# Offensive Language Detection – Project progress report

For our project we have decided to divide our entire project work into four basic modules,

- View
- Extractor
- Analyzer
- Repository

## View

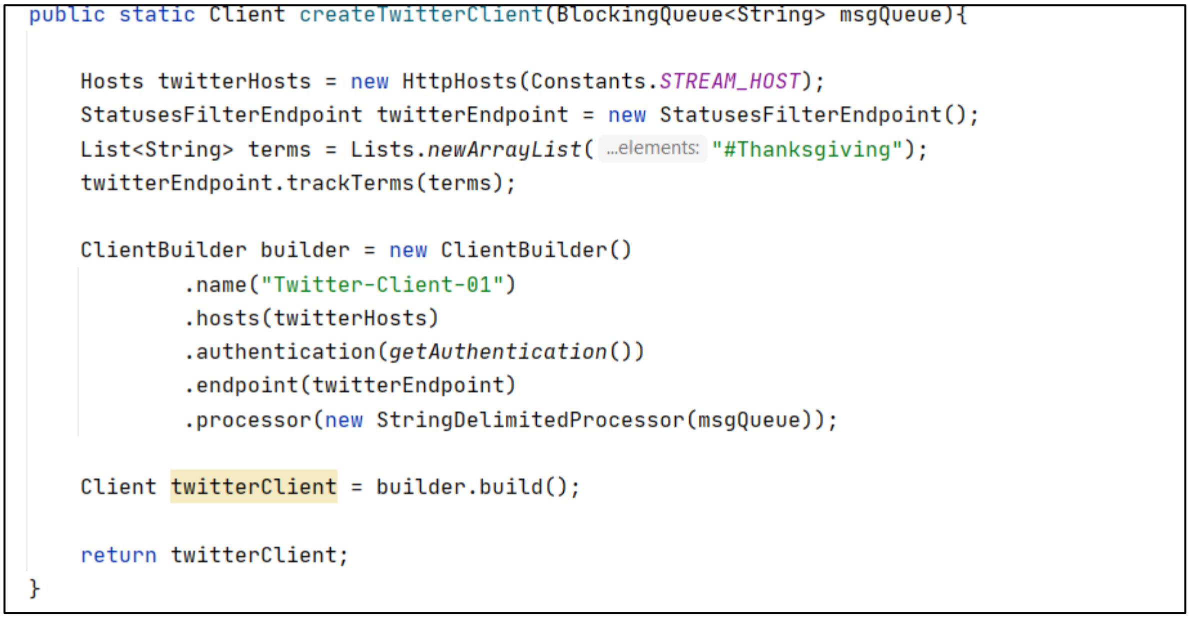
For view module we have decided on using react / dash combination, we may switch to static html webpage based on our needs, view will be used for two main reason, one from ui we will input what hashtags we want to search, that will be fed into our extractor system. Also, our ui will have separate dashboard, where will have visual elements of different datasets, like tweet counts, their overall sentiment, etc.

## Extractor

This module will be used to get the tweets for a list of hashtags, we have decided to use Java for this module, we have signed up for Twitter developer account, we are using twitter official hbc api for getting our tweets,

twitter/hbc: A Java HTTP client for consuming Twitter's realtime Streaming API (github.com)

We have been able to complete the coding of this module, and we were able to get tweets from api successfully for particular hashtags, please find code snippets below,



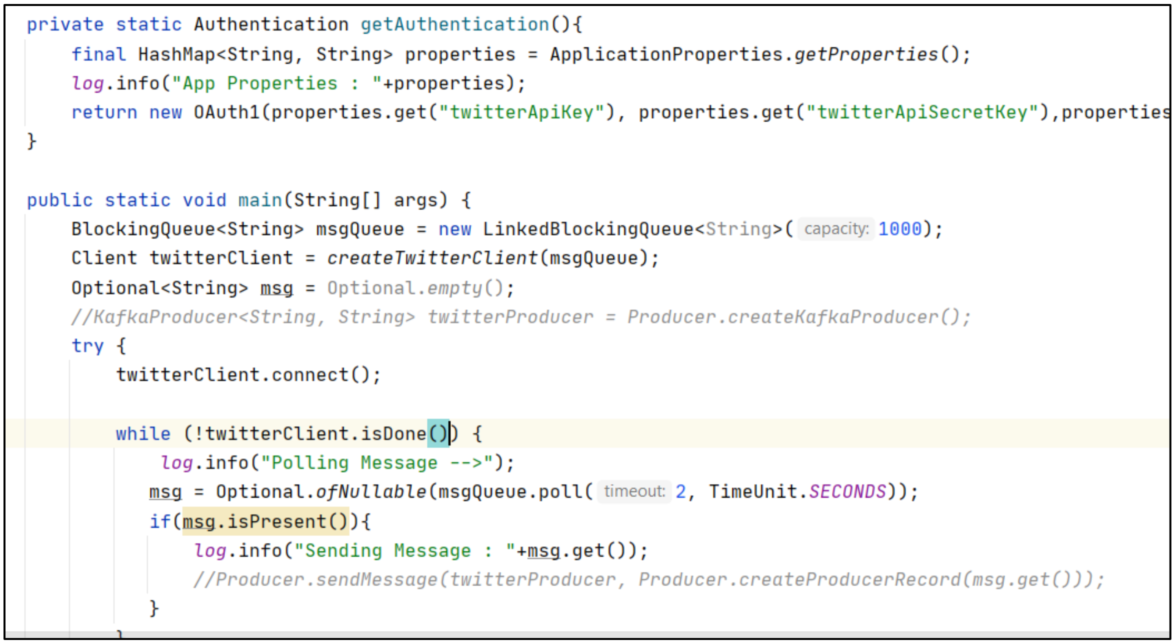
```java
public static Client createTwitterClient(BlockingQueue<String> msgQueue){

    Hosts twitterHosts = new HttpHosts(Constants.STREAM_HOST);
    StatusesFilterEndpoint twitterEndpoint = new StatusesFilterEndpoint();
    List<String> terms = Lists.newArrayList( ...elements: "#Thanksgiving");
    twitterEndpoint.trackTerms(terms);

    ClientBuilder builder = new ClientBuilder()
            .name("Twitter-Client-01")
            .hosts(twitterHosts)
            .authentication(getAuthentication())
            .endpoint(twitterEndpoint)
            .processor(new StringDelimitedProcessor(msgQueue));

    Client twitterClient = builder.build();

    return twitterClient;
}
```

