

# JPE\_Twitter

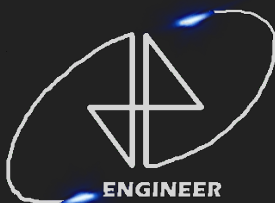
Splunk App for Twitter Mining

This document was made, designed and wrote uniquely to general understand of JPE\_Twitter application, so it can't be edited and distribute whitout the written authorization of the author.

---

May 2019

Juan Alejandro Pérez Chandía  
Computer Engineering  
jalejandro.ingeniero@gmail.com



JPEngineer ©

## 1 Introduction

JPE\_Twitter is a Splunk Application developed in Python, whose purpose is the Twitter Mining and complete integration with the log analysis tool previously mentioned. With JPE\_Twitter you'll can extract of Twitter important information based on words filter, phrases, hashtag, among other and get data indicate what's happening with our business or service on Twitter, building a development base to sentiment analysis on Twitter.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Basic concepts of development</b>	<b>4</b>
2.1	Dashboards . . . . .	5
2.1.1	Token and Security . . . . .	5
2.1.2	Filter . . . . .	6
2.1.3	Service . . . . .	6
2.1.4	Twitter Users . . . . .	7
2.1.5	Twitter Hashtag . . . . .	7
2.1.6	Twitter Comments . . . . .	8
<b>3</b>	<b>Instalación</b>	<b>9</b>
<b>4</b>	<b>Validaciones</b>	<b>10</b>
<b>5</b>	<b>Soporte, consultas o comentarios</b>	<b>10</b>

## 2 Basic concepts of development

The application is developed in Python 2.7, with the purpose of sharing the Python base installation of the Splunk platform (6.x or higher) and has five essential modules:

- **config** : The application has a configuration file called *twitter.conf*, where config module can read and write the necessary parameter to the authentication with Twitter API, log files configuration, encrypt, among others.
- **security** : For the security of confidential data (authentication token), this module has what it needs to encrypt and decrypt, based on the key that the user defines in the configuration of the tokens in Splunk.
- **splunk\_read** : The purpose of this modules is take all parameters entered from the Splunk platform and take them to the corresponding module (config, security, service, among others).
- **service** : Responsible to start, stop and restart the service, besides to obtain its status.
- **twitter** : Corresponding to the main module, who will return everything filtered on Twitter.

Besides, JPE\_Twitter has its own log files, [for | to] make process debug through Splunk or directly looking at log files:

- **config.log** : Show the ran process by *config* module, including its problem (if there are any).
- **service.log** : Show the service state and its PID, besides include problems that this service could have.
- **twitter.log** : Show the authentication state with Twitter API and filtered messages from this social network.
- **splunk\_read.log** : Show the ran process by *splunk\_read* module, including its problem (if there are any).

