Extracción de datos

En este notebook se tratará la extracción de datos de sitios web como Reddit o Github, mediante APIs expresamente hechas por el sitio, o funciones de Python que servirán para realizar las peticiones.

Para empezar, se cargarán los ajustes definidos en el apartado anterior junto con la importación de algunas librerías básicas necesarias para comenzar.

```
import sys, os

#Reset del entorno virtual al iniciar la ejecución
%reset -f

#Carga del archivo setup.py
%run -i ../pyenv_settings/setup.py

#Carga del archivo settings.py
#%run "$BASE_DIR/settings.py"
%reload_ext autoreload
%autoreload 2
%config InlineBackend.figure_format = 'png'
```

```
You are working on a local system.
Files will be searched relative to "..".
```

Extracción de datos con la librería *request*

Para este primer ejemplo se va a hacer uso de la librería *request* que se incluye con la instalación de python. Esta librería es el método más básico para acceder y extraer información a través de una API, pero al mismo tiempo es una herramienta muy potente.

Véamos a continuación como se realiza una petición GET en la que se tratará de listar todos los repositorios que cuenten con unas características determinadas. En este caso, por ejemplo, vamos a listar todos los repositorios relacionados con el protocolo Zigbee que, al no usar el token de autenticación de un usuario registrado, nos devolverá aquellos repositorios que sean públicos, si se quieren listar también aquellos privados, hay que proporcionar el token de acceso.

```
import requests
import json

response = requests.get('https://api.github.com/search/repositories',
    params={'q': 'zigbee'},
    headers={'Accept': 'application/vnd.github.v3.text-match+json'})

print(response.status_code)
#print(response.json())
```

```
200
```

Para poder entender de forma más clara la lista de repositorios obtenida, vamos a dar formato Markdown al archivo json y se mostrarán los 5 primeros resultados de la respuesta obtenida:

```
from IPython.display import Markdown, display

def printmd(string):
    display(Markdown(string))

for item in response.json()['items'][:5]:
    printmd('**' + item['name'] + '**' + ': repository ' +
            item['text_matches'][0]['property'] + ' - \"*' +
            item['text_matches'][0]['fragment'] + '*\" matched with '
+ '**' +
            item['text_matches'][0]['matches'][0]['text'] + '**')
```

```
<IPython.core.display.Markdown object>

<IPython.core.display.Markdown object>

<IPython.core.display.Markdown object>

<IPython.core.display.Markdown object>

<IPython.core.display.Markdown object>
```

Listar comentarios del apartado *Issues* de un repositorio

Es posible listar todos los comentarios del apartado Issues de un repositorio si así se especifica en la petición el nombre del repositorio y el dueño del mismo:

```
response = requests.get(

'https://api.github.com/repos/zigbee2mqtt/hassio-zigbee2mqtt/issues/
comments')
print('Response Code', response.status_code)
print('Number of comments', len(response.json()))
```

```
Response Code 200
Number of comments 30
```

Se observa que únicamente ha recopilado 30 comentarios, esto es porque la API de github limita el número de elementos para cada respuesta. Si mostramos por pantalla los enlaces de la respuesta obtendremos el número de páginas contenidas en la respuesta.

```
response.links
```

```
{'next': {'url':
'https://api.github.com/repositories/302841413/issues/comments?
```

```
page=2',
  'rel': 'next'},
 'last': {'url':
'https://api.github.com/repositories/302841413/issues/comments?
page=97',
  'rel': 'last'}}
```

Paginación

Se usa la paginación para limitar el número de elementos en una que se devolverán tras una petición.

Al mostrar los links de *response* se ve que proporciona el enlace a la siguiente página a la respuesta y a la última del total.

Para obtener todos los resultados se debe definir una función que llame a la siguiente página, y así hasta que se hayan procesado todas de forma recursiva.

Del mismo modo, importaremos *Pandas* a nuestro código para transformar los datos obtenidos en un Data Frame.

