Práctica: Tika

Introducción

Apache TIKA, una herramienta que nos permite detectar y extraer metadatos y texto estructurado de varios tipos de documentos, entre los que podemos encontrar:

HTML
XML y derivados
.doc, .xls, .ppt (formatos de documentos de Microsoft Office)
.odt (Formato OpenDocument de OpenOffice)
Formatos comprimidos (ar, cpio, tar, zip, gzip, bzip2 y zip)
.pdf (Portable Document Format)
.epub (formato para libros electrónicos)
.rtf (Rich Text Format)
.txt (detectando el juego de caracteres)
Audio (.mp3, .mid, .wav, …)
Imagen (.jpeg, …)
Correos en formato mbox
Video (.flv)
Ficheros Java

Además, si nos descargamos el código fuente del programa, podremos crear nuestras propias clases y métodos en Java por si queremos añadirle alguna funcionalidad más. Solo tendríamos que compilarlo todo y generar el .jar correspondiente.

En nuestro caso, nos descargaremos ya el archivo .jar y trabajaremos con la funcionalidad que nos ofrece Tika.

Curiosidades

Existe una marca de patatas en chile que tienen el mismo nombre [link].

Requisitos

Al estar programada en java, esta herramienta se puede usar en cualquier sistema operativo.

1. Tener configurado Java en el equipo
2. Descargar la herramienta http://tika.apache.org/download.html

Instalación

Se aconseja el uso del archivo .jar, aunque también se puede descargar el archivo fuente y compilarse de forma normal.

Una vez descargado el archivo jar, hay que colocarlo en una carpeta a la que podamos acceder fácilmente ya que vamos a tener que usar la terminal.

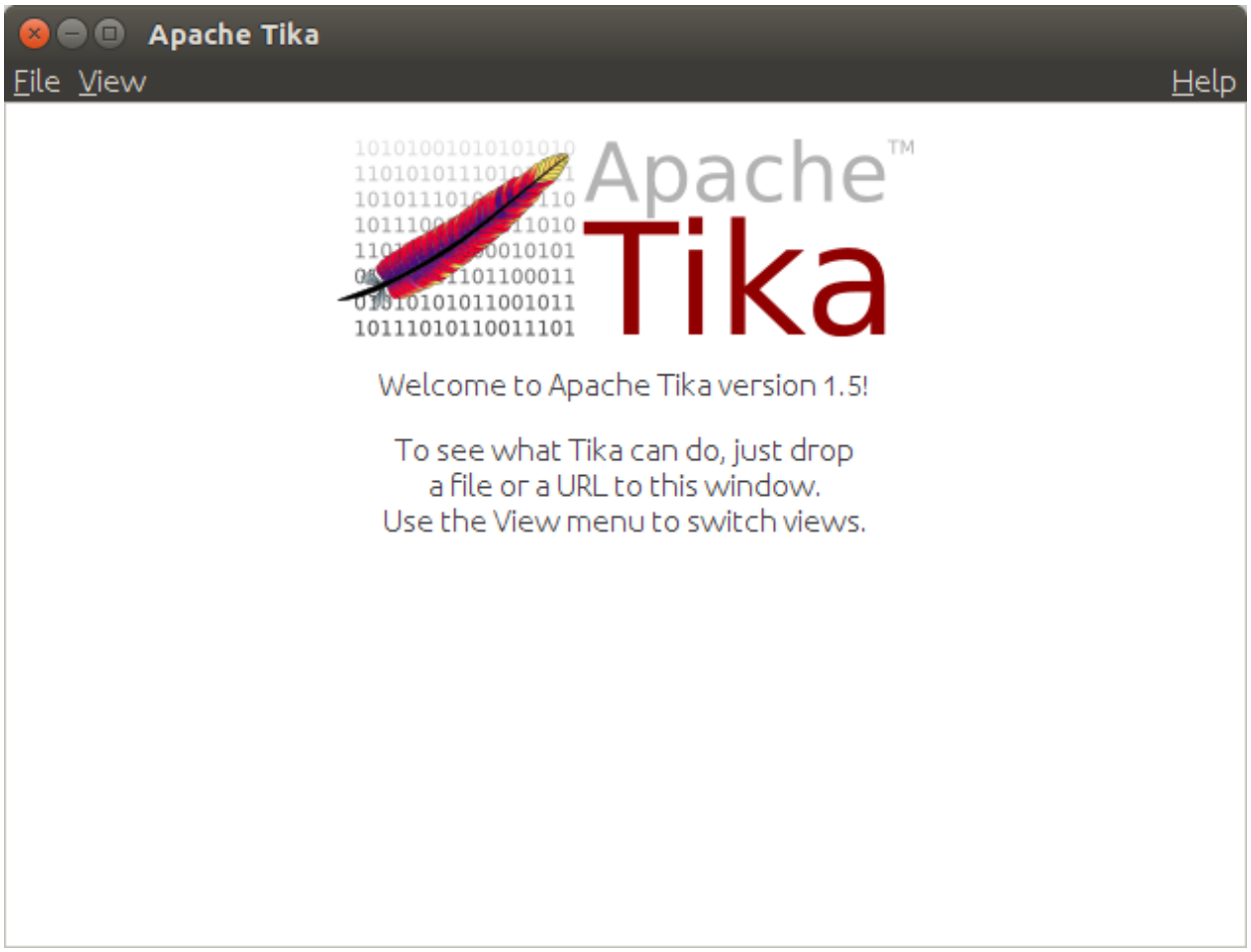
Para acceder a Tika se ejecutara el siguiente comando:

```
java -jar tika-app-X.X.jar (siendo X.X la versión de Tika)
```