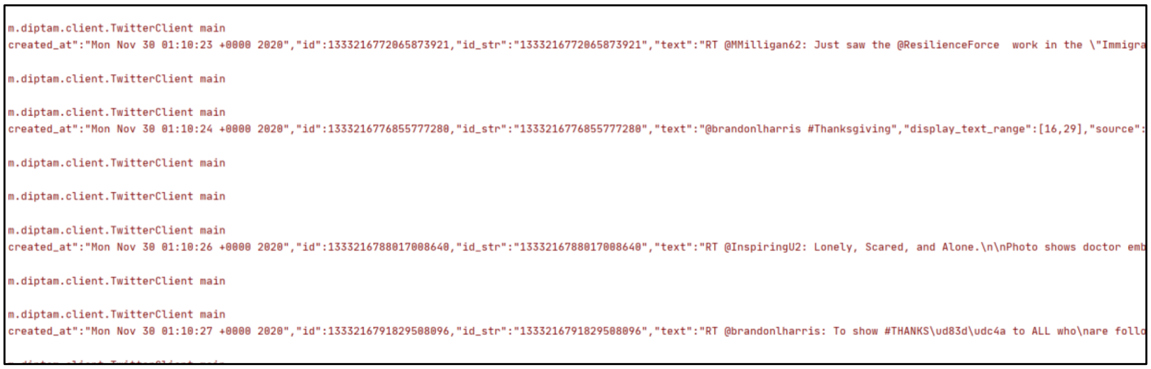


```
private static Authentication getAuthentication(){
    final HashMap<String, String> properties = ApplicationProperties.getProperties();
    log.info("App Properties : "+properties);
    return new OAuth1(properties.get("twitterApiKey"), properties.get("twitterApiSecretKey"),properties
}

public static void main(String[] args) {
    BlockingQueue<String> msgQueue = new LinkedBlockingQueue<String>( capacity: 1000);
    Client twitterClient = createTwitterClient(msgQueue);
    Optional<String> msg = Optional.empty();
    //KafkaProducer<String, String> twitterProducer = Producer.createKafkaProducer();
    try {
        twitterClient.connect();

        while (!twitterClient.isDone()) {
             log.info("Polling Message -->");
            msg = Optional.ofNullable(msgQueue.poll( timeout: 2, TimeUnit.SECONDS));
            if(msg.isPresent()){
                log.info("Sending Message : "+msg.get());
                //Producer.sendMessage(twitterProducer, Producer.createProducerRecord(msg.get()));
            }
```

And we are getting outputs like this,



```
m.diptam.client.TwitterClient main
created_at":"Mon Nov 30 01:10:23 +0000 2020","id":1333216772065873921,"id_str":"1333216772065873921","text":"RT @MMilligan62: Just saw the @ResilienceForce  work in the \"Immigra

m.diptam.client.TwitterClient main

m.diptam.client.TwitterClient main
created_at":"Mon Nov 30 01:10:24 +0000 2020","id":1333216776855777280,"id_str":"1333216776855777280","text":"@brandonlharris #Thanksgiving","display_text_range":[16,29],"source":

m.diptam.client.TwitterClient main

m.diptam.client.TwitterClient main

m.diptam.client.TwitterClient main
created_at":"Mon Nov 30 01:10:26 +0000 2020","id":1333216788017008640,"id_str":"1333216788017008640","text":"RT @InspiringU2: Lonely, Scared, and Alone.\n\nPhoto shows doctor emb

m.diptam.client.TwitterClient main

m.diptam.client.TwitterClient main
created_at":"Mon Nov 30 01:10:27 +0000 2020","id":1333216791829508096,"id_str":"1333216791829508096","text":"RT @brandonlharris: To show #THANKS\ud83d\udc4a to ALL who\nare follo
```

Now we are working on cleaning the tweets, so that we can use them directly to our analyzer module without any deformed text

## Analyzer

We have decided on using Python for this module, there will be python script running in background, where we will feed our cleaned tweets, and we will scan for offensive words in that text and mark that tweet accordingly, also we are planning for doing sentiment analysis of the tweets, we are still deciding on that topic.

## Repository

We plan on storing all our analyzed tweets on mongoDB on cloud, so that we use data from our UI component.