En el siguiente ejemplo se muestra lo anteriormente explicado además de mostras por pantalla el número de filas que se desee del Data Frame:

```
import pandas as pd

def get_all_pages(url, params=None, headers=None):
    output_json = []
    response = requests.get(url, params=params, headers=headers)
    if response.status_code == 200:
        output_json = response.json()
        if 'next' in response.links:
            next_url = response.links['next']['url']
            if next_url is not None:
                output_json += get_all_pages(next_url, params,
headers)
    return output_json


out = get_all_pages(

"https://api.github.com/repos/zigbee2mqtt/hassio-zigbee2mqtt/issues/
comments",
    params={
        #Como parámetros indicamos la fecha desde la cuál queremos
obtener los resultados,
        #el orden (de creación) y la dirección
        'since': '2020-07-01T10:00:01Z',
        'sorted': 'created',
        'direction': 'desc'
```

```
    },
    headers={'Accept': 'application/vnd.github.v3+json'})

#Los resultados obtenidos los transformamos en un Data Frame para su
posterior análisis
df = pd.DataFrame(out)
pd.set_option('display.max_colwidth', 1)

# #Muestra por pantalla el total de resultados
print(df['body'].count())

# #Muestra por pantalla algún ejemplo de los resultados obtenidos
df[['id', 'created_at', 'body']].sample(10, random_state=42)
```

```
1770
```

```
               id            created_at  \
974   1745724759  2023-10-03T21:05:25Z
275   2514384953  2024-12-03T12:08:21Z
411   2439930531  2024-10-27T09:47:00Z
962   1791747019  2023-11-03T00:39:23Z
518   2233443563  2024-07-17T14:18:10Z
1252  1435803889  2023-02-19T00:51:45Z
1085  1591386877  2023-06-14T14:53:16Z
344   2512280708  2024-12-02T17:51:12Z
1050  1620357373  2023-07-04T14:30:07Z
1457  1322678384  2022-11-21T21:34:45Z


body
974   > Hi moldalex! I only have Aqara D1 with neutral. To confirm,
the instructions say that the device must be connected to the
electrical network. So it turns out that this instruction is just what
you need. But just for devices without a neutral, I don’t know how to
do restoration. It's probably just as easy to remove and insert the
battery when needed.\r\n\r\nThank you for the answer!  Good news for
me! There is a chance :-)\r\n\r\nI just asked this question, because
your link №2 at the bottom is about the firmware for this switch and
there is information (in russian language) that it is for "no neutral"
version. This switch has 2 versions. Power with neutral line (+ -) or
with + only. And the wireless version on the bettery is the 3rd type.
I have the full (+ -) version. As I understood it's the same as yours.
\r\n

275   > Right... I finally have it working after realising the ui
layout was confusing me on top of everything else...\r\n> \r\n> The
files need to be moved to the directory\r\n> \r\n> ```\r\n>
/addon_configs/45df7312_zigbee2mqtt/zigbee2mqtt\r\n> ```\r\n> \r\n>
note the "s" on addon_configs,\r\n> \r\n> and the data_path in the
addon config needs to be\r\n> \r\n> ```\r\n>
```

/addon_config/zigbee2mqtt\r\n> ```\r\n> \r\n> note no "s" on "addon_config",\r\n> \r\n> and on my system, when I give that field focus, it puts red dots under "addon_config" which just adds to the confusion - what does this mean, is it saying there isn't actually a directory called that (which there isn't)???\r\n> \r\n> EDIT - for heaven's sake... it's chrome spell checker putting the red dots there, so ignore that...\r\n> \r\n> How this naming mismatch came about needs looking at, because it really isn't helpful...\r\n\r\nI think it because `/addon_configs/45df7312_zigbee2mqtt/zigbee2mqtt` for multiple addons configs and `/addon_config/zigbee2mqtt` is inside of z2m docker container so only one config will be down there.\r\nBut it confuses a lot. May be use something like `/addon_config_zigbee2mqtt` inside of docker will be much cleaner for the end user.

