1. **Info**

Stream is an API for building scalable news feeds and activity streams.

**Feed types**:

1. Timeline feeds
   1. Basic feeds contain a simple list of chronologically sorted activities from the topics or people you follow
2. Notification feeds
   1. Add the concept of "seen" or "read" to the activities, common with aggregation
3. Ranked feeds
   1. Allows to control how activities are sorted. Helps to improve engagement
4. Personalized feeds
   1. Leverage machine learning to further improve the relevancy of your feed
5. Aggregated feeds
   1. Enables to group activities together based on rules. Keeps feed relevant when there is a lot of activity
6. Real time
   1. Listen to feed changes in real-time and increase conversations on platform

**Features**: hashtags #, mentions @, likes, comments

It is an activity feed the iOs app with determine the layout

Where do we want to pull data from? Twitter, Facebook

We will use get stream API but then have to write our own API for Flash to iOs

we are using their API to access their services from our python code

but we are writing our own API for the iOS to get the data

**Core Concepts:**

Authentication Methods:

(1) application

\*For performing actions that are not specific to a single feed

*IETF: "Signing HTTP Messages"*

(2) feed

* Uses JSON Webtokens to authenticate every request
* It works by signing your API secret key which is the JSON formatted token
* Tokens (API secret key) will include a permission scope defined by '**resource ', 'action', 'feed\_id'**
  + Resource allows your to access 1 Endpoint
    - activities --> Activities Endpoint
    - feed --> Feed Endpoint
    - follower --> Following + Followers Endpoint
    - --> Access to any resource
  + Action defines which HTTP verb is allowed by request
    - Read --> GET, OPTIONS, HEAD
    - Write --> POST, PUT, PATCH
    - Delete --> DELETE
    - \* --> Permission to all verbs
  + feed\_id specifies for which feed permissions defined before are granted
    - The feed\_id value is a concatenation of the feed group + user\_id
    - \* --> grants permission on all feeds

\*\*When the token is generated correctly, it is used to authenticate a request. Two HTTP headers will be on the request: (Remove the prefix 'Bearer' if the library we use has it)

Stream-Auth-Type: jwt

Authorization: <token>

\*Both authentication methods create a signature that is specific to the requested resource (feed or application)

REST Endpoints:

1 – activities

* *Located* at ***activities/***
* Uses feed Authentication
* resource name **activities**
* Updates activity data stored by stream API

2 – feed

* *Located* at ***feed/(feed\_slug)/(user\_id)/***
* Uses feed Authentication
* resource name **feed**
* For retrieving activities GET or submitting activities POST
  + 6 parameters available for GET and 6 for POST
    - POST:
      * submit as single activity or a list
      * Base fields for a single activity: 'actor', 'verb', 'object', optional: 'target', 'time', 'to', 'foreign\_id'
      * Parameter for list of activities is 'activities' and then it follows the base fields above

3 – feed detail

* *Located at* **feed/(feed\_slug)/(user\_id)/(activity\_id|foreign\_id)/**
* Uses feed Authentication
* Resource name **feed**
* Delete activities by activity\_id (returned in response when activity is first created)

4 – followers

* *Located at* **feed/(feed\_slug)/(user\_id)/followers/**
* Uses feed Authentication
* Resource name **follower**
* Retrieve list of feed's followers
* GET

5 – following

* *Located at* **feed/(feed\_slug)/(user\_id)/follows/**
* Uses feed Authentication
* Resource name **follower**
* GET retrieve list of followers
* POST follow a target feed

6 – following detail

* *Located at* **feed/(feed\_slug)/(user\_id)/following/(target)/**
* Uses feed Authentication
* Resource name **follower**
* Allows to unfollow
* DELETE
* There is a keep\_history paramter, otherwise purged from feed
* Syntax = [feed]:[id]

7 – batch activity add

* *Located at* **feed/add\_to\_many/**
* Uses application Authentication
* Add activity to multiple feeds in one request
* POST

8 – batch follow

* *Located at* **follow\_many/**
* Uses application authentication
* Create multiple follows in one request
* POST

Basic app URL: [https://**us-east**-api.stream-io-api.com/api/v1.0/](https://us-east-api.stream-io-api.com/api/v1.0/)

Get parameters to send for every request:

Name = api\_key

Type = string

Description =

1. **Tasks**

* Install stream-python package
* Install flask-pymongo package
* Import stream and stream-python
* List fields for the database and their definitions by Stream:
  + 'actor'
  + 'verb'
  + 'object'
  + 'foreign\_id' - unique reference that represents an activity in the database
  + 'id', 'target'
  + 'time'
  + 'origin'
  + 'video\_name'
  + 'tweet'
  + 'youtube\_id'
  + 'to'
* Set request timeout to 3 seconds
* Variable Names for a consistent naming convention among API's
  + Api\_key
  + Api\_secret
  + App\_id
  + Api\_version
  + Client
  + Feed
  + Feed\_slug
  + User\_id
  + Feed\_id
  + Token
  + Signature
  + Signer
  + Foreign\_id
  + Location
  + STREAM\_URL

1. **Tutorial**
2. Flat feed/Twitter: (Post text/picture/video)
   1. Fields:
      1. Actor = user id of person performing activity
      2. Tweet = custom field containing message
      3. Verb = type of activity actor is engaging in
      4. Object = ID of tweet object in db
3. Follow:
   1. The posts of one user show up on the feed of another user

Sample API response:

{ "results": [

{

"actor": "eric",

"foreign\_id": "",

"id": "c660a7c7-222d-11e8-8414-0a51ae8e7f7a",

"object": "1",

"origin": "user:eric",

"target": "",

"time": "2018-03-07T17:34:23.189601",

"verb": "watch",

"video\_name": "Star Wars Trailer",

"youtube\_id": "JNwNXF9Y6kY"

},

{

"actor": "eric",

"foreign\_id": "",

"id": "7364ce82-221f-11e8-84c4-128899f22c76",

"object": "1",

"origin": "user:eric",

"target": "",

"time": "2018-03-07T15:51:51.011700",

"tweet": "Hello world",

"verb": "tweet"

},

{

"actor": "eric",

"foreign\_id": "",

"id": "2c10ae32-1ffd-11e8-a450-0a51ae8e7f7a",

"object": "1",

"origin": "user:eric",

"target": "",

"time": "2018-03-04T22:41:26.127775",

"verb": "watch",

"video\_name": "Star Wars Trailer",

"youtube\_id": "JNwNXF9Y6kY"

}

],

"next": "/api/v1.0/feed/timeline/jessica/?api\_key=zhzypp7eu79m&id\_lt=2c10ae32-1ffd-11e8-a450-0a51ae8e7f7a&limit=3",

"duration": "8.24ms"

}