



UNIVERSIDAD TECNOLÓGICA DEL NORTE DE GUANAJUATO

Tecnologías de la Información y Comunicación

Programa educativo:

TSU en Infraestructura de Redes Digitales

Área académica:

Programación de Redes

Asignatura:

Unidad II: Programación de dispositivos IdC **Grupo:** GIR0441

Practica API Parsing JSON

Tarea:

González Manzano Sandra Dania

Alumno:

Gabriel Barrón Rodríguez

Docente:

Dolores Hidalgo, C.I.N., Gto., Martes 07 de Noviembre de 2022

Lugar y fecha:

Nombre de la API: Abbreviations

API Portal / Home Page: Apli list

Descripción: Esta Api proviene de una plataforma basada en un diccionario web por lo que la función de este api es proveer la definición de cualquier palabra insertada por el usuario, en este caso no contiene un código de status como en otras Apis al momento de hacer la consulta

```
import urllib.parse
import requests
"""
Las variables utilizadas fueron el userID, Key y word que contiene la palabra a definir
"""
main_api = "https://www.abbreviations.com/services/v2/defs.php?format=json&"
userID = "8193"
key = "hUR2LjqmfIlk6YSs"

while True:
    palabra = input("Ingrese la palabra: ")
    if palabra == "quit" or palabra == "q":
        break
    url = main_api + urllib.parse.urlencode({"uid":userID,"tokenid":key,"word":palabra})
    print("URL: " + (url))
    print("")
    json_data = requests.get(url).json()
    json_def = json_data["result"][0]["definition"]

    print("La definicion de ",palabra, " es:")
    print(json_def,"\n")
```

```
===== RESTART: D:/Python/API_1.py =====
Ingrese la palabra: pull
URL: https://www.abbreviations.com/services/v2/defs.php?format=json&uid=8193&tokenid=hUR2LjqmfIlk6YSs&word=pull

La definicion de  pull  es:
the act of pulling; applying force to move something toward or with you

Ingrese la palabra: source
URL: https://www.abbreviations.com/services/v2/defs.php?format=json&uid=8193&tokenid=hUR2LjqmfIlk6YSs&word=source

La definicion de  source  es:
the place where something begins, where it springs into being

Ingrese la palabra: agnostic
URL: https://www.abbreviations.com/services/v2/defs.php?format=json&uid=8193&tokenid=hUR2LjqmfIlk6YSs&word=agnostic

La definicion de  agnostic  es:
someone who is doubtful or noncommittal about something

Ingrese la palabra: q
>>> |
```

Nombre de la API: MusixMatch

API Portal / Home Page: Apli list

Descripción: Esta API consiste en regresar la letra de una canción definiendo el artista y evidentemente la canción, esta API si cuenta con un statuscode con el 200 como consulta exitosa.

```
import urllib.parse
import requests
"""
Las variables utilizadas son key para la clave, q_track para el nombre de la cancion, q_artist para definir
el nombre del artista, y status_code para definir el status
"""
main_api = "https://api.musixmatch.com/ws/1.1/matcher.lyrics.get?"
key = "020d6a15da05a237fclbbbb0dcdbcd15"

while True:
    artista = input("Ingrese el artista: ")
    if artista == "quit" or artista == "q":
        break
    cancion = input("Ingrese la canción: ")
    if cancion == "quit" or cancion == "q":
        break
    url = main_api + urllib.parse.urlencode({"q_track":cancion,"q_artist":artista,"apikey":key})
    print("URL: " + (url),"\n")
    json_data = requests.get(url).json()
    json_status = json_data["message"]["header"]["status_code"]
    if json_status == 200:
        json_letra = json_data["message"]["body"]["lyrics"]["lyrics_body"]
        print("=====")
        print("Código de estatus: ",json_status," La API respondió correctamente")
        print("=====")
        print("*****La letra de la canción ",cancion," es: ***** \n \n",json_letra,"\n")
    elif json_status == 404:
        print("=====")
        print("Código de estatus: ",json_status, " No se encontro la canción")
        print("=====")
```