411   > Interesting, do you still have the debug log of this? Could also be a frontend issue\r\n> \r\n> See [this](https://www.zigbee2mqtt.io/guide/configuration/logging.html) on how to enable debug logging.\r\n\r\n@Koenkk \r\nApologies for the delay, life just stepped in and did not have any spare time since, however I could make some time today and managed to peform a **successful repro**.\r\n\r\nZ2M configuration.yaml debug setting (Z2M service restarted after changing):\r\n log_debug_to_mqtt_frontend: true\r\n log_level: debug\r\n\r\nSize of configuration.yaml:\r\nOrig: 42.6KB\r\nLocalised: 6.01MB\r\n\r\n[Localtime: 10:25] The localization on the UI stalled marking items "done", however, all devices in config were processed.\r\n[Localtime: 10:30] Devices still showing up in Z2M at this point. \r\n[Localtime: 10:33] Stopped Z2M service. \r\n[Localtime: 10:34] Started Z2M service. \r\n[Localtime: 10:34] Devices are not showing up, all devices dissapeared. Repro successful. \r\n[Localtime: 10:36] Stopped Z2M service: \r\n[Localtime: 10:40] Config restored, Z2M service started, devices are shown\r\n\r\nShowing the process for Localise device images, which appears to be stalled after "kitchen-mainlight":\r\n![Z2M_Localise device images_01](https://github.com/user-attachments/assets/eab3128f-db29-4184-ba85-903538c5b892)\r\n\r\nShowing the empty device list after localise device images completed and a Z2M restart:\r\n![Z2M_Localise device images_02](https://github.com/user-attachments/assets/6f9803a4-c5e2-4822-b515-ef6b43209981)\r\n\r\nShowing a device list after the config restore:\r\n![Z2M_Localise device images_03](https://github.com/user-attachments/assets/1eb5a81b-a5b1-4c15-aa77-4852b154a7c5)\r\n\r\nShowing Z2M version and device count:\r\n![Z2M_Localise device images_04](https://github.com/user-attachments/assets/b9283e6e-fea4-405e-b6c9-b5586ed0ba05)\r\n\r\nCould not upload 7z or logs directly, in-line would have been flooding so renamed it to **txt** extension. Please rename to Z2M_Localise.device.images_LOGS_u3.**7z**\r\n[Z2M_Localise.device.images_LOGS_u3.txt](https://github.com/user-attachments/files/17533502/Z2M_Localise.device.images_LOGS_u3.txt)\r\n\r\n\r\nPS: Would be great if the following sensitive data would not be exposed in the debug log by the way, each property have multiple

```
types of exposure: extendedPanID, panID, channelList. I've marked all
references as REDACTED.\r\n
962   This issue is stale because it has been open 30 days with no
activity. Remove stale label or comment or this will be closed in 7
days

518   So glad seeing that there is a solution for the D1 switches, but
reflashing them is a pain for me, not sure if we have any other
solution.

1252  This issue is stale because it has been open 30 days with no
activity. Remove stale label or comment or this will be closed in 7
days

1085  Just happened again with the 2023.06.2 supervisor update. \r\n\
r\nNo other addon stopped.

344   Same issues.. .oooof tried all the above still cannot control
any Z2M devices..

1050  Should also be solved for latest-edge now

1457  "Error: Failed to connect to the adapter (Error: SRSP - SYS -
ping after 6000ms)"\r\nSame error for me. I obtain a "502" if I try to
open "Zigbee2MQTT"
```

La razón por la que aparece este error en la ejecución es porque se ha superado el límite de peticiones y la respuesta es vacía. Esto se solventa en los siguientes apartados, en los que se usa un token de autenticación para un mayor límite y poder realizar diversas ejecuciones asegurando obtener una respuesta

Límites de las APIs

Las APIs tienen un límite a la hora de devolver los resultados, por ejemplo, se pueden haber obtenido 400 comentarios, pero si se accede al sitio del que estos han sido extraídos, cabe la posibilidad de que el total sea ampliamente mayor.

Por esto, se va a definir una función que impida sobrecargar el servidor al que se le realizan las peticiones, disminuyendo la velocidad entre una petición y la siguiente y asegurarnos de que toda la información que se ha solicitado sea descargada correctamente.

```
from datetime import datetime
import time

def handle_rate_limits(response):
    now = datetime.now()
    reset_time = datetime.fromtimestamp(
        #X_RateLimit indica cuántas peticiones se puden realizar por
unidad de tiempo
        int(response.headers['X-RateLimit-Reset']))
```

```
    #X-RateLimit-Remaining indica cuántas peticiones pueden aún
hacerse
    #sin superar el límite establecido
    remaining_requests = response.headers['X-RateLimit-Remaining']
    remaining_time = (reset_time - now).total_seconds()
    intervals = remaining_time / (1.0 + int(remaining_requests))

    print('Esperando por ', intervals)
    time.sleep(intervals)

    return True
```

La librería *requests* no contempla una función que permita reintentar la petición en caso de error, aún con esto, se puede implementar gracias a la librería *HTTPAdapter*.

```
from requests.adapters import HTTPAdapter
from urllib3.util import Retry

retry = Retry(
    #5 reintentos
    total=5,
    #Códigos de erros los cuáles si se reciben, se reintentará
    status_forcelist=[500, 503, 504],
    #Retraso entre reintentos después del segundo intento
    backoff_factor=1
)

retry_adapter = HTTPAdapter(max_retries=retry)

http = requests.Session()
http.mount("https://", retry_adapter)
http.mount("https://", retry_adapter)

response = http.get('https://api.github.com/search/repositories',
                    params={'q': 'zigbee'})

for item in response.json()['items'][:5]:
    print(item['name'])
```

```
zigbee2mqtt.io
zigbee2mqtt
zigbee
zigbee-herdsman-converters
hassio-zigbee2mqtt
```

