1.0 LEARNING ANALYTICS DASHBOARD

1.1 System Overview

This document describes the installation and use of the Learning Analytics (LA) Dashboard. It is an instructor-facing, online tool for visualizing Twitter activity collected from students who post as part of a course.

The dashboard system consists of **two** main components

1. A **web interface** (front-end). This is the web-based interface the instructor sees when using the dashboard. It consists of a set of visualization panels that are each created from various queries that use collected Twitter data. It is housed in a single html file that uses Javascript and a pre-defined set of library functions to create queries and visualizations. It uses a third party repository to query data and set of library functions for the visualizations. An example of the dashboard can be seen here:

   https://dashboard.socialmediadata.org/educhat/

2. A **data processing system** (back-end). The back-end consists of a single Python script that uses the Twitter streaming API to collect tweets based on a particular query. It processes this data and sends it to the third-party repository which is subsequently used by the front-end interface

1.2 Purpose

This describes how to use and set up the dashboard environment as well as details of the system itself.

1.3 Audience

The audience for the document is for researchers who wish to use and deploy the system, and for developers who wish to modify and customize the system as necessary. Some knowledge of the Linux operating system is required, and preferably knowledge of the Python programming language.

1.4 Points of Contact

1.4.1 Information

Dr. Anatoliy Gruzd, Associate Professor
Nadia Conroy, Postdoctoral Researcher

1.4.2 Coordination

This tool is managed, authored and owned by Social Media Lab @ Ryerson TRSM

2.0 BACKGROUND

The affordances of Twitter in teaching and learning are many. The use of social media such as Twitter may be helpful in enriching learning by relying on networks of social ties, improving engagement, and allowing autonomy to create tailored learning environments.

This software is a Learning Analytics (LA) dashboard to support Twitter use in teaching. It is an instructor-facing, web tool that collects and monitors Twitter data. It uses data visualizations to summarize and present information on real-time Twitter use by students. The Learning Analytics (LA) Dashboard was developed by the Social Media Lab at Ryerson University.

The dashboard analyzes tweets produced under a specific course hashtag, and displays the resulting visualizations for an instructor to use for their own exploration. The goal is to allow instructors to see the ways in which students engage with each and respond to course material using Twitter.

2.1 Functionality Overview (Querying Course Hashtags)

The system is driven by the data collected through the Twitter Streaming API. The system listens for the presence of incoming tweets that are matched against a specific course hashtag. For example, if the dashboard were being used in the course, *Information Literacy and Instruction LIS2001*, the hashtag #LIS2001 could be used to designate tweets relating specifically to content for that course.

Each tweet with a hashtag is collected. The data is processed, and certain metadata from the tweet are extracted and then stored in an external database repository (Keen.io). The dashboard system queries this repository to generate visualizations that compose the dashboard. The transactional flow of information is represented in Figure 1.

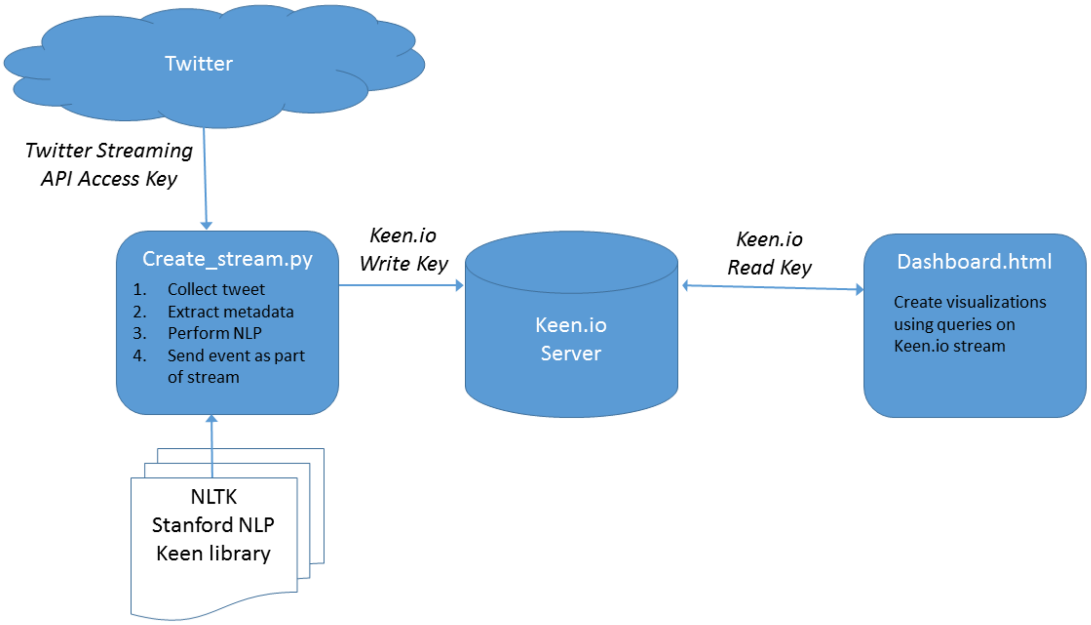


Figure 1: Dashboard system dataflow

3.0 SETTING UP DASHBOARD INSTANCE

The section describes how to instantiate a version of the dashboard to work with a particular course, using a unique hashtag to query the Twitter stream for incoming tweets.

It consist of a back-end script to collect, analyze and store Tweets, and a front-end to use a data source of tweets to query and present the content.

The following steps are based on using the Ubuntu Linux-based operating system, version 14.04

3.1 Install and Configure Apache Webserver

The Apache web server allows the html to be viewable to external requests.

- Make yourself the root superuser

```
$ sudo su
```

- From the terminal, install Apache with the command

```
$ apt install apache2
```

- Confirm the installation of Apache

```
$ dpkg --get-selections | grep apache
```

- Locate the web root directory on the Apache server:

```
$ grep -R "DocumentRoot" /etc/apache2/sites-available
```

This will show you the default location of the web directory that has been installed in your system. Typically this will be the directory:

```
/var/www/html
```

- To customize the directory location you would like to contain the dashboard, first create the directory. For example,

```
$cd /var/www/
$ mkdir –p dashboard.socialmediadata.org/public_html/dashboard
```

- Now we must have the browser point to correct default directory. Go to the Sites-Available directory

```
$ cd /etc/apache2/sites-available
```

- Create a template config file using the copy command. Replace LIS2001 with the name of your chosen hashtag:

```
$ sudo cp 000-default.conf LIS2001.conf
```

- Then edit the new .conf file by opening it in an editor:

```
$ sudo vim LIS2001.conf
```

- Replace the line

```
DocumentRoot /var/www/html
```

with:

```
DocumentRoot /var/www/dashboard.socialmediadata.org/public_html/dashboard/
```

To match the directory that we created in the previous steps

- Now we must delete the symbolic link from sites-available to sites-enabled for the 000-default.conf file so that it isn't the default:

```
$ cd /etc/apache2/sites-enabled
$ sudo rm 000-default.conf
```

- After this, we create a symbolic link from /etc/apache2/sites-available/la_dashboard_gen.conf to /etc/apache2/sites-enabled

```
$ cd /etc/apache2/sites-enabled/
$ sudo ln -s ../sites-available/LIS2001.conf LIS2001.conf
```

