Archer Documentation

Release 0.1

Praekelt Dev

February 12, 2014

Contents

Contents:

CHAPTER 1

User Service

A simple rest based service, backed by Neo4J, to store information on users and relationships between them.

1.1 Installation

```
$ virtualenv ve
$ source ve/bin/activate
(ve)$ pip install -e .
(ve)$ twistd -n user-service \
        --endpoint=tcp:8081 \
        --database-connection-string=http://localhost:7474
```

1.2 API

GET /users/

Search for an existing user node.

Query Parameters

- **username** (*string*) – the username (optional).
- **msisdn** (*string*) – the MSISDN (optional).
- **email_address** (*string*) – the email address (optional).

Note: All three query string parameters are optional but at least one must be specified.

Response Headers

- **Content-Type** – will always be *application/json*.

Status Codes

- **200** – as this query cannot fail, it may return an empty result.

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

[
```

```
        {
            "user_id": "uuid1",
            "username": "the username",
            "msisdn": "27000000001",
            "email_address": "email@domain2.com"
        },
        {
            "user_id": "uuid2",
            "username": "the username",
            "msisdn": "27000000002",
            "email_address": "email@domain2.com"
        }
    ]
```

POST /users/

Create a new user node.

Json Parameters

- **username** (*string*) – The user's username.
- **msisdn** (*string*) – The user's MSISDN.
- **email_address** (*string*) – The user's email address.

Response Headers

- **Content-Type** – will always be *application/json*.

Status Codes

- **302** – and then redirects to `GET /users/(uuid:user_id)/`
- **400** – when json parameters are invalid or missing.

PUT /users/ (**uuid:** *user_id*) **/**

Update a user node.

Json Parameters

- **username** (*string*) – The user's username.
- **msisdn** (*string*) – The user's MSISDN.
- **email_address** (*string*) – The user's email address.

Response Headers

- **Content-Type** – will always be *application/json*.

Status Codes

- **200** – when update was successful.
- **400** – when json parameters are invalid or missing.

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "user_id": "uuid",
    "username": "the username",
    "msisdn": "27000000000",
```

```
    "email_address": "email@domain.com"
}
```

GET /users/ (**uuid:** *user_id*) **/**

Get a user node.

Response Headers

- **Content-Type** – will always be *application/json*.

Status Codes

- **200** – when the node was found.
- **404** – when the node was not found.

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "user_id": "uuid",
    "username": "the username",
    "msisdn": "27000000000",
    "email_address": "email@domain.com"
}
```

DELETE /users/ (**uuid:** *user_id*) **/**

Delete a user node.

Status Codes

- **204** – when the node was deleted.
- **404** – when the node was not found.