Juntando las funciones definidas en el apartado de paginación junto con las definidas en este último, quedaría como resultado algo así:

```
import pandas as pd
from requests.adapters import HTTPAdapter
from urllib3.util import Retry

retry = Retry(
    #5 reintentos
    total=5,
    #Códigos de erros los cuáles si se reciben, se reintentará
    status_forcelist=[500, 503, 504],
    #Retraso entre reintentos después del segundo intento
    backoff_factor=1
)

retry_adapter = HTTPAdapter(max_retries=retry)

http = requests.Session()
http.mount("https://", retry_adapter)
http.mount("https://", retry_adapter)

def get_all_pages(url, params=None, headers=None):
    output_json = []
    response = requests.get(url, params=params, headers=headers)
    if response.status_code == 200:
        output_json = response.json()
        if 'next' in response.links:
            next_url = response.links['next']['url']
            if next_url is not None:
                output_json += get_all_pages(next_url, params,
headers)
    return output_json

#Función que lee el token de autenticación de github de un .txt
def load_token(filepath):
    try:
        with open(filepath, 'r', encoding='utf-8') as file:
            token = file.read().strip()  # .strip() elimina espacios y
saltos de línea
            return token
    except FileNotFoundError:
        raise Exception(f"El archivo {filepath} no se encontró.
Asegúrate de que existe y contiene el token.")

token = load_token('../token.txt')

out = get_all_pages(

"https://api.github.com/repos/zigbee2mqtt/hassio-zigbee2mqtt/issues/
comments",
    params={
        #Como parámetros indicamos la fecha desde la cuál queremos
```

```
obtener los resultados,
        #el orden (de creación) y la dirección
        'since': '2020-07-01T10:00:01Z',
        'sorted': 'created',
        'direction': 'desc'
    },
    #Introduzco el token de autenticación para que no supere el número
de peticiones y puede realizar varias ejecuciones seguidas
    headers={'Authorization': f'token {token}',
            'Accept': 'application/vnd.github.v3+json'}
            )

#Los resultados obtenidos los transformamos en un Data Frame para su
posterior análisis
df = pd.DataFrame(out)
#Guardo el dataframe en formato csv/json para su uso en operaciones
posteriores
df.to_csv('../data/output.csv', index=False)
df.to_json('../data/output.json', orient='records', lines=True)

pd.set_option('display.max_colwidth', 1)

# #Muestra por pantalla el total de resultados
#print(df['body'].count())

# #Muestra por pantalla algún ejemplo de los resultados obtenidos
df[['id', 'created_at', 'body']].sample(5, random_state=42)
```

```
              id            created_at  \
471   2318970592  2024-08-29T20:53:54Z
2447  897722230   2021-08-12T15:11:41Z
2380  906339050   2021-08-26T11:50:51Z
1602  1253879951  2022-09-21T15:34:25Z
1094  1576897043  2023-06-05T14:18:26Z


body
471   I'm using a zbdongle-p, so zstack, not ezsp or ember, and I'm
running into this. Just re-paired my whole network from zha to z2m.
Everything worked fine for most of the day. All of a sudden the hass
UI will no longer load. That definitely seems to point to the z2m
addon or z2m itself as the culprit, not the driver for the stick.
2447  If you are using Home Assistant OS with Supervisor and the Add-
On you should always manage your configuration using the add-on UI. \
r\n\r\nIf you are deleting config options and saving and they appear
back it might be because you messed some permissions and Home
Assistant Supervisor can't save the config for it.
2380  Yes... z2m see it as Router. This is correctly powered. I have
used router firmware like others 2 in my network same procedure.
But ... I am not able to see router connected to another router, only
```

```
router to C.\r\nthanks ciotlosm

1602  That means we don't have the OTA file yet; you can extract it
from the TuYa gateway:
https://www.zigbee2mqtt.io/advanced/more/tuya_xiaomi_ota_url.html

1094  i am have that issue too on MOES version of TS0601:
_TZE200_w4cryh2i\r\n\r\nreproduces in clean (not hassio addon)
install. Tried latest and latest-dev containers and in both these
behavior are same\r\n\r\n@Koenkk can it bee fixed?
```