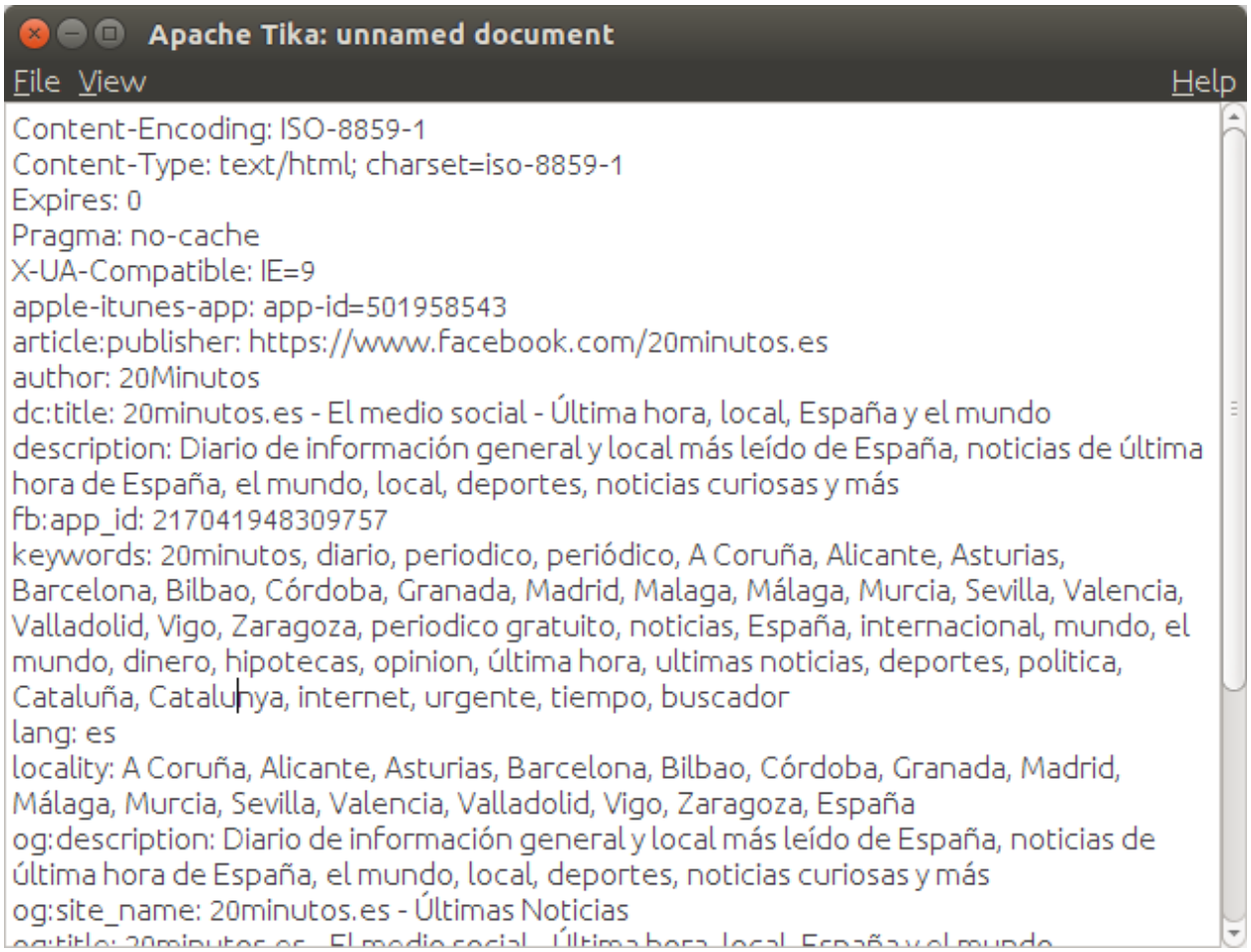
Así accedemos al modo ventana de Tika. Para acceder al modo consola se deben añadir más parámetros que veremos a continuación.

