: 200
Content-Type: application/json
{
        "param1": param1_value,
        "param2": param2_value,
        "object" : {
                "data1": data1_value,
                "data2": data2_value,
        },
        "status" : 1
}
```

Status value 1 or greater than 1 indicates success.

This type of APIs may return either parameters or object or both. The object may contain other type of objects.

3.2.2   API Response – List of Objects

```
Response Code: 200
Content-Type: application/json
{
        "param1": param1_value,
        "param2": param2_value,
        "objects" : [
          {
                "param1": param1_value,
                "param2": param2_value,
          },
          {
                "param1": param1_value,
                "param2": param2_value,
          },
        ],
        "paginator": {
                "count": COUNT,
                "page_number": PAGE_NUMBER,
                "page_size": PAGE_SIZE,
                "total": TOTAL,
                "has_more": MORE
        }
         "status" : 1
}
```

Where

COUNT – count of list items in this response

PAGE_NUMBER – current page number

PAGE_SIZE –may be different for different APIs. Generally, it should be between 5 and 25.

TOTAL – total items available on server.

MORE – true if there are more pages, false otherwise.

Status value 1 or greater than 1 indicates success.

This type of APIs return a list of objects and a pagination object. They may also return parameters. The objects may contain other type of objects.

3.2.3   API Response – Error

API response error is indicated by 'status' which is either 0 or less than 0 (i.e. any negative value).

The API client must check 'status' value before processing the response. The API response may contain 'error' object (and status value 0 or less than 0) when,

-   HTTP status code is 200
-   HTTP status code indicates an error (i.e. 4xx or 5xx)

```
Response Code: ANY (200/4xx/5xx)
Content-Type: application/json
{
        "error" : {
                "code": CODE,
                "title": TITLE,
                "detail": DETAIL,
                "ref_code": REF_CODE,
        },
        "status" : 0
}
```

Where,

TITLE (string) – A short, human-readable summary of the problem type.

CODE (integer) – It could be HTTP status code or any other integer value.

DETAIL (string) – Empty or a human-readable explanation of the problem.

REF_CODE (string) – A string that identifies this specific occurrence of the problem.

3.2.4   Query Parameters and Query Operators

All search APIs support query parameters. Some of the search APIs also support 'operators'. The operators can be used to narrow down the search criterion.

Though the details of the 'operators' supported the API are provided in the GN API Reference document, here we are describing its typical usage.

Suppose you want to search all contacts which were created after 2020. You can pass one query parameter and one query operator.

?created_at=2020&created_at_operator=gt

As shown in the above example, you can add "_operator" to the query parameter name to pass its operator.

Mapping of supported operators and their operator value if provided in the table below:

S. No.	Operator	Operator Value	Meaning
1	>	gt	Greater than
2	<	lt	Less than
3	>=	gte	Greater than or equal to
4	<=	lte	Less than or equal to
5	=	eq	Equal to
6	!=	neq	Not equal to
7	SQL like	like	String like

3.3   API Response Codes

In general, APIs may return the following HTTP status code:

HTTP Status Code	Title	Detail
404	Not Found	The item does not exist
400	Bad Input	The request format is not correct
401	Unauthorized	Authentication credentials were missing or incorrect
403	Forbidden	The request is understood, but it has been refused or access is not allowed
429	Too Many Requests	The request cannot be served due to the rate limit having been exhausted for the resource
500	Internal Server Error	Something is broken
503	Service Unavailable	The server is up, but overloaded with requests. Try again later!