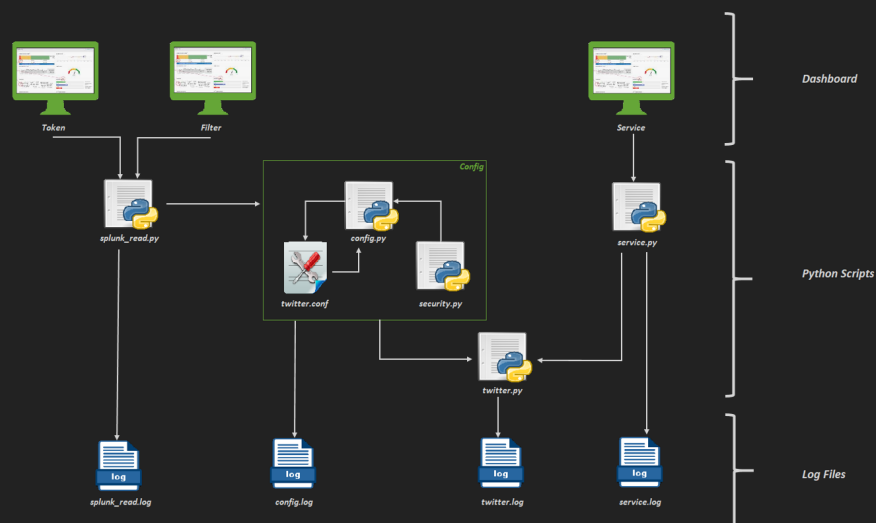


Fig. 1: JPE\_Twitter General Diagram

**Important:** All necessary libraries to proper running of application, is includes on the installation package.

## 2.1 Dashboards

For administration of Twitter listener process it exists three important dashboard, that allow configure the authentication tokens and filters, besides allow start, stop and restart service or verificate its status:

- Token and Security
- Filter
- Service

And also have with three basic dashboard, that allow show Twitter general information base on the filters configured, such as user with most followers, hashtag the most used and users mention:

- Twitter Users
- Twitter Hashtag
- Twitter Comments

### 2.1.1 Token and Security

This dashboard allow to configure the authentication tokens of Twitter API, which are: *consumer key*, *consumer secret*, *access token* and *access secret*. This tokens are stored in configuration file *twitter.conf* and security concern, the authentication tokens can be encrypted to prevent that other people access to them, only is necessary to write an encryption key in the same panel, after to indicate that you want to encrypt the tokens.

It is important consider that: to obtain tokens, it must create a developer account on Twitter (<https://developer.twitter.com>)

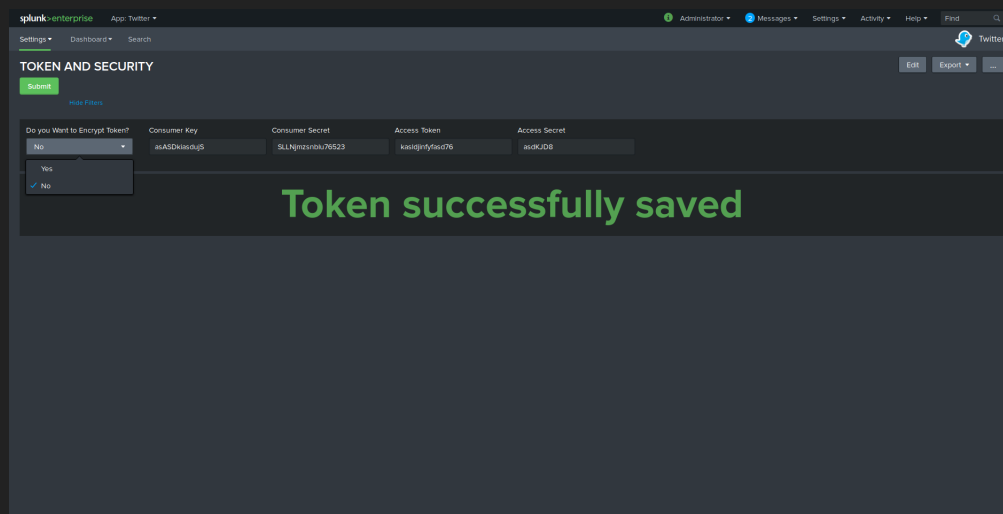


Fig. 2: Setting: Token and Security Dashboard

//developer.twitter.com)

### 2.1.2 Filter

Este dashboard permite configurar los filtros que se aplicarán en el proceso de escucha. Las opciones de configuración incluyen la acción de agregar nuevo filtro al que se encuentra actualmente configurado o configurar nuevos filtros reemplazando a los anteriores, mediante la elección de *add* o *new* respectivamente. Los filtros ingresados pueden ser palabras, frases, hashtag, entre otros y deben ser separados por una coma.