Modo ventana

Para el primer ejemplo usaremos el modo ventana, para ello vamos a utilizar en la terminal el comando visto justo en el apartado anterior y nos saldría la ventana de Apache Tika:



Pulsando en File podemos seleccionar el tipo de archivo de entrada del que extraeremos los datos, en este caso una URL (Open Url) y escribimos/copiamos la url de la web que deseamos descargar, p.e. http://www.20minutos.es/



En la pestaña View podemos acceder a los distintos tipos de información, metadatos, html…

Modo consola

En este ejemplo vamos a usar el modo consola, para ello primero vamos a darle un alias al comando que se repetirá, así nos ahorramos escribir siempre la misma parte de la llamada:

```
alias tika="java -jar tika-app-X.X.jar"
```

Para acceder a los mismos datos que hemos visto en el modo ventana nos bastaría con la siguiente orden;

```
tika http://www.20minutos.es/
```

El problema es que nos mostrará todos los datos en la misma consola (metadatos, xhtml, contenido).

Para evitar esto hay que seleccionar el tipo de información que deseamos, añadiendo al comando una de las siguientes opciones:

```
--metadata
--text
--xml
--html
--json
--text-main
--languaje (detecta el tipo de idioma)
--detect (detecta el tipo de documento)
--enconding=X (nos muestra el archivo con la codificación X)
```

Quedando:

```
tika --metadata http://www.20minutos.es/
```

Esto nos mostraría en la consola los metadatos de la página web.

La herramienta nos proporciona una funcionalidad para el almacenamiento de los datos que hemos obtenido. Para ello, guardaremos la información obtenida en el formato que prefiramos. Es más, la herramienta nos da la opción de traspasar información entre diferentes formato, por ejemplo, de un archivo .pdf a un .txt.

Si queremos almacenar el código html de una página en un archivo html haríamos lo siguiente:

```
tika --html http://www.20minutos.es/ > minutos.html
```

Se nos creará un archivo en la carpeta en la que nos encontremos con el contenido de la web en html. Cuando lo probemos, veremos que el estado en el navegador no es exactamente igual que a web original, esto es debido a que nos descargamos el html de la web, pero no el css.

Si queremos almacenar el contenido principal de la web sería:

```
tika --text-main http://www.20minutos.es/ > minutos.txt
```

Si queremos hacer lo mismo sobre un archivo local, solo hay que cambiar la dirección http por la dirección local del fichero.

Ejercicios

Describe el procedimiento de cada uno de los ejercicios siguientes:

1. Abrir un archivo pdf en el modo ventana y decir quién es el autor del archivo, si tiene.
2. Almacenar el contenido de un archivo .pdf en un archivo .doc.
3. Ver los metadatos de un archivo que esté subido en una página web.
4. Comprimir un archivo de texto sencillo y abrir con Tika el archivo comprimido.
5. Pasar por correo y Whatsapp una foto y comparar los metadatos comprobando las diferencias.

6. Ver los metadatos de http://www.uca.es/es/ y guardarlo en un archivo.txt.
7. Pasar un archivo .rdf a .doc. ¿Pasar el archivo .doc o .rdf al formato .pdf dará error?
8. Descargar las 3 imágenes que se proporcionan en la carpeta "Material Tika" y decir qué imagen (o imágenes) se sacó (sacaron) con un producto perteneciente a apple computer inc.
9. Describir el procedimiento seguido para guardar el contenido de una web cualquiera en un archivo html, este convertirlo en doc y comprobar los metadatos de este último.
10. Descargar tres imágenes, a elección del alumno, de tres sitios diferentes donde los usuarios compartan imágenes como pueden ser: facebook, instagram, flickr, twitter... y comentar las diferencias que encontramos en los metadatos.
11. Se puede trabajar con Tika desde Eclipse. Describe los pasos que han de realizarse para poder crear un proyecto Tika en Eclipse en el que se extraiga el contenido de un fichero PDF.

Referencias

http://tika.apache.org/
Chris A. Mattman, Jukka L. Zitting (2012) Tika in Action. Manning Publications Co.