

MUSIC API

David Blanco Fuentes
Ildefonso de la Cruz Romero
Miguel Ángel Pérez García
Oscar Delgado Miranda

24 de mayo de 2015

1. Introducción

En este proyecto trabajamos con las Apis de Soundcloud, Dropbox y Twitter, así como con los lenguajes python, css, html y javascript.

En este proyecto lo que hacemos es dar una funcionalidad a una página web utilizando la nueva api que hemos creado (MusicApi) para que podamos realizar una búsqueda a través de Soundcloud, ya sea de un artista, un álbum o una canción, y que, en la misma página, nos muestre los resultados de las búsquedas, en función de si queremos escucharla vía streaming y/o descargarla, así como compartir en Twitter y poder subir a la nube usando Dropbox.

Toda la música que nos proporciona la api de Soundcloud respeta los derechos de autor.

Este proyecto es ideal para llevarlo a un servidor que oferte la web al público, pero debido a la falta de tiempo, no se ha podido llevar a cabo.

2. Requisitos previos

Para realizar nuestro proyecto, necesitamos de unos requisitos previos:

1. Soundcloud. Necesitamos tener instaladas las librerías de Soundcloud: sudo pip install soundcloud.
2. Dropbox. Necesitamos tener instaladas las librerías de dropbox: sudo pip install dropbox.
3. El directorio donde se almacenen las descargas, deberá ser el directorio descargas(/home/user/descargas/) siendo user el usuario del pc. Esta carpeta funciona como lugar de almacenamiento de las canciones descargadas.

3. API Soundcloud

Esta api contiene las funciones necesarias para:

BuscarStream(def Buscar stream(búsqueda,nombre,download = False)). Cuando la variable download está a true, se muestran las urls de streaming de las canciones descargables. Obtiene una lista de los enlaces de streaming. El contenido

de las listas nombre y url coinciden con el nombre del track y su enlace. Nombre debe ser una lista vacía.

BuscarDescargas(def Buscar Descargas(búsqueda, nombre)).Obtiene una lista de los enlaces de descarga. El contenido de las listas nombre y url coinciden con el nombre del track y su enlace. Nombre debe ser una lista vacía.

Para utilizar esta Api, necesitaremos tener creada tanto la librería en python(sudo pip install soundcloud) de Soundcloud, como una cuenta en Soundcloud, y una app de la misma(esto es importante, pues sin las claves no podríamos hacer nada).

4. API Dropbox

Esta api contiene las funciones necesarias para:

GuardarMúsica(def Guardar Music()).Guardar toda la música que se haya descargado, y se encuentre en el directorio de descarga.

AutorizarDropbox(def Autorizar dropbox()).Generar la url para poder autenticar al usuario

Al igual que con Soundcloud, necesitaremos tener creada la librería en python(sudo pip install dropbox) de Dropbox, una cuenta y una app developer de la misma, además de necesitar tener los permisos de full dropbox al crear la app.

Para poder usar su cuenta privada de dropbox para guardar las canciones, es necesario cambiar las credenciales del fichero Auth.py, por esta razón se necesita la app developer para la propia cuenta.

El directorio donde se almacenen las descargas, deberá ser el directorio descargas(/home/user/descargas/) siendo user el usuario del pc. Esta carpeta funciona como lugar de almacenamiento de las canciones descargadas.

Nota: la funcionalidad de las funciones que se han implementado a partir de la api de Dropbox sólo tienen en cuenta si se ejecuta en una máquina con Linux, y que la carpeta personal de la misma no esté cifrada.

5. Twitter

Tanto en descargas como en streamings, todas las canciones te permiten compartirla en Twitter.

6. MusicApi

Es la app principal a la cual, le damos la funcionalidad de poder compartir, escuchar, descargar, subir a dropbox remixes de nuestras canciones favoritas, donde tenemos las siguientes funciones: -def Vacía all() -def Vacía stream() Evitamos la aparición de errores en la búsqueda, y no haya nada en el cuadro de búsqueda

-def buscar(name). En esta función, exponemos a modo de lista todas las canciones descargables, además del streaming de dicha canción. Dentro de la función, obtenemos la url para la autenticación de dropbox, además de obtener las listas con las urls de streaming y descarga. En el caso de que no se encuentren resultados, nos muestra un mensaje. En caso contrario, nos mostrará la lista de

las canciones, así como las opciones de descarga. -def only stream(name). Expone sólo el streaming de todas las canciones a modo de lista. En el caso de que no se encuentren resultados, nos mostraría un mensaje de "no encontrado". -def Guardar dropbox() Esta función sube a dropbox la música descargada, y llama a otro html en caso de éxito. -def index() Esta función devuelve el html de la página principal.

7. Auth

En este fichero están todas las claves necesarias, tanto de Soundcloud como de Dropbox. Para utilizar tu propia app de Soundcloud o de Dropbox, tendrás que sustituir estas claves por las proporcionadas por tus apps.

8. Página web de pruebas

En esta página web probamos todas las funciones realizadas por nuestra api.