- Restart the Apache webserver to update the changes

```
$ sudo service apache2 restart
```

- Apache2 makes use of both global and site specific configuration files. For background about the hierarchy of configuration files, take a look at [How To Configure the Apache Web Server on an Ubuntu or Debian VPS]

3.2 Download Dashboard Files from BitBucket Repository

The system consists of the modules and external packages listed in Table 1. The file package is located on the Bit Bucket repository la_dasboard_gen. All application files are installed from the repository to the directory tree:

File Type	Description
LA_dashboard_gen/html/index.html	Dashboard front end files containing visualization s and queries
LA_dashboard_gen/html/sub_sent.html	Secondary html file showing samples of sentiment and subjectivity classification
LA_dashboard_gen/scripts/create_stream.py	Python program files used for collecting and processing data
LA_dashboard_gen/config/dashboard.config	Configuration file containing access keys, project keys, read and write keys for Twitter and Keen.io APIs
LA_dashboard_gen/assets/css/keen-dashboards.css	Files used for formatting html

Table 1: File types comprising dashboard system

Complete the following steps to use the repository to install the dashboard.

- From the terminal, navigate to the web root directory you set up in the previous step, (3.1 Install and Configure Apache Webserver)

  ```
  $ cd /var/www/dashboard.socialmediadata.org/public_html/dashboard/
  ```

- Make a directory for your dashboard. It should match the hashtag for the course. For example if your hashtag is `#LIS2001`. The name you choose here will be displayed in the web URL of your dashboard

  ```
  $ mkdir LIS2001

  $ cd LIS2001
  ```

- Clone the repository. This will copy all the content files listed in Table 1 to your current directory

  ```
  git clone https://n1conroy@bitbucket.org/n1conroy/LA_dashboard_gen.git
  ```

- Go to the new dashboard directory

  ```
  $ cd LA_dashboard_gen
  ```

- Ensure the files were pulled from the repository successfully

  ```
  $ ls -l

  drwxr-xr-x 5 root root 4096 Oct 16 19:51 assets
  drwxr-xr-x 2 root root 4096 Jan 29 20:26 config
  drwxr-xr-x 2 root root 4096 Jan 29 21:11 html
  -rw-r--r-- 1 root root   34 Oct 17 20:11 README.md
  drwxr-xr-x 2 root root 4096 Jan 29 20:21 scripts
  ```

- Copy the file tree to the proper dashboard directory

  ```
  $ cp -a * ../
  $ mv html/index.html ../
  $ mv html/sent_sub.html ../
  ```

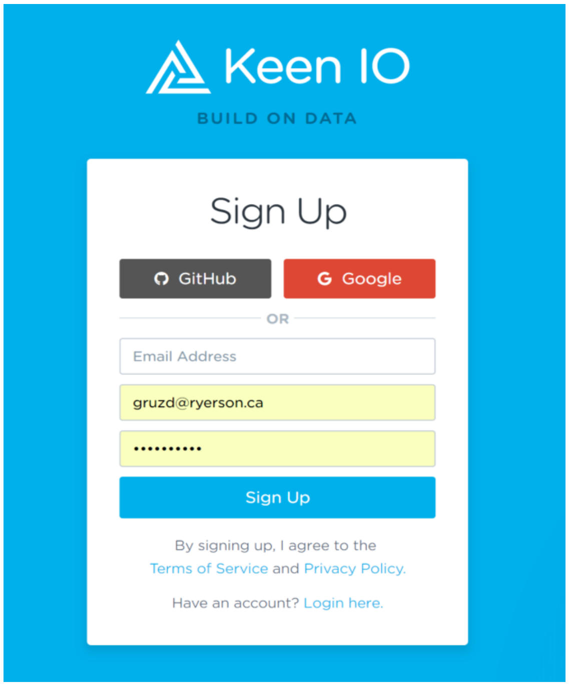
Optional: If you are using an OS other than Ubuntu, ensure you follow the installation instructions for your particular platform: http://httpd.apache.org/docs/2.4/install.html

3.2 Register a Keen.IO Account

The dashboard is largely driven by functionality from Keen.io, a third party data repository and set of visualization libraries. Keen.io allows data to be uploaded, stored, queried, and visualized through an API.

To access the Keen.io API it is necessary to register a free user account with the Keen.io platform.

- Visit the Keen.io main site https://keen.io/ and click the Sign Up button on the top right.



Keen IO

BUILD ON DATA

Sign Up

Email Address

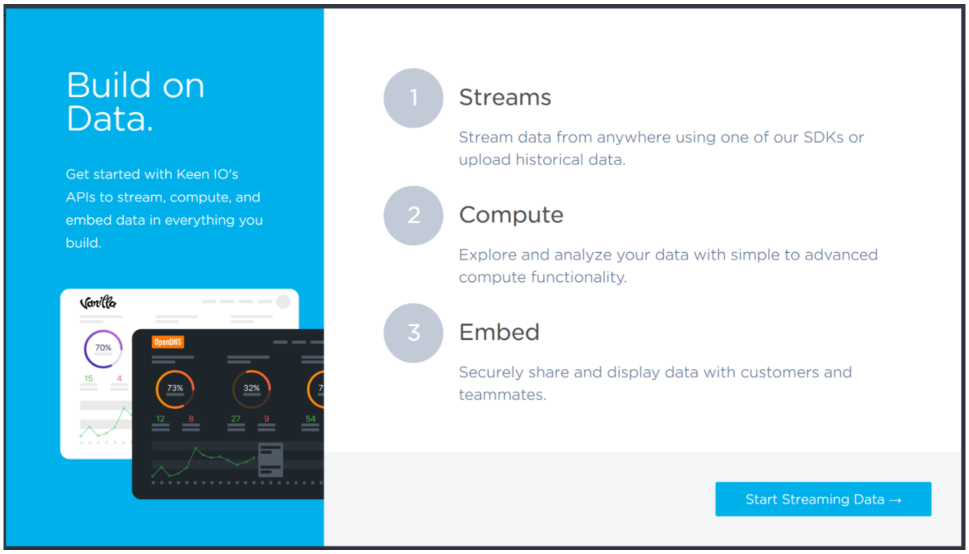
gruzd@ryerson.ca

••••••••••

By signing up, I agree to the Terms of Service and Privacy Policy.

Have an account? Login here.

- Log in with your Google id, your GitHub id or an email address with a password



Build on Data.

Get started with Keen IO's APIs to stream, compute, and embed data in everything you build.

1 Streams

Stream data from anywhere using one of our SDKs or upload historical data.

