Go Contacts API Documentation

Release 0.1.10a

Praekelt Foundation

April 03, 2015

Contents

Contents:

CHAPTER 1

Go Contacts HTTP API

This API is to be used to gain access to Contact and Group data within Vumi Go.

The API is indended to cover the following parts of the Vumi Go functionality:

- Contacts/Groups
  - Get one
  - Get all
  - Create one
  - Update one
  - Delete one

Request responses and bodies are all encoded in JSON, with the exception of errors. Streaming requests are encoded in newline separated JSON.

1.1 Contents

1.2 Response Format Overview

In the case of modifying a single object, that object will be returned formatted as JSON.

Example response (single object request):

```
HTTP/1.1 200 OK
{... "name": "foo" ...}
```

In the case of fetching multiple objects, there are two methods that can be used. The first method, pagination, separates the data into pages. The JSON object that is returned contains a `cursor` field, containing a cursor to the next page, and a `data` field, which contains the list of objects.

Example response (paginate request):

```
HTTP/1.1 200 OK
{"cursor": ..., "data": [{...},{...},...]}
```

The second method, streaming, returns one JSON object per line.

Example response (streaming request):

```
HTTP/1.1 200 OK
{... "name": "foo" ...}
{... "name": "bar" ...}
...
```

Errors are returned with the relevant HTTP error code and a json object, containing `status_code`, the HTTP status code, and `reason`, the reason for the error.

Example response (error response):

```
HTTP/1.1 404 Not Found
{"status_code": 404, "reason": "Group 'bad-group' not found."}
```

1.3 API Authentication

Authentication is done using an OAuth bearer token.

Example request:

```
GET /api/contacts/ HTTP/1.1
Host: example.com
Authorization: Bearer auth-token
```

Example response (success):

```
HTTP/1.1 200 OK
{"cursor": null, "data": []}
```

Example response (failure):

```
HTTP/1.1 403 Forbidden
```

Example response (no authorization header):

```
HTTP/1.1 401 Unauthorized
```

1.4 JSON Fields

The following section lists the valid fields that can be specified for each of the collections when creating and updating objects.

Contacts:

ANY /contacts/

JSON Parameters

- **msisdn** (*string*) – The MSISDN of the contact. Required to be non-null.
- **groups** (*list*) – A list of the group keys for the groups that this contact belongs to. Defaults to an empty list.
- **twitter_handle** (*string*) – The Twitter handle of the contact. Defaults to `null`.
- **bbm_pin** (*string*) – The BBM pin of the contact. Defaults to `null`.
- **extra** (*object*) – An object of extra information stored about the contact. Defaults to `{}`.
- **created_at** (*string*) – The timestamp of when the contact was created. Defaults to the current date and time. Uses the ISO 8601 format, ie. `YYYY-MM-DD hh:mm:ss`.
- **mxit_id** (*string*) – The MXIT ID of the contact. Defaults to `null`.
- **dob** (*string*) – The date of birth of the contact. Defaults to `null`.
- **key** (*string*) – The unique key used to identify the contact. Defaults to an automatically generated UUID4 key.
- **facebook_id** (*string*) – The Facebook ID of the contact. Defaults to `null`.
- **name** (*string*) – The name of the contact. Defaults to `null`.
- **surname** (*string*) – The surname of the contact. Defaults to `null`.
- **wechat_id** (*string*) – The WeChat ID of the contact. Defaults to `null`.
- **email_address** (*string*) – The email address of the contact. Defaults to `null`.
- **gtalk_id** (*string*) – The GTalk ID of the contact. Defaults to `null`.
- **subsription** (*object*) – An object storing the subscription information for the contact. Defaults to `null`.

Immutable contact fields:

ANY /contacts/

JSON Parameters

- **$VERSION** (*string*) – Represents the version of the contact.
- **user_account** (*string*) – The user account that the contact is linked to.

Groups

ANY /groups/

JSON Parameters

- **name** (*string*) – The name of the group. Required to be non-null.
- **key** (*string*) – The unique key used to identify the group. Defaults to an automatically generated UUID4 key.
- **query** (*string*) – The string representing the query for a smart group. Defaults to `null` representing a static group.
- **created_at** (*string*) – The timestamp of when the group was created. Defaults to the current date and time. Uses the ISO 8601 format, ie. `YYYY-MM-DD hh:mm:ss`.

Immutable group fields:

ANY /groups/

JSON Parameters

- **$VERSION** (*string*) – Represents the version of the group.
- **user_account** (*string*) – The user account that the group is linked to.

1.5 API Methods

GET / (str: *collection*) /
str: *object_key* Get a single object from the collection. Returned as JSON.

Request Headers

- Authorization – OAuth bearer token.

Parameters

- **collection** (*str*) – The collection that the user would like to access (i.e. `contacts` or `groups`)
- **object_key** (*str*) – The key of the object that the user would like to retrieve.

Status Codes

- 200 OK – no error
- 401 Unauthorized – no auth token
- 403 Forbidden – bad auth token
- 404 Not Found – contact for given key not found

Example request:

```
GET /api/contacts/b1498401c05c4b3aa6929204aa1e955c HTTP/1.1
Host: example.com
Authorization: Bearer auth-token
```

Example response (success):

```
HTTP/1.1 200 OK
Server: ...
Date: ...
Content-Type: text/html; charset=UTF-8
Content-Length: ...
Connection: keep-alive

{..., "key": "b1498401c05c4b3aa6929204aa1e955c", ...}
```

Example response (object not found):

```
HTTP/1.1 404 Not Found
Server: ...
Date: ...
Content-Type: application/json; charset=utf-8
Content-Length: 62
Connection: keep-alive

{"status_code": 404, "reason": "Contact 'bad-key' not found."}
```

GET **/** (**str:** *collection*) **/**
Returns all the objects in the collection, either streamed or paginated.