9. Manual de Pruebas

1. El programa python principal es el MusicApi.py, que es el que debemos ejecutar. Como consecuencia de su ejecución, nos dará un enlace, que será el de la página web donde se realizan las pruebas.
2. Una vez que estamos en la página web, veremos un cuadro de búsqueda, donde introduciremos el nombre del artista, canción o álbum que queramos. Debajo del cuadro de búsqueda, tenemos 2 opciones para realizar dicha búsqueda, que son:

- a) Buscar: si seleccionamos esta opción, nos aparecerá una lista de canciones que podremos escuchar con el reproductor que aparecerá, descargar a partir del botón de descarga, y compartir en Twitter, mediante el botón de Twitter, si estás escuchando una canción, o si te has descargado una canción. En el caso de que estés escuchando una canción, el tuit estará acompañado del enlace de reproducción. Si por el contrario, has descargado una canción, y decides compartirlo, el tuit aparecerá con el enlace de descarga.

Al descargar una canción, tenemos la opción de subirla a Dropbox. Para ello, seleccionaremos la opción de autenticar Dropbox. Esta opción nos abrirá una pestaña, donde tendremos que aceptar e iniciar sesión en Dropbox, si no teníamos una sesión abierta. Luego, volvemos a la página, y le damos a subir a Dropbox. Esto creará una carpeta en Dropbox llamada Music con la música que te hayas descargado. Dependiendo de lo que descarguemos, el tiempo de creación será mayor o menor. Es recomendable no tener más canciones en la carpeta de descargas.

- b) Buscar streaming: seleccionando esta opción, nos aparecerá una lista de canciones que podremos escuchar mediante el reproductor. Al igual que en la anterior opción, también podemos compartir en Twitter.

10. Evolución del proyecto

En esta parte se explicará como ha ido evolucionando el proyecto, desde la primera versión hasta la última.

Fecha	Actividad
5/05/15	Reunión y exposición de ideas
8/05/15	Reparto de tareas
12/05/15	Unión de todas las tareas e intento de montaje
14/05/15	Corrección de errores y últimos añadidos
17/05/15	Documentación de código y proyecto
18/05/15	Subida de trabajo a github

Fecha	Actividad
9/05/15	api soundcloudv1: definición de funciones básicas
10/05/15	prueba.html estado primario de la web; api dropbox.py funciones primarias y auth.py
12/05/15	desarrollo web. Carpeta con el contenido html
14/05/15	apiv1 versión primaria de la app principal. Enlaces entre soundcloud y dropbox
16/05/15	apiv2 versión enlazada con la web
17/05/15	music api.py api principal y enlace con la web
18/05/15	subida de la api definitiva

11. Responsabilidades del grupo

1. David: documentación del proyecto y de la utilización de la api Twitter.
2. Ildefonso: estructuración de los ficheros y la utilización de las api de Soundcloud y de Dropbox.
3. Miguel Ángel: colaboración con el desarrollo de la página web y utilización de la api de Twitter.
4. Óscar: desarrollo de la página web completa y colaboración con Dropbox.

12. Problemas Apis

Soundcloud:

1. Se ha tenido que hacer a mano la URL para obtener los enlaces tanto de descargas como de streaming, debido a un error en en lo que se cree de la API de soundcloud, debido a que no devuelve la URL formada ya con el id de usuario.
2. Soundcloud no termina de ser claro con lo que devuelven sus funciones de los tracks, y lo que se creía como un fichero JSON Soundcloud lo transforma automaticamente en un objeto, motivo por el cual no se ha usado JSON en el trabajo.
3. No ha sido posible imbuir el widget de Soundcloud en la web, en su lugar la previsualización de la música se ha realizado a mano los códigos HTML correspondientes.

Dropbox:

1. A pesar de que en la documentación oficial nos enseña a autenticar una sesión, debido al proceso que se debe realizar, algo nada cómodo para el usuario, se ha optado por hacerlo más fácil rápido y sencillo, lo que ha supuesto una mayor labor de investigación y siendo casi inservible la documentación oficial.
2. La función para obtener un JSON con el metada necesario para saber que contenido que hay en dropbox no se ha podido llevar acabo debido a que esta función devolvía vacío el campo contents, el cual contenía información sobre los directorios de este, no siendo real su resultado. Esto ha provocado que cada vez que suba algo a dropbox sobrescriba los ficheros en caso de que existan. Habiendo sido más óptimo comprobar si existía dicho fichero y descartar la subida.
3. No ha sido posible enlazar los títulos de las canciones con Dropbox, por lo que cada vez que se sube una canción se subirán todas las canciones con extensión .mp3 y .wav de la carpeta de descargas.
4. Nos ha sido difícil ejecutar el archivo php en Xampp, tanto que no hemos podido verificar a los usuarios. Al no funcionar Xampp no hemos podido lanzar Mysqlatabase por lo que tampoco teníamos acceso a la BD. En windows funcionaba como debería, pero en Linux no hemos podido llevar a cabo la verificación.

Por último, hemos tenido problemas con los ficheros css, ya que en Firefox no los carga.