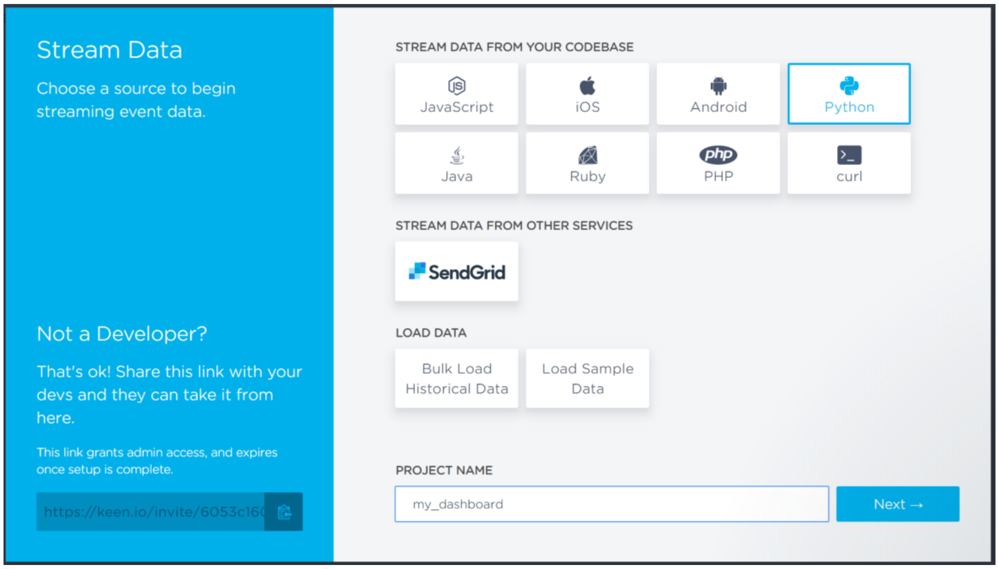
2 Compute

Explore and analyze your data with simple to advanced compute functionality.

3 Embed

Securely share and display data with customers and teammates.

- Click the link “Start Streaming Data”



Stream Data

Choose a source to begin streaming event data.

Not a Developer?

That's ok! Share this link with your devs and they can take it from here.

This link grants admin access, and expires once setup is complete.

https://keen.io/invite/6053c16(

STREAM DATA FROM YOUR CODEBASE

JavaScript | iOS | Android | Python

Java | Ruby | PHP | curl

STREAM DATA FROM OTHER SERVICES

SendGrid

LOAD DATA

Bulk Load Historical Data | Load Sample Data

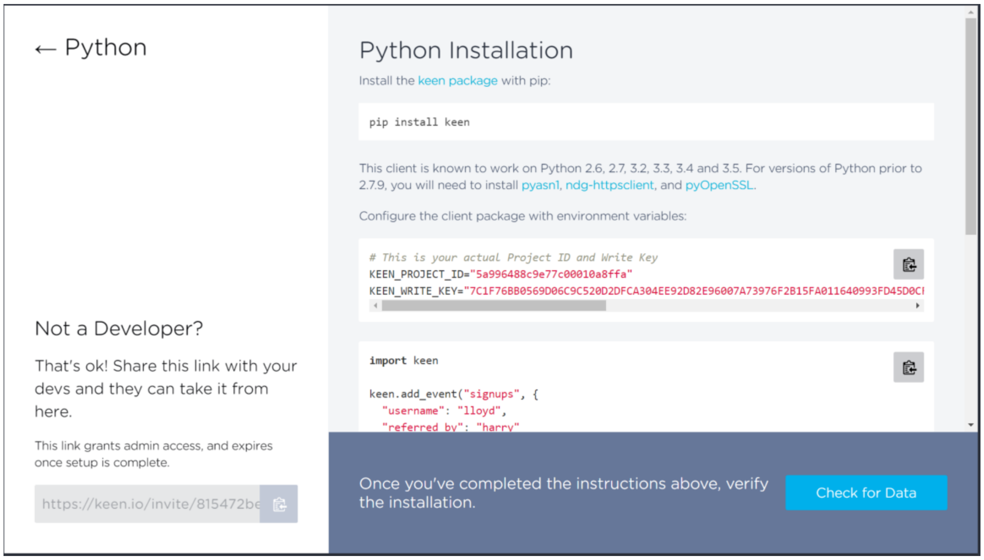
PROJECT NAME

my_dashboard

Next →

- Choose the codebase you will stream data from, in this case select “Python”

- Next, choose a name for your project. You can choose any name you wish. Select “Next”



← Python

Not a Developer?

That's ok! Share this link with your devs and they can take it from here.

This link grants admin access, and expires once setup is complete.

https://keen.io/invite/815472be

Python Installation

Install the keen package with pip:

```
pip install keen
```

This client is known to work on Python 2.6, 2.7, 3.2, 3.3, 3.4 and 3.5. For versions of Python prior to 2.7.9, you will need to install pyasn1, ndg-httpsclient, and pyOpenSSL.

Configure the client package with environment variables:

```
# This is your actual Project ID and Write Key
KEEN_PROJECT_ID="5a996488c9e77c00010a8ffa"
KEEN_WRITE_KEY="7C1F76BB0569D06C9C520D2DFCA304EE92D82E96007A73976F2B15FA011640993FD45D0CI
```

```
import keen

keen.add_event("signups", {
  "username": "lloyd",
  "referred_by": "harry"
```

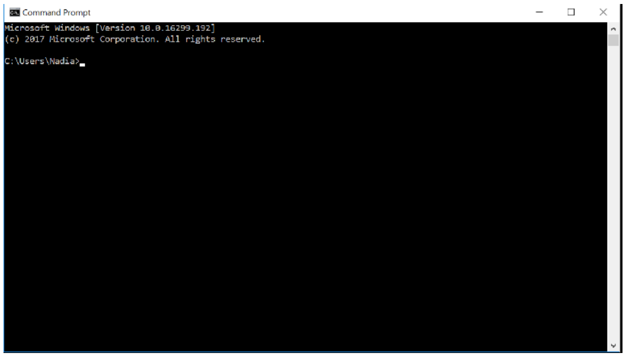
Once you've completed the instructions above, verify the installation.

Check for Data

- Looks great! Now pause here. Before proceeding, we must install Python on the server, and then test the configuration with Keen. We will be returning to this screen shortly, so please keep the browser window open.

3.3 Install Python & Library Dependencies

a. First, open a Terminal console



```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Nadia>
```

b. On Ubuntu 14.04 Python 2.7 and 3.4 is installed by default. If you are on a different platform follow steps b and c to install a Python environment. Otherwise proceed to step d.

```
$ sudo apt-get install build-essential checkinstall
$ sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev
libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev
```

c. Then install a version of Python 2.7

```
$ sudo apt-get update
$ sudo apt dist-upgrade
$ sudo apt-get install python2.7 python-pip
```

d. Install the various third-party library dependencies:

NLTK interface:

```
$ sudo pip install -U nltk
$ sudo pip install -U textblob
$ python -m textblob.download_corpora
```

Keen libraries:

```
$ sudo npm install keen.io
```

Twitter API:

```
$ sudo pip install tweepy
```

Java:

```
$ sudo apt-get update
$ java -version
```

If you don’t see a version of java installed, run the command to get the Java runtime environment

```
$ sudo apt-get install default-jre
```

e. Download Stanford Named Entity Recognizer

```
$ wget http://nlp.stanford.edu/software/stanford-ner-2014-06-16.zip
```

Alternatively, download the stanford-ner.jar and classifier model located here: Named Entity Recognizer version 3.4