Ingrese el artista: Metallica
Ingrese la canción: Enter Sandman
URL: https://api.musixmatch.com/ws/1.1/matcher.lyrics.get?q_track=Enter+Sandman&q_artist=Metallica&apikey=020d6a15da05a237fclbbbb0dcdbcd15

=====

Código de estatus: 200 La API respondió correctamente

=====

*****La letra de la canción Enter Sandman es: *****

Say your prayers, little one, don't forget, my son
To include everyone
I tuck you in, warm within, keep you free from sin
Til the sandman he comes

Sleep with one eye open
Gripping your pillow tight

Exit light
Enter night
Take my hand
We're off to never never-land

Something's wrong, shut the light, heavy thoughts tonight
And they aren't Snow White
Dreams of war, dreams of liars, dreams of dragon's fire
And of things that will bite, yeah

Sleep with one eye open
...

***** This Lyrics is NOT for Commercial use *****

Ingrese el artista: DNSNJDV
Ingrese la canción: NJVKSDN
URL: https://api.musixmatch.com/ws/1.1/matcher.lyrics.get?q_track=NJVKSDN&q_artist=DNSNJDV&apikey=020d6a15da05a237fclbbbb0dcdbcd15

=====

Código de estatus: 404 No se encontro la canción

=====

Nombre de la API: words

API Portal / Home Page: Apli list

Descripción: Esta api tiene un motor de búsqueda interesante pues regresa el antónimo y el sinónimo de una palabra ingresada por el usuario, no cuenta con un status code ya que si se presenta un error lo hace en un diccionario diferente por lo que si se hace referencia

```
import urllib.parse
import requests
"""
Las variables utilizadas son userID y key en la parte de autenticación en la API, y la
variable word como parametro para capturar la palabra
"""
main_api = "https://www.stands4.com/services/v2/syno.php?format=json&"
userID = "8193"
key = "hUR2LjqmfIlk6YSs"

while True:
    palabra = input("Ingrese la palabra: ")
    if palabra == "quit" or palabra == "q":
        break

    url = main_api + urllib.parse.urlencode({"uid":userID,"tokenid":key,"word":palabra})
    print("URL: " + url, "\n")
    json_data = requests.get(url).json()
    json_sin = json_data["result"][0]["synonyms"]
    json_ant = json_data["result"][0]["antonyms"]
    print("=====")
    print("Palabra: ", palabra)
    print("Sinónimos: ", json_sin, "\n")
    print("Antónimos: ", json_ant)
    print("=====")
```

```
===== RESTART: D:/Python/API_3.py =====
Ingrese la palabra: cold
URL: https://www.stands4.com/services/v2/syno.php?format=json&uid=8193&tokenid=hUR
2LjqmfIlk6YSs&word=cold

=====
Palabra: cold
Sinónimos: common cold, low temperature, coldness, frigidness, frigidity, cold

Antónimos: flaming, red-hot, mild, thermal, overheated, hot, fervid, imperfect, t
ropical, calefactive, igneous, torrid, enthusiastic, fiery, blistering, sweltry, h
eatable, sizzling, close, sweltering, nigh, sultry, calefactory, near, white, whit
e-hot, baking, alive(p), calefacient, fervent, scorching, heated up, heated, human
e, warm, het up, warming, stifling, blistery, sensual, het, sulphurous, conscious,
tropic, calorifacient, baking hot, fresh, hottish, live, calorific, sulfurous, lov
ing, original

=====
Ingrese la palabra: run
URL: https://www.stands4.com/services/v2/syno.php?format=json&uid=8193&tokenid=hUR
2LjqmfIlk6YSs&word=run

=====
Palabra: run
Sinónimos: count, running game, rivulet, political campaign, rill, numeration, re
ckoning, ladder, ravel, discharge, campaign, run, footrace, foot race, test, enume
ration, running play, tally, streak, trial, outpouring, runnel, counting, streamle
t, running

Antónimos: idle, tick over

=====
Ingrese la palabra: q
```