Airport Tweets Application

This document contains the design and information about the airport tweets application.

# Objective

Develop a JAVA service that can run on a scheduled interval to hit a twitter API for number of airport accounts, download and store any new tweets so that another service could query that database and be provided with the latest tweets for any given airport.

# Design

The application is created over Spring Boot leveraging over Maven and MySQL database. The architecture of AiportTweetApplication can be understood by the following diagram:

## 2.1 Application Architecture Flowchart

**AirportTweetServiceManager.java**

**Twitter Authentication.java**

**Twitter Api**

Calls GetTweets()

**AirportTweetController.java**

**AirportTweetSchedular.java**

**MySQL DB**

(schema: airport tweets)

**Airport**

Contains airport account names

**AirportTweetDao.java**

Insert tweets

**Airport\_Tweet\_Activity\_Log**

Stores logging info

**Airport\_Tweet**

Stores Tweet Data

*AirportTweetApplication Process Flowchart*

## 2.2 Steps of Process Flow

1. On Execution of Java Service, AirportTweetController.java starts the application and the scheduler runs as per the mentioned cron expression in application.properties file.
2. After the Scheduler runs, the AirportTweetServiceManager retrieves the list airport account names from Airport table in DB for which the tweets must be retrieved.
3. Following this after the Twitter Authentication, we call Twitter api to retrieve all the status for all the accounts provided in the airport table. The retrieval of tweets from twitter is divided in two steps further:
4. On first call, it retrieves all the last 20 tweets (where no of tweets is configurable) for each airport.
5. After the first call it fetches only the updates tweets from the twitter using the last call’s twitter ID.
6. After fetching the records, map them to AirportTweet.java entity and insert them through DB.
7. Insert all the logging info, and error info in Airport\_Tweet\_Activity\_Log Table for logging.

## 2.3 Service Monitor & Configuration

Monitoring Services should be provided to all applications with a view to reducing the time, cost and risks of development through re-use. For this, the following services within the application have been provided:

### **2.3.1 Application Configurable**

All the properties like service running frequency and the MySQL DB credentials and twitter credentials are configurable.

* application.properties: contains DB credentials and application configurable properties.
* twitter.properties: contains twitter access tokens

2.3.2 Performance monitoring and Logging

Through proper logging with in the application and the service monitoring through Airport\_Tweet\_Activity\_Log Table.

### Error Handling

There has been error handling and error logging with in the application.

## 2.4 Software development tools

The Software Development Tools used with in the application are:

1. Eclipse IDE
2. MySQL Database
3. Reference of spring.start.io to build the maven Project with Spring boot.

## 2.5 Utility by Another Service

There is a column in Airport\_Tweet Table **is\_Tweet\_new**, so as mentioned in the requirement document that these records of airport tweets will be used by another service, so that service can have the check on this column which will help to fetch the latest records and can later mark it as flag false after sending them for further notification.

# Future Action Items

These are the future steps that should be taken as per the objective:

## 3.1 Data cleaning

Clean data like emoticons, special characters and other redundant information from the data.

## Data Purging/Data Analysis

Data Purging: Remove the tweets from the airport\_tweet table every week deleting the records older more than three months.

Data Analysis: Use this data for further data mining or data analysis purpose.