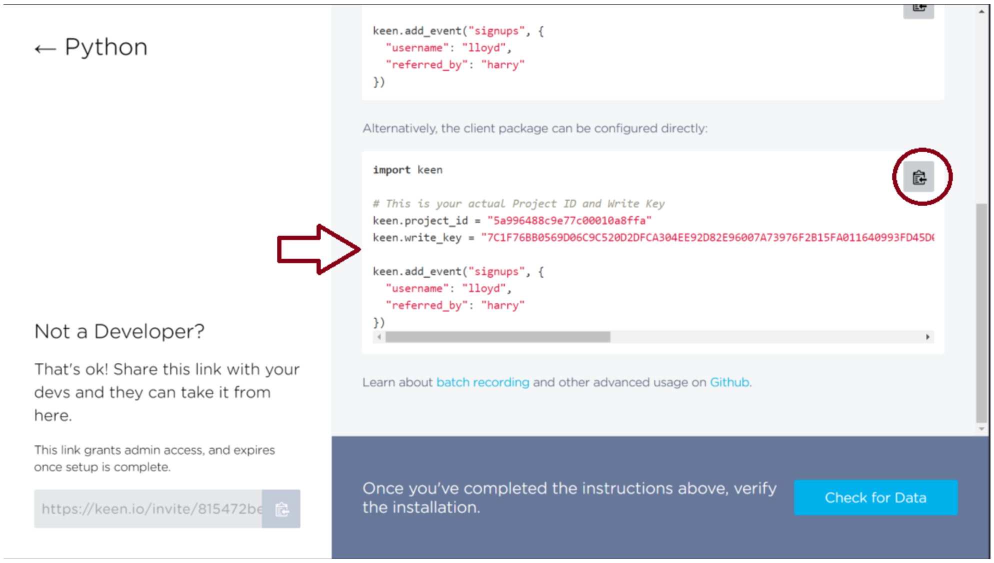
f. Unzip the contents to the destination folder. Install the unzip tool if necessary. The backend script will look for the entity recognizer in the `/usr/local` directory

```
$ sudo apt-get install unzip
$ unzip stanford-ner-2014-06-16.zip
$ sudo mv stanford-ner-2014-06-16 /usr/local/
$ sudo mv /usr/local/stanford-ner-2014-06-16 /usr/local/stanford-ner
```

3.4 Test Python and Keen.io configuration

- Return to the browser window that was left open from the end of Step 3.2, Register a Keen.io account

- Copy the code in the box under the text "Alternatively, the client package can be configured directly." This is the code used to test the Keen configuration. (You can use the Copy to Clipboard icon at the top right)



```
keen.add_event("signups", {
  "username": "lloyd",
  "referred_by": "harry"
})
```

Alternatively, the client package can be configured directly:

```
import keen

# This is your actual Project ID and Write Key
keen.project_id = "5a996488c9e77c00010a8ffa"
keen.write_key = "7C1F76BB0569D06C9C520D2DFCA304EE92D82E96007A73976F2B15FA011640993FD45D

keen.add_event("signups", {
  "username": "lloyd",
  "referred_by": "harry"
})
```

Learn about batch recording and other advanced usage on Github.

Not a Developer?

That's ok! Share this link with your devs and they can take it from here.

This link grants admin access, and expires once setup is complete.

https://keen.io/invite/815472be

Once you've completed the instructions above, verify the installation.

- Using a text editor in the Terminal, create a Python test file. Call it keentest.py

```
$ vi keentest.py
```

- Paste the following lines of code you just copied into the file, replacing XXX with your current Keen Project ID and Keen Write Key.

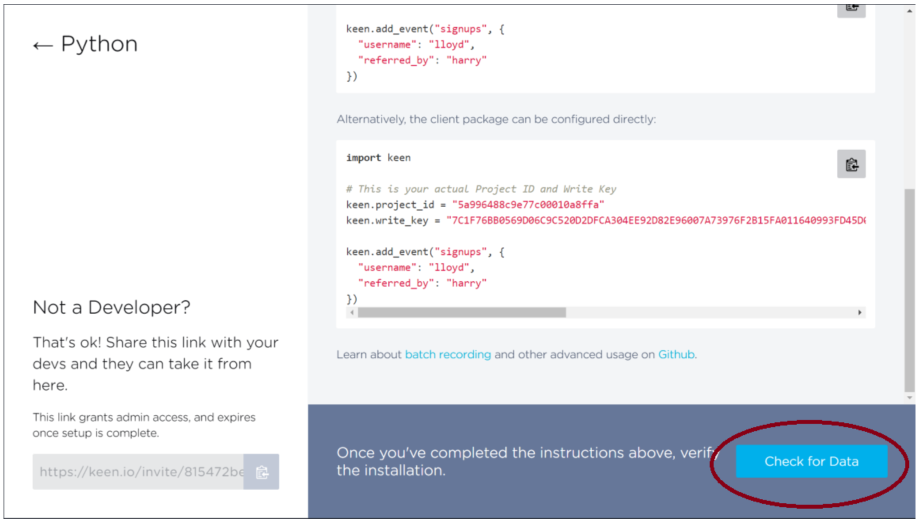
```
import keen

# This is your actual Project ID and Write Key

keen.project_id = "XXX”
keen.write_key =  "XXX”

keen.add_event("signups", {
  "username": "lloyd",
  "referred_by": "harry" })
```

This script imports the Keen library, sets your Project ID and Write Key values, and then adds an event to a stream using these credentials. Save and exit the editor.

- Run the Python test file.

  ```
  $ python keentest.py
  ```

  You should not see any output or errors as result of running the script. (If you see errors, either your Python installation failed or there is a mistake in the keentest.py file.)



← Python

```
keen.add_event("signups", {
  "username": "lloyd",
  "referred_by": "harry"
})
```

Alternatively, the client package can be configured directly:

```
import keen

# This is your actual Project ID and Write Key
keen.project_id = "5a996488c9e77c00010a8ffa"
keen.write_key = "7C1F76BB0569D06C9C520D2DFCA304EE92D82E96007A73976F2B15FA011640993FD45D

keen.add_event("signups", {
  "username": "lloyd",
  "referred_by": "harry"
})
```

Learn about batch recording and other advanced usage on Github.

Not a Developer?

That's ok! Share this link with your devs and they can take it from here.

This link grants admin access, and expires once setup is complete.

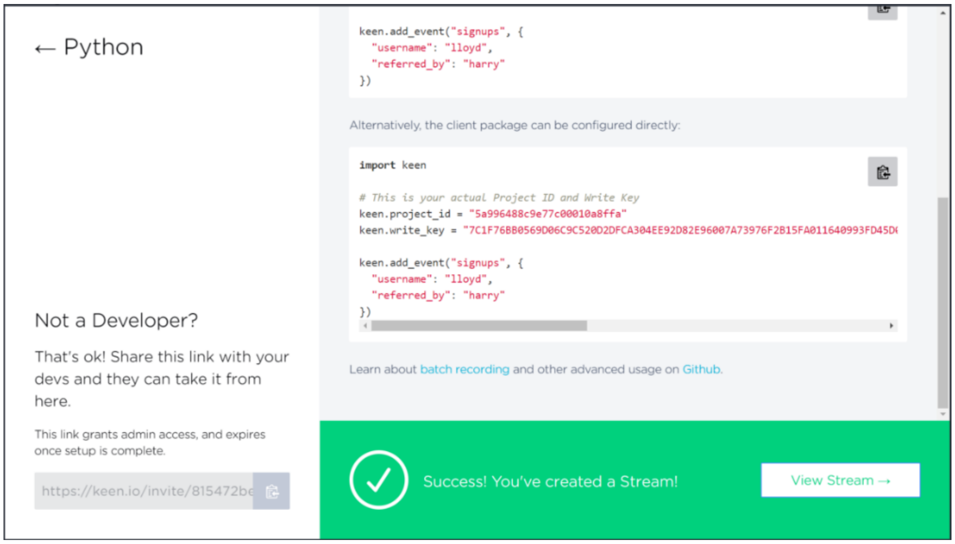
https://keen.io/invite/815472be

Once you've completed the instructions above, verify the installation.