Query Parameters

- **query** – Not implemented.
- **stream** (*boolean*) – If `true`, all the objects are streamed, if `false`, the objects are sent in pages. Defaults to `false`.
- **max_results** (*int*) – If `stream` is false, limits the number of objects in a page. Defaults to server config limit. If it exceeds server config limit, the server config limit will be used instead.
- **cursor** (*string*) – If `stream` is false, selects which page should be returned. Defaults to `None`. If `None`, the first page will be returned.

Request Headers

- Authorization – OAuth bearer token.

Parameters

- **collection** (*str*) – The collection that the user would like to access (i.e. `contacts` or `groups`)

Status Codes

- 200 OK – no error
- 400 Bad Request – invalid query parameter usage
- 401 Unauthorized – no auth token
- 403 Forbidden – bad auth token

Pagination:

Response JSON Object

- **cursor** (*string*) – Cursor to send to get the next page.
- **data** (*list*) – List of collection objects within the page.

Streaming:

Response JSON Object

- **list** – New line separated list of JSON objects in the collection.

Example request (paginated):

```
GET /api/contacts/?stream=false&max_results=1&cursor=92802q70r52s4717o4ps413s12po5o63 HTTP/1.1
Host: example.com
Authorization: Bearer auth-token
```

Example response (paginated):

```
HTTP/1.1 200 OK
Server: ...
Date: ...
Content-Type: text/html; charset=UTF-8
Content-Length: ...
Connection: keep-alive

{"cursor": ..., "data": [{..., "name": "foo", ...}]}
```

Example request (streaming):

```
GET /api/contacts/?stream=true HTTP/1.1
Host: example.com
Authorization: Bearer auth-token
```

Example response (streaming):

```
HTTP/1.1 200 OK
Server: ...
Date: ...
Content-Type: text/html; charset=UTF-8
Connection: keep-alive

{..., "name": "bar", ...}
{..., "name": "foo", ...}
```

POST / (str: *collection*) **/**

Creates a single object in the collection.

Request Headers

- Authorization – OAuth bearer token.

Parameters

- **collection** (*str*) – The collection that the user would like to access (i.e. `contacts` or `groups`)

Request JSON Object

- **object** – The data that the new object should contain.

Status Codes

- 200 OK – no error
- 400 Bad Request – invalid JSON data
- 401 Unauthorized – no auth token
- 403 Forbidden – bad auth token

Response JSON Object

- **object** – The object that was created.

Example request:

```
POST /api/contacts/ HTTP/1.1
Host: example.com
Authorization: Bearer auth-token
Content-Length: 35

{"name": "Foo", "msisdn": "+12345"}
```

Example response:

```
HTTP/1.1 200 OK
Server: ...
Date: ...
Content-Type: text/html; charset=UTF-8
Content-Length: ...
Connection: keep-alive
```

```
{..., "msisdn": "+12345", "name": "Foo", ...}
```

PUT **/** (**str:** *collection*) **/**
str: *object_key* Updates a single object in the collection

Request Headers

- Authorization – OAuth bearer token.

Parameters

- **collection** (*str*) – The collection that the user would like to access (i.e. `contacts` or `groups`)
- **object_key** (*str*) – The key of the object that is to be modified.

Request JSON Object

- **object** – The data that the fields should be updated with.

Status Codes

- 200 OK – no error
- 400 Bad Request – invalid JSON data
- 401 Unauthorized – no auth token
- 403 Forbidden – bad auth token
- 404 Not Found – cannot find contact or bad contact key

Response JSON Object

- **object** – The object that was updated, with the updated fields.

Example request:

```
PUT /api/contacts/1e2dea8cffde4446a119c72697c38d5b HTTP/1.1
Host: example.com
Authorization: Bearer auth-token
Content-Length: 35

{"name": "Foo", "msisdn": "+12345"}
```

Example response:

```
HTTP/1.1 200 OK
Server: ...
Date: ...
Content-Type: text/html; charset=UTF-8
Content-Length: ...
Connection: keep-alive

{..., "msisdn": "+12345", "name": "Foo", ...}
```

DELETE **/** (**str:** *collection*) **/**
str: *object_key* Removes a single object from the collection.

Request Headers

- Authorization – OAuth bearer token.

Parameters

- **collection** (*str*) – The collection that the user would like to access (i.e. `contacts` or `groups`)
- **object_key** (*str*) – The key of the object that is to be removed.

Status Codes

- 200 OK – no error
- 401 Unauthorized – no auth token
- 403 Forbidden – bad auth token
- 404 Not Found – cannot find contact or bad contact key

Response JSON Object

- **object** – The object that was deleted.

Example request:

```
DELETE /api/contacts/68e456a0c8da43bea162839a9a1669c0 HTTP/1.1
Host: example.com
Authorization: Bearer auth-token
```

Example response:

```
HTTP/1.1 200 OK
Server: ...
Date: ...
Content-Type: text/html; charset=UTF-8
Content-Length: ...
Connection: keep-alive

{..., "key": "68e456a0c8da43bea162839a9a1669c0", ...}
```

CHAPTER 2

Indices and tables

- *genindex*
- *modindex*
- *search*

/(str:collection)

```
GET /(str:collection)/, 6
GET /(str:collection)/(str:object_key),
      6
POST /(str:collection)/, 8
PUT /(str:collection)/(str:object_key),
      9
DELETE /(str:collection)/(str:object_key),
      9
```

/contacts

```
ANY /contacts/, 5
```

/groups

```
ANY /groups/, 5
```