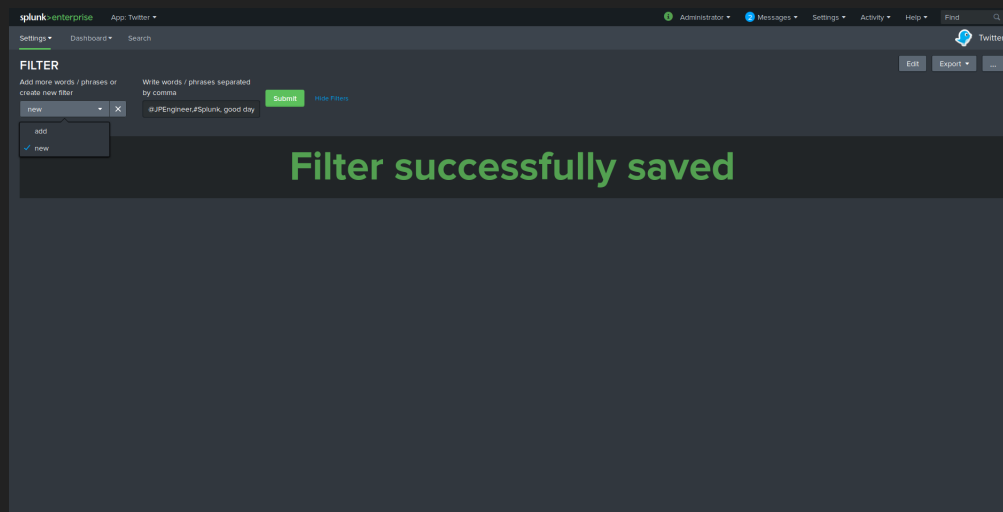


Fig. 3: Setting: filter Dashboard

### 2.1.3 Service

Este dashboard permite ejecutar acciones sobre el servicio de Twitter. La opción Start permite iniciar el servicio y comenzar a indexar data obtenida desde Twitter y como requisito previo, se requiere tener configurado los Token de autenticación y los filtros. La opción Stop es la encargada de detener el servicio. La opción Restart permite reiniciar el servicio si es requerido por desarrollos propios u otras razones. La opción Status indica si el servicio se encuentra activo o detenido.

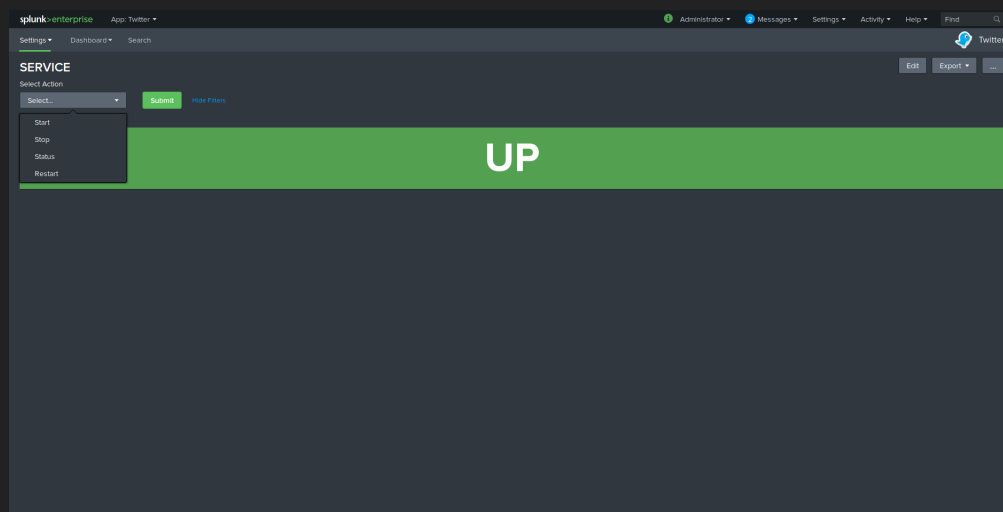


Fig. 4: Setting: Service Dashboard

## 2.1.4 Twitter Users

Este dashboard presenta información relacionada a usuarios de la red social, tales como: cantidad de seguidores, nombre, alias y ubicación. Sobre el mismo dashboard se pueden aplicar desarrollos para obtener información adicional.