- Return to the browser window showing the Keen registration

- Click the button, Check for Data

- If the test was successful you should see the message “Success! You’ve created a Stream!” If you do not see this message go and verify the contents of keentest.py file. Otherwise, click the View Stream button.



← Python

```
keen.add_event("signups", {
  "username": "lloyd",
  "referred_by": "harry"
})
```

Alternatively, the client package can be configured directly:

```
import keen

# This is your actual Project ID and Write Key
keen.project_id = "5a996488c9e77c00010a8ffa"
keen.write_key = "7C1F76BB0569D06C9C520D2DFCA304EE92D82E96007A73976F2B15FA011640993FD45D

keen.add_event("signups", {
  "username": "lloyd",
  "referred_by": "harry"
})
```

Learn about batch recording and other advanced usage on Github.

Not a Developer?

That's ok! Share this link with your devs and they can take it from here.

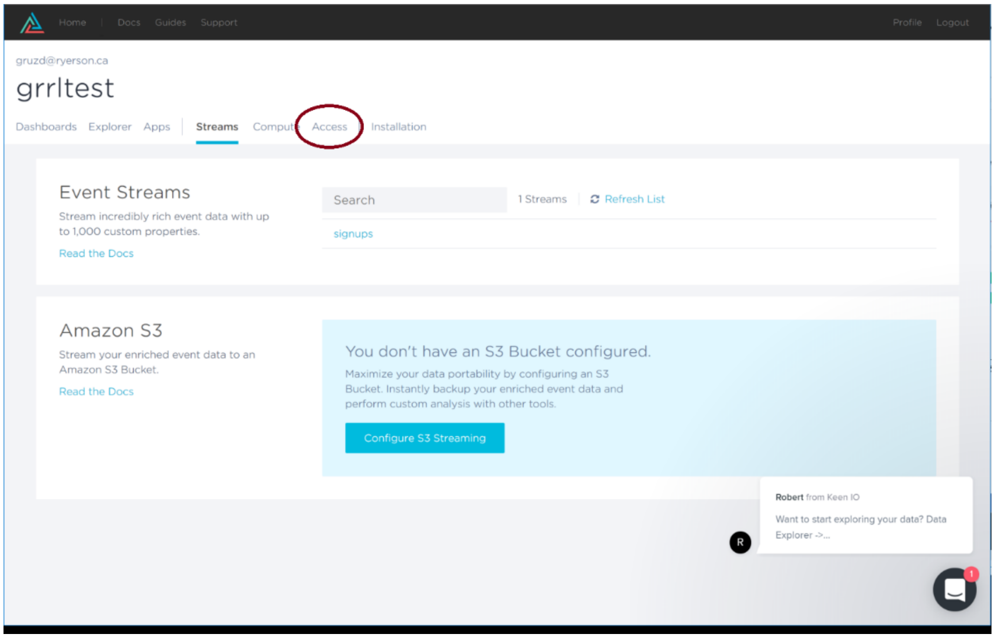
This link grants admin access, and expires once setup is complete.

https://keen.io/invite/815472be

Success! You've created a Stream!

- You should then see your account front page with the stream you just created. The stream should be called ‘signups’ containing one event.

- Select the Access tab at the top of the page



Event Streams

Stream incredibly rich event data with up to 1,000 custom properties.

Read the Docs

Amazon S3

Stream your enriched event data to an Amazon S3 Bucket.

Read the Docs

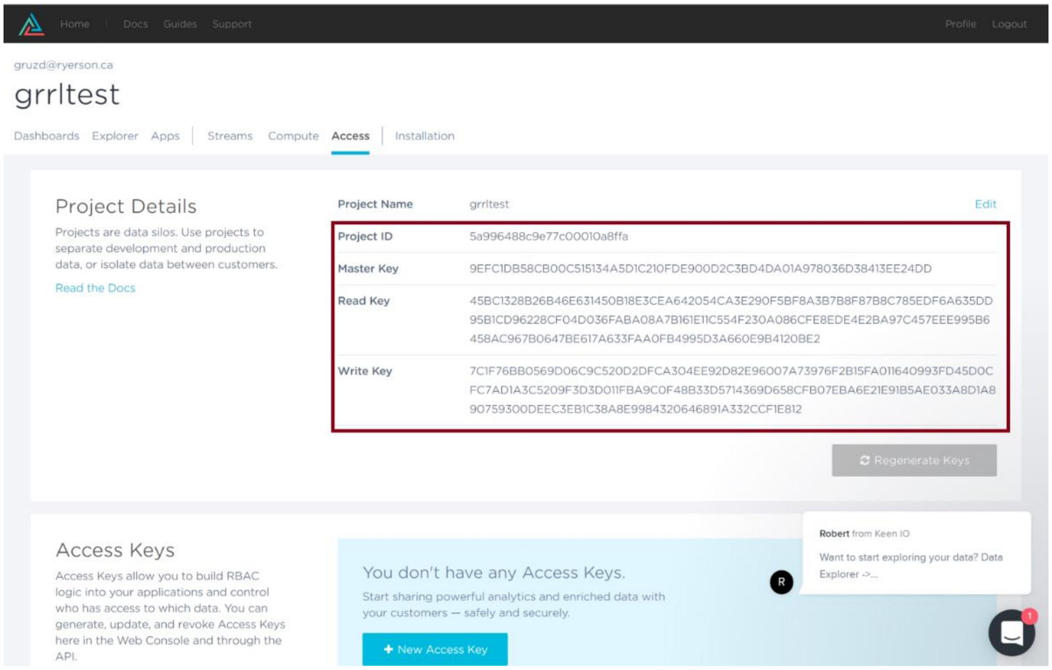
You don't have an S3 Bucket configured.

Maximize your data portability by configuring an S3 Bucket. Instantly backup your enriched event data and perform custom analysis with other tools.

Robert from Keen IO

Want to start exploring your data? Data Explorer ->...

- You will be provided the Access Keys, credentials to use the Keen.io API. These keys identify the source or user making a request to Keen.io. The keys you require are:
  1. Project ID
  2. Read Key
  3. Write Key



gruzd@ryerson.ca

grrltest

Project Details

Projects are data silos. Use projects to separate development and production data, or isolate data between customers.