```
HTTP/1.1 204 No Content
Vary: Accept
```

CHAPTER 2

User Relationships

PUT /users/ (**uuid:** *user_id_1*) **/relationship/**
uuid: *user_id_2***/** Create a new relationship from *user_id_1* to *user_id_2*.

Json Parameters

- **relationship_type** (*string*) – The type of relationship. Currently only *LIKE* is supported.
- **relationship_props** (*dict*) – The extra properties to be stored in this relationship.

Response Headers

- **Content-Type** – will always be *application/json*.

Status Codes

- **302** – when update successful.
- **404** – when *user_id_1* or *user_id_2* does not exist.

GET /users/ (**uuid:** *user_id_1*) **/relationship/**
uuid: *user_id_2***/** Get the relationship of *user_id_1* to *user_id_2*.

Response Headers

- **Content-Type** – will always be *application/json*

Status Codes

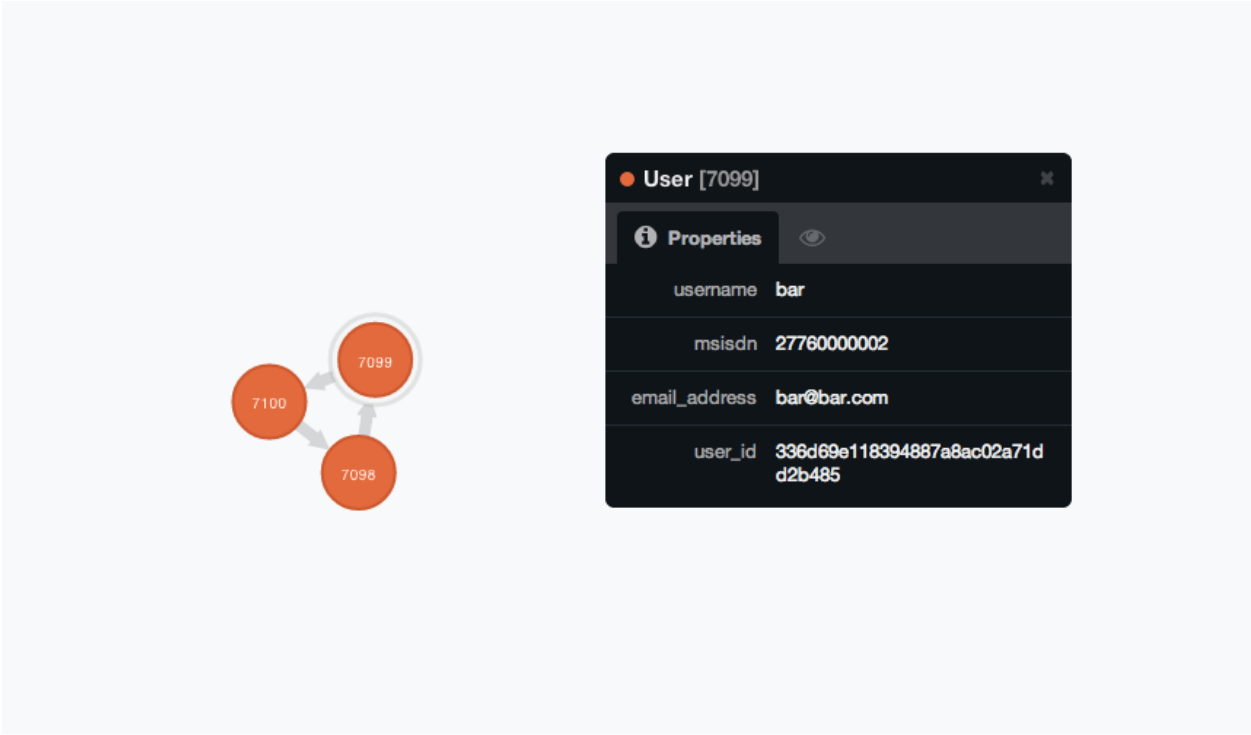
- **200** – when a relationship exists.
- **404** – when a relationship does not exist.

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
    "type": "LIKE"
    "data": {
        "kind": "comment"
    },
}
```

2.1 Example with curl

```
# Create user 1
$ curl -X POST -v http://localhost:8081/users/ \
    -d '{
    "msisdn": "27760000001",
    "username": "foo",
    "email_address": "foo@foo.com"}'

# redirects to created user with id `87e8da17eea541ed92bd861658d7e85e`

# Create user 2
$ curl -X POST -v http://localhost:8081/users/ \
    -d '{
    "msisdn": "27760000002",
    "username": "bar",
    "email_address": "bar@bar.com"}'

# redirects to created user with id `336d69e118394887a8ac02a71dd2b485`

# Create user 3
$ curl -X POST -v http://localhost:8081/users/ \
    -d '{
    "msisdn": "27760000003",
    "username": "baz",
    "email_address": "baz@baz.com"}'

# redirects to created user with id `e9b3d9e454c34516a318357fdfe04a6f`

# User 1 likes a comment by User 2
$ curl -X PUT -v http://localhost:8081/users/87e8da17eea541ed92bd861658d7e85e/relationship/336d69e11
    -d '{
            "relationship_type": "LIKE",
            "relationship_props": {
                "kind": "comment"
            }
        }'

# User 2 likes a comment by User 3
$ curl -X PUT -v http://localhost:8081/users/336d69e118394887a8ac02a71dd2b485/relationship/e9b3d9e454
    -d '{
            "relationship_type": "LIKE",
            "relationship_props": {
                "kind": "comment"
            }
        }'

# User 3 likes a comment by User 1
$ curl -X PUT -v http://localhost:8081/users/e9b3d9e454c34516a318357fdfe04a6f/relationship/87e8da17ee
    -d '{
            "relationship_type": "LIKE",
            "relationship_props": {
                "kind": "comment"
            }
        }'
```

This creates the following users and relationships:


7099
7100
7098
User [7099]
Properties
username bar
msisdn 27760000002
email_address bar@bar.com
user_id 336d69e118394887a8ac02a71dd2b485

CHAPTER 3

Indices and tables

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

/users

```
GET /users/, 3
POST /users/, 4
GET /users/(uuid:user_id)/, 5
PUT /users/(uuid:user_id)/, 4
DELETE /users/(uuid:user_id)/, 5
GET /users/(uuid:user_id_1)/relationship/(uuid:user_id_2)/,
      7
PUT /users/(uuid:user_id_1)/relationship/(uuid:user_id_2)/,
      7
```