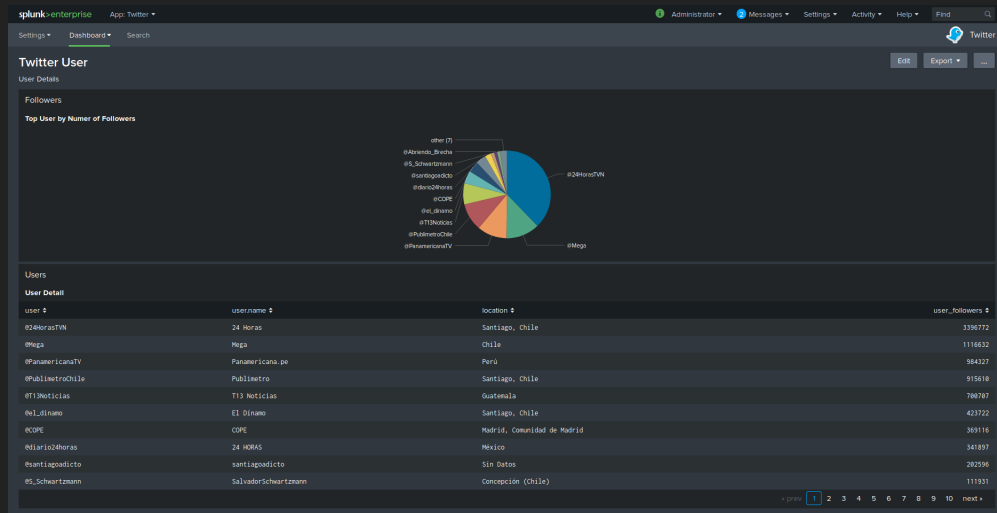


Fig. 5: Twitter User Dashboard

## 2.1.5 Twitter Hashtag

Este dashboard presenta información relacionada a los hashtag más usados durante el día. Sobre el mismo dashboard se pueden aplicar desarrollos para obtener información adicional.

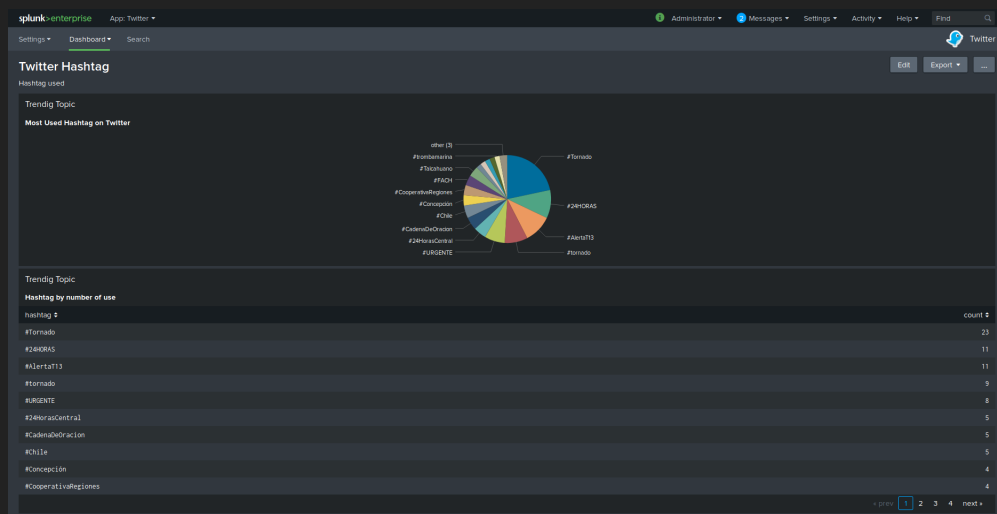


Fig. 6: Twitter Hashtag Dashboard

## 2.1.6 Twitter Comments

Esta dashboard presenta el detalle de los usuarios más comentados del día, comentarios realizados e información del usuario. Sobre el mismo dashboard se pueden aplicar desarrollos para obtener información adicional.