Read the Docs

Project Name	grrltest
Project ID	5a996488c9e77c00010a8ffa
Master Key	9EFC1DB58CB00C515134A5D1C210FDE900D2C3BD4DA01A978036D38413EE24DD
Read Key	45BC1328B26B46E631450B18E3CEA642054CA3E290F5BF8A3B7B8F87B8C785EDF6A635DD 95B1CD96228CF04D036FABA08A7B161E11C554F230A086CFE8EDE4E2BA97C457EEE995B6 458AC967B0647BE617A633FAA0FB4995D3A660E9B4120BE2
Write Key	7C1F76BB0569D06C9C520D2DFCA304EE92D82E96007A73976F2B15FA011640993FD45D0C FC7AD1A3C5209F3D3D011FBA9C0F48B33D5714369D658CFB07EBA6E21E91B5AE033A8D1A8 90759300DEEC3EB1C38A8E9984320646891A332CCF1E812

Access Keys

Access Keys allow you to build RBAC logic into your applications and control who has access to which data. You can generate, update, and revoke Access Keys here in the Web Console and through the API.

You don't have any Access Keys.

Start sharing powerful analytics and enriched data with your customers — safely and securely.

Robert from Keen IO

Want to start exploring your data? Data Explorer ->...

- Copy the three keys, as you will need to use them later as part of the installation process.

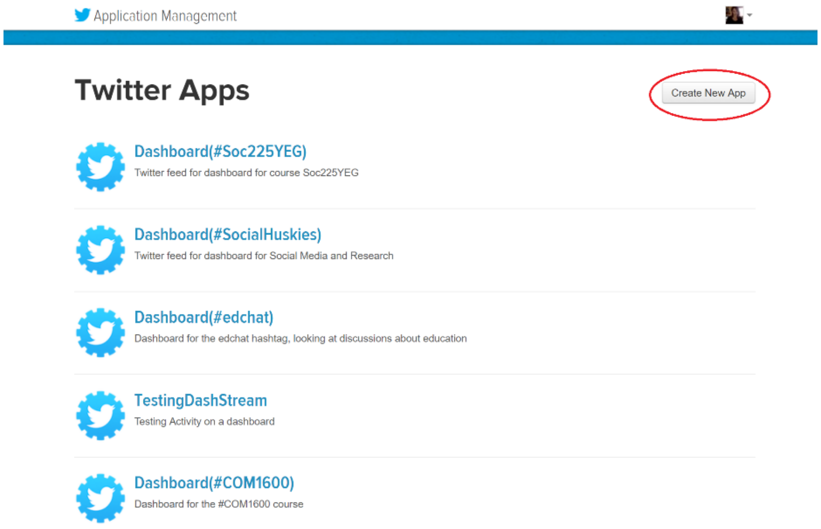
3.5 Register the Twitter App

To restrict the traffic on Twitter to a single API key, a new Application must be registered that specifies the version of the dashboard you wish to instantiate.

- If you have a Twitter user account, go to the Application Management page (https://apps.twitter.com/). Here you will see a list of your existing applications you have registered with Twitter.

- If you don’t have a Twitter account you must create one fist and then proceed with the remaining steps.

- Click on the Create New App button on the top right to create a new app for the dashboard.



Application Management

Twitter Apps

Dashboard(#Soc225YEG)
Twitter feed for dashboard for course Soc225YEG

Dashboard(#SocialHuskies)
Twitter feed for dashboard for Social Media and Research

Dashboard(#edchat)
Dashboard for the edchat hashtag, looking at discussions about education

TestingDashStream
Testing Activity on a dashboard

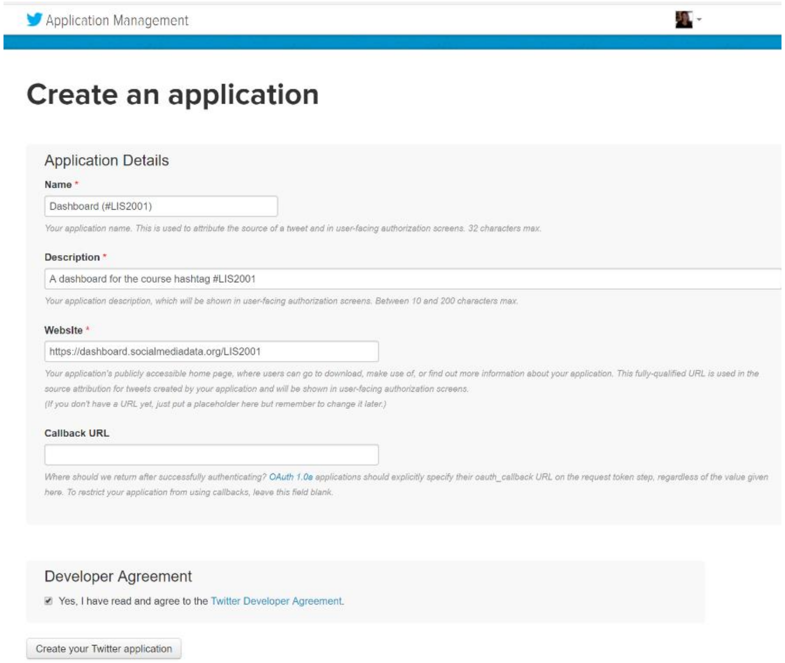
Dashboard(#COM1600)
Dashboard for the #COM1600 course

- This will take you to the App registration page. Fill in the required fields, including the name of the app should follow the naming convention:

  `Dashboard(#COURSECODE)`

  The URL for the dashboard should be the actual web address of the dashboard. The URL follows the naming convention, `https://dashboard.socialmediadata.org/<hashtag>/`

- When you are done filling out the fields, click ‘Create Your Twitter Application’.



Application Management

Create an application

Application Details

Name *
Dashboard (#LIS2001)
Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *
A dashboard for the course hashtag #LIS2001
Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *
https://dashboard.socialmediadata.org/LIS2001
Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

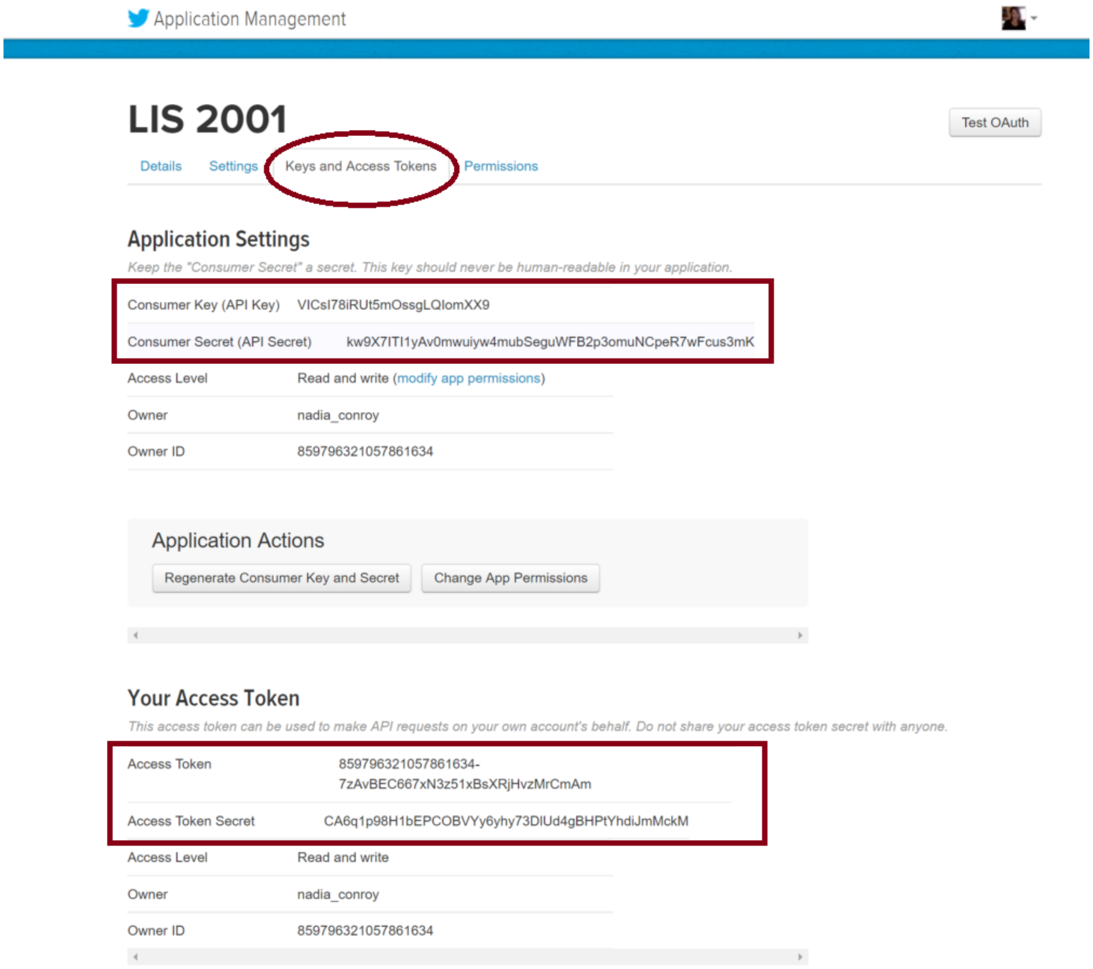
Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Agreement

