

1ª Práctica: Cliente IoT

SISTEMAS WEB



BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO



MIKEL GOTI
15/02/2023

Información general

Mikel Goti Garayalde

Sistemas Web G.2

20/02/2023

Script de python que implementa el cliente web

Descripción

Dentro de la carpeta project se encuentran el archivo main.py y el directorio src que contienen todo el código para el proyecto. También está el directorio resources pero este no es relevante para la ejecución.

El programa se ha programado en un ambiente de anaconda usando **python=3.7.16**. Para el correcto funcionamiento del programa será necesario instalar los siguientes paquetes.

- `pip install pandas`
- `pip install tabulate`
- `pip install psutil`
- `pip install openpyxl`
- `pip install request`
- `pip install selenium`

Teniendo en cuenta las dependencias y la versión customizada de python, la mejor forma para desplegar el proyecto es usando docker. Por eso junto con el directorio project hay un fichero Dockerfile que se encarga de automatizar una imagen con los requerimientos especificados.

Para desplegar el proyecto usando docker hay que navegar hasta donde se haya descargado descomprimir el fichero y abrir una terminal donde se encuentre el directorio descomprimido. Para construir la imagen de docker podemos correr el siguiente comando. Para que este comando tenga efecto tenemos que tener docker instalado y iniciado:

- `docker build -t <nombre contenedor> .`

IMPORTANTE EL ESPACIO Y PUNTO DEL FINAL

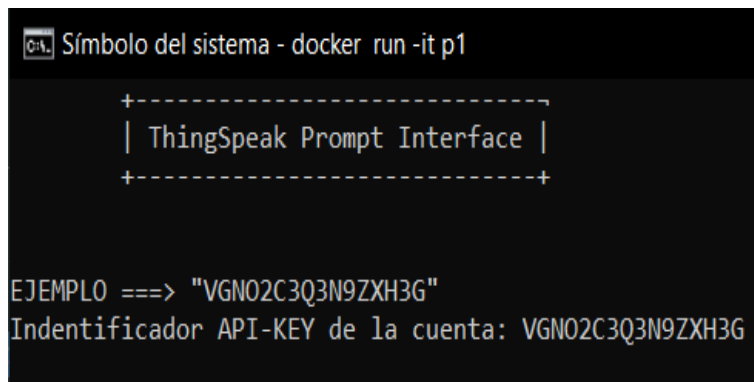
Una vez hemos construido la imagen del proyecto podemos iniciarlo mediante el siguiente comando:

- `docker run -it <nombre contenedor>`

IMPORTANTE AGREGAR LA OPCION -it PARA QUE SE EJECUTE EN MODO INTERACTIVO

Otra opción para ejecutar el programa sin usar docker sería realizar la instalación de la versión de python y sus dependencias manualmente.

Cuando iniciemos el programa se deberá mostrar una pantalla como la siguiente. Ver **imagen 1**. Hay que introducir la api-key de la cuenta usuario de ThingSpeak para poder iniciar sesión. Se puede introducir cualquiera mientras esta sea valida. Si no como ejemplo para realizar la practica esta la cuenta que cree yo.



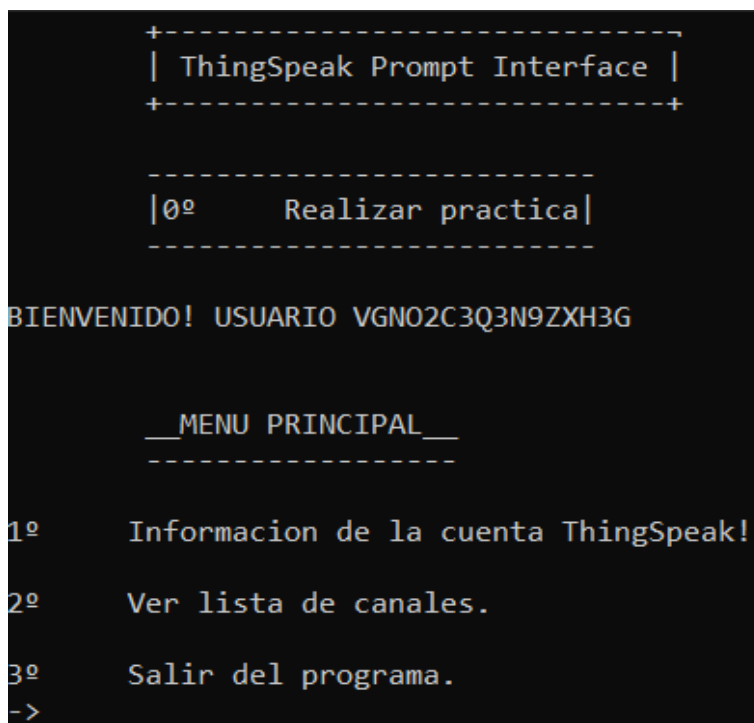
```
Símbolo del sistema - docker run -it p1

+-----+
| ThingSpeak Prompt Interface |
+-----+

EJEMPLO ==> "VGN02C3Q3N9ZXH3G"
Identificador API-KEY de la cuenta: VGN02C3Q3N9ZXH3G
```

Imagen 1: Iniciar sesión mediante la api-key.

Una vez hemos iniciado sesión el programa irá al menú principal del usuario. Aun faltan detalles por pulir. Hay varias opciones que se pueden tomar dependiendo de lo que se necesite.



```
+-----+
| ThingSpeak Prompt Interface |
+-----+

-----
|0º Realizar practica|
-----

BIENVENIDO! USUARIO VGN02C3Q3N9ZXH3G

__MENU PRINCIPAL__
-----

1º Informacion de la cuenta ThingSpeak!
2º Ver lista de canales.
3º Salir del programa.
->
```

Imagen 2: Menú principal.

La idea general de este programa es una forma sencilla y rápida de poder interactuar con ThingSpeak. Para ello no hay nada mas rápido que la terminal. Desde el script se pueden crear canales, borrarlos, subir datos, exportarlos, borrar los campos, actualizarlos, etc... Todo esto y mas cosas desde el menú de cada canal. Se que no es lo que se pedía pero me pareció útil. De todas forma desde el menú principal se puede ejecutar la practica entera. Tanto la primera parte como la segunda. **Si se introduce 0 en el menú principal se ejecutará la practica.**