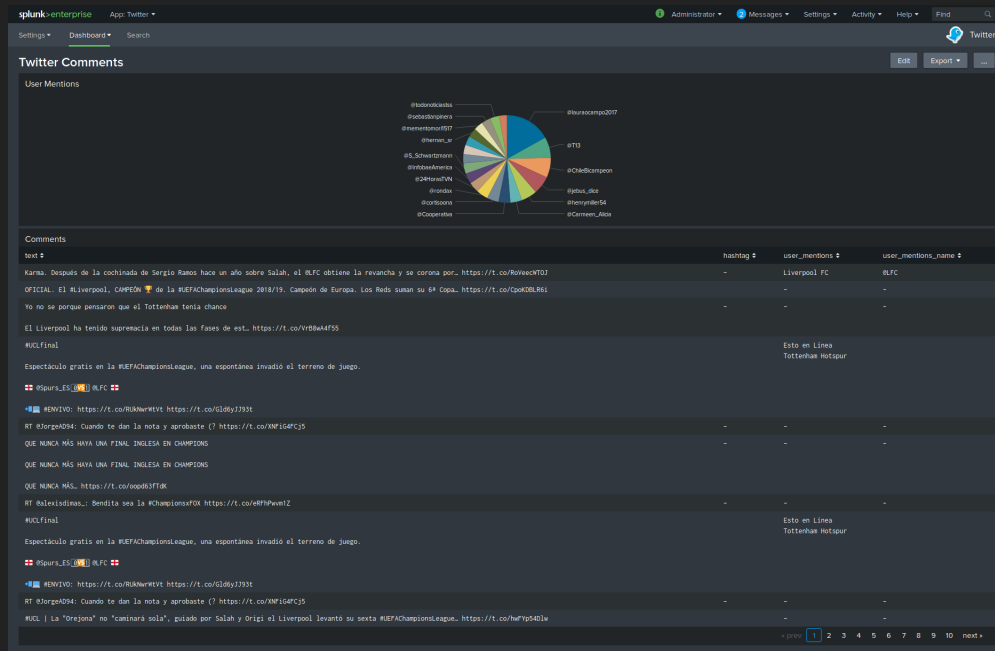


Fig. 7: Twitter Comments Dashboard

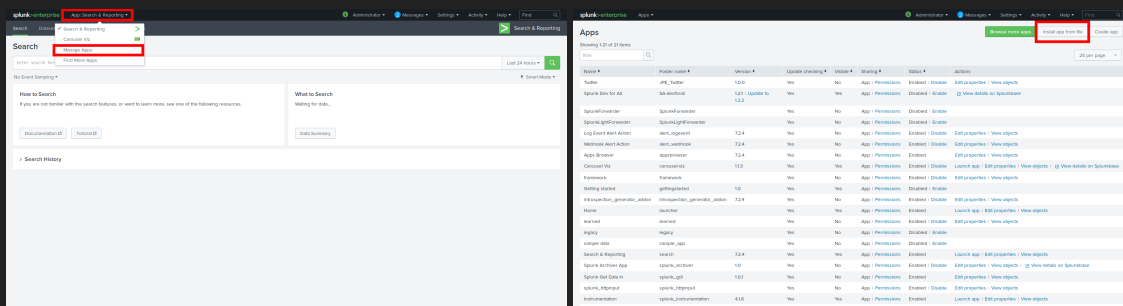


### 3 Instalación

JPE\_Twitter esta probado en versiones 6.x de Splunk o superior y fue desarrollado en Splunk 7, el cual será usado en la guía de instalación.