☑ Yes, I have read and agree to the Twitter Developer Agreement.

- After filling out the required fields, click the ‘Create your Twitter application’ on the bottom of the page. Upon doing so you will be taken to the main page for the application.

- Select the Keys and Access Tokens tab.
- Then click the Generate Access Token button on the bottom, as shown,



LIS 2001

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	VICsI78iRUt5mOssgLQlomXX9
Consumer Secret (API Secret)	kw9X7ITI1yAv0mwuiyw4mubSeguWFB2p3omuNCpeR7wFcus3mK
Access Level	Read and write (modify app permissions)
Owner	nadia_conroy
Owner ID	859796321057861634

Application Actions

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	859796321057861634- 7zAvBEC667xN3z51xBsXRjHvzMrCmAm
Access Token Secret	CA6q1p98H1bEPCOBVYy6yhy73DlUd4gBHPtYhdiJmMckM
Access Level	Read and write
Owner	nadia_conroy
Owner ID	859796321057861634

- You will need the keys and access tokens on this page, specifically Consumer Key, Consumer Secret, Access Token, and Access Token Secret. Make note of them for later.

3.6 Edit the Config file

The next step is to edit the config file so that it reads the proper Twitter Keys and Keen.io access keys. .

- Navigate to the directory of dashboard config files. (Use your own hashtag in the place of LIS2001)

```
$ cd
/var/www/dashboard.socialmediadata.org/public_html/dashboard/LIS2001/config
```

- Open the dashboard.config file in a text editor:

```
$ vi dashboard.config
```

- Upon opening the file, you will see blank access keys. Populate the key/value pairs in this file, replacing each XXX to the values that were provided when you registered the Twitter app and when you created a Keen.io account.

  ```
  CONSUMER_KEY=XXX
  CONSUMER_SECRET=XXX
  ACCESS_TOKEN_SECRET=XXX
  ACCESS_TOKEN=XXX
  KEEN_PROJECT_ID=XXX
  KEEN_WRITE_KEY=XXX
  KEEN_READ_KEY=XXX
  ```

- Export all 7 values to environment variables. Type the following form the current directory,

  ```
  $ for i in `cat`; do export $i; done < dashboard.config
  ```

- Change the name of the configuration file so it is appropriate for the dashboard hashtag (replace LIS2001 with your hashtag).

  ```
  $ mv dashboard.config LIS2001.config
  ```

- To ensure the environment variables were set, type

  ```
  $ env
  ```

  You should see all the environment variables you just set. Ensure you see the required Twitter and Keen access keys are displayed with the correct values.

3.7 Run the Back-end Script

- Now you are ready to actually execute the script and begin listening for tweets. To run the script use the –q required flag to specify the name of the hashtag you wish to use in your query. (Replace the LIS2001 in the examples below with your hashtag)

  Use the –h flag for help with running the script

  ```
  $ cd
  /var/www/dashboard.socialmediadata.org/public_html/dashboard/LIS2001/scripts

  $ python create_stream.py –q LIS2001 (-c <config file location>) (-h)
  ```

  From there you will see the script running. It is listening for incoming tweets based on the hashtag you provided. It will not return control over to the shell.

3.8 Create Twitter Timeline Widget

- Part of the dashboard includes an embedded Twitter timeline widget to display the content of recent tweets. Follow these guidelines to create a Twitter search widget for your hashtag.

- Copy the generated HTML after creating the widget on the Twitter interface.

3.9 Modify the Front-end Dashboard Interface

Each dashboard front-end uses a single html file containing html code to structure the layout of dashboard panels. The front end html must be altered to collect content from the correct Keen.io stream for each query.

- In a new terminal, navigate to the location of the dashboard. (Replace LIS2001 to your hashtag)

```
% cd /var/www/dashboard.socialmediadata.org/public_html/dashboard/LIS2001
```

- First make a copy of the original dashboard file, then configure your dashboard point to the appropriate Keen.io stream for your hashtag. (Replace LIS2001 to your course hashtag. Also replace the Twitter Widget HTML in the place indicated)

```
% cp index.html index.html.old
% sed -i 's/COURSE_HASHTAG/LIS2001/g' index.html
% sed -i 's/COURSE_HASHTAG/LIS2001/g’ sent_sub.html
```

- Open the file index.html in a text editor

- In file replace the text, ‘PASTE_TWITTER_WIDGET_HERE’ with the HTML of the Twitter timeline widget you copied at the end of step 3.8

- Save and Exit the file

  The new dashboard is now running and ready to be used. Open the URL in a web browser to view the interface. (Replace LIS2001 to your course hashtag)