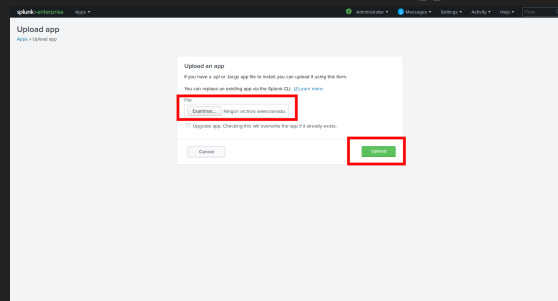
Considerando que la aplicación ya se encuentra descargada, debes seguir los siguientes pasos:

- Hacer click en el apartado superior izquierdo de la plataforma Splunk, que dice **App: Search & Reporting > Manage Apps**
- Hacer click en el extremo superior derecho de la plataforma, en el botón **Install Apps from File**
- Hacer click en **Examine** y seleccionar JPE\_Twitter.zip, luego hacer click en **Upload**



(a) Manage Apps

(b) Install Apps from File



(c) Upload

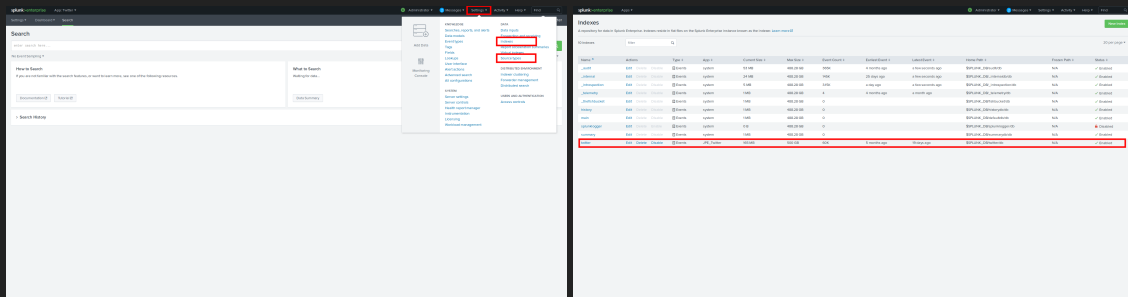
Fig. 8: Instalación de la aplicación JPE\_Twitter

Con estos simples pasos, JPE\_Twitter se encontrará instalado. La recomendación pos-instalación es reiniciar Splunk.

## 4 Validaciones

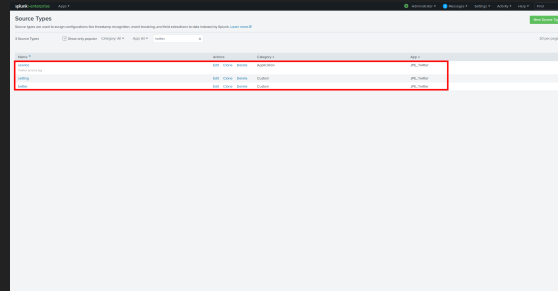
Para la validación de la instalación se deben seguir los siguientes pasos:

- Hacer click en el apartado superior derecho de la plataforma, en la sección **Setting > Indexes**
- Validar que exista el index **twitter**
- Hacer click en **Setting > Source Types** y validar que existan los source types **twitter**, **setting** y **service**



(a) Setting: Index - Sourcetype

(b) Index



(c) Sourcetype

Fig. 9: Validaciones

**Index twitter:** Es el repositorio de datos del aplicativo instalado.

**Source Type twitter:** Identifica la estructura de los datos filtrados de Twitter.

**Source Type service:** Identifica la estructura de los datos generador por el módulo *service.py*

**Source Type setting:** Identifica la estructura de los datos generador por los módulos *twitter.py* y *security.py*

## 5 Soporte, consultas o comentarios

**Nombre:** Juan Alejandro Pérez Chandía

**E-Mail:** jalejandro.ingeniero@gmail.com

**Idioma:** Español - Inglés