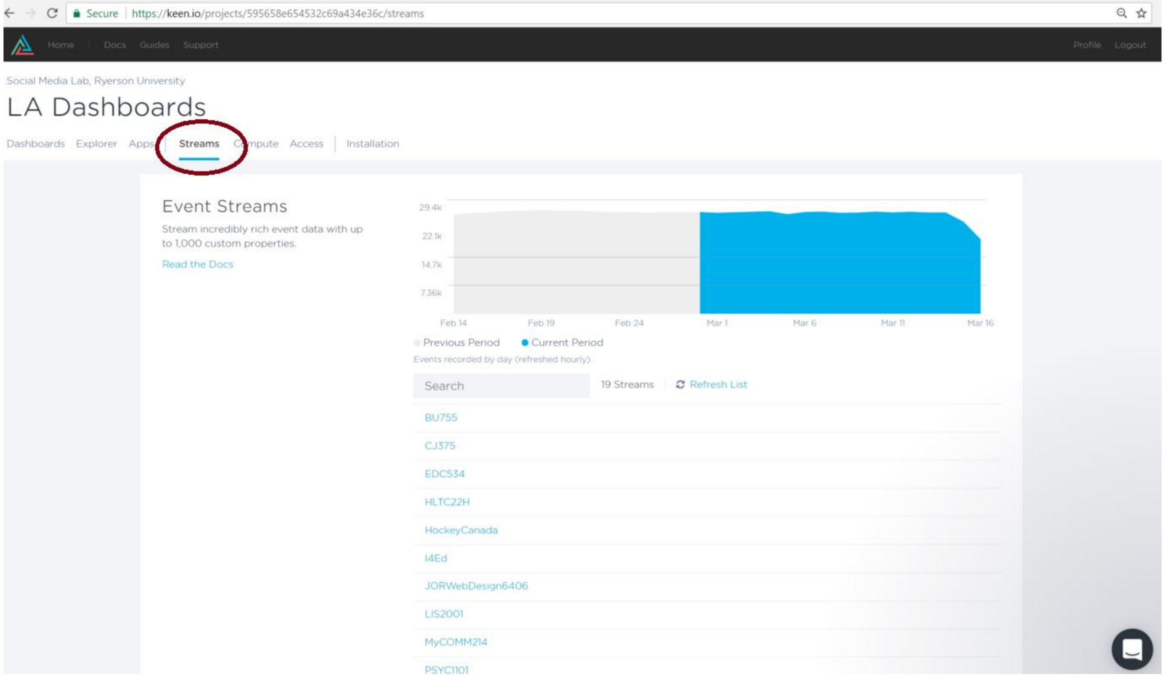
  `https://127.0.0.1/LIS2001/`

4.0 Testing the Dashboard

- To first test your dashboard, send some tweets! Log into your Twitter account and create a tweet to make sure the script will add it to your stream. Make sure to include the hashtag for the course.

- Log into your keen.io account https://keen.io/login

- Select your project name, then select the Streams tab for your project. In the list of streams you should see a stream with the same name as your hashtag.



Social Media Lab, Ryerson University

LA Dashboards

Event Streams

Stream incredibly rich event data with up to 1,000 custom properties.

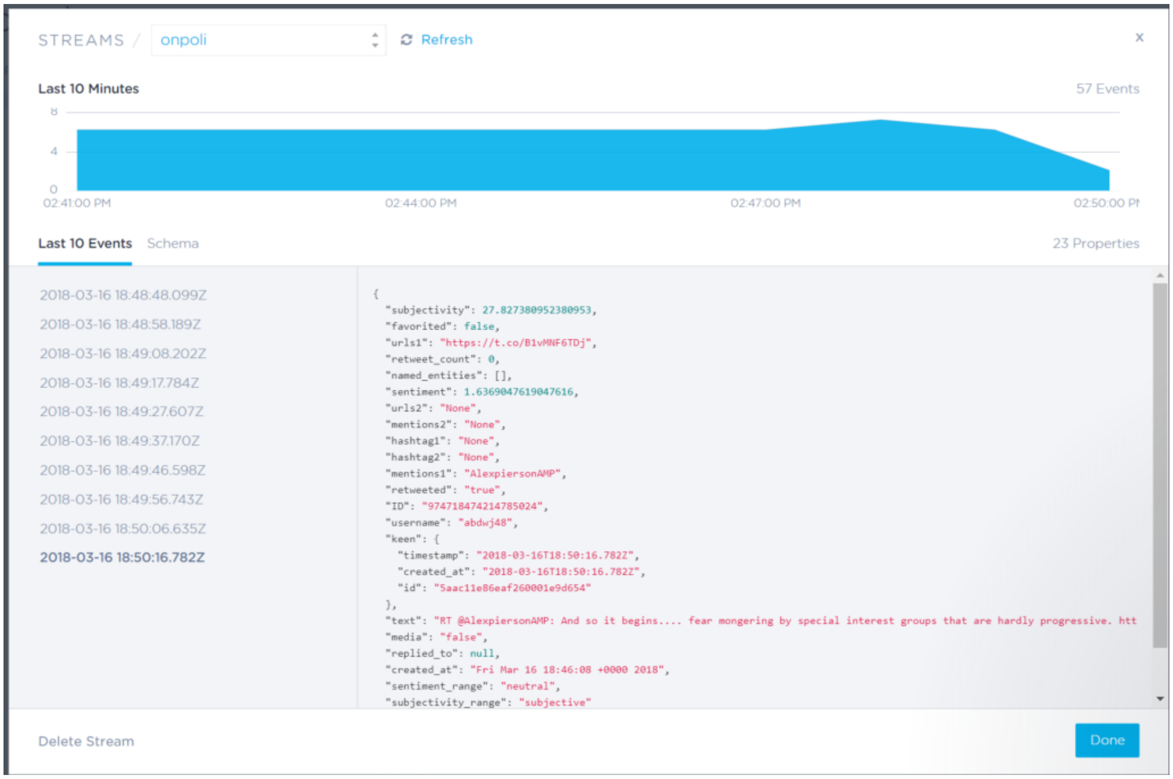
Read the Docs

Events recorded by day (refreshed hourly).

19 Streams

BU755
CJ375
EDC534
HLTC22H
HockeyCanada
I4Ed
JORWebDesign6406
LIS2001
MyCOMM214
PSYC1101

- Select the stream and you should see the tweet you just sent, with all the associated metadata including text, the hashtag, the user handle and other fields.



Last 10 Minutes

57 Events

23 Properties

2018-03-16 18:48:48.099Z
2018-03-16 18:48:58.189Z
2018-03-16 18:49:08.202Z
2018-03-16 18:49:17.784Z
2018-03-16 18:49:27.607Z
2018-03-16 18:49:37.170Z
2018-03-16 18:49:46.598Z
2018-03-16 18:49:56.743Z
2018-03-16 18:50:06.635Z
2018-03-16 18:50:16.782Z

```
{
  "subjectivity": 27.827380952380953,
  "favorited": false,
  "urls1": "https://t.co/B1vMNF6TDj",
  "retweet_count": 0,
  "named_entities": [],
  "sentiment": 1.6369047619047616,
  "urls2": "None",
  "mentions2": "None",
  "hashtag1": "None",
  "hashtag2": "None",
  "mentions1": "AlexpiersonAMP",
  "retweeted": "true",
  "ID": "974718474214785024",
  "username": "abdwj48",
  "keen": {
    "timestamp": "2018-03-16T18:50:16.782Z",
    "created_at": "2018-03-16T18:50:16.782Z",
    "id": "5aac11e86eaf260001e9d654"
  },
  "text": "RT @AlexpiersonAMP: And so it begins.... fear mongering by special interest groups that are hardly progressive. htt
  "media": "false",
  "replied_to": null,
  "created_at": "Fri Mar 16 18:46:08 +0000 2018",
  "sentiment_range": "neutral",
  "subjectivity_range": "subjective"
```

- Open the dashboard http://127.0.0.1/LIS2001 (with your hashtag replacing LIS2001) You should see the information about your test tweets displayed on the dashboard.

- If you want to delete the test tweets, you can select the Delete Stream button in the Streams tab. You can also log into twitter and delete the Tweets as well. Deleting the tweets through the Twitter interface will not affect what was saved on Keen.io. They have to be deleted separately.