

Práctica API Parsing JSON

Instrucciones

Paso 1) Visita los siguientes sitios

- a. [Yandex](#)
- b. [Apli List](#)

Paso 2) Seleccionar al menos cuatro API's y describirlas Valor 0.5 puntos

Nombre de la API	Tamaños y tipos de pizza
Descripción de la API	Distintos tamaños y tipos de pizza para elegir
API Portal / Home Page	https://api.nutritionix.com/v1_1/search/pizza?

Nombre de la API	Personajes de anime
Descripción de la API	Busqueda de personajes de anime
API Portal / Home Page	https://api.jikan.moe/v4/top/anime

Nombre de la API	Post de Instagram
Descripción de la API	Localizar post de instagram
API Portal / Home Page	https://instagram47.p.rapidapi.com/location_posts

Paso 3) Cerciorarse que las API's no se repitan con tus compañeros de clase

Paso 4) Ve al portal o página de inicio de la API seleccionada.

Paso 5) Regístrate en el portal. Complete el formulario para crear una nueva cuenta para obtener la clave o KEY

Paso 6) Copia la clave de consumidor en el Bloc de notas para uso futuro.

Paso 7) Abra un archivo de script en blanco y guárdalo como practica01_unidad2.py, practica02_unidad2.py, practica03_unidad2.py

Paso 8) Agregar los siguientes datos de comentario

Descripción de la API

Autor

Fecha de creación

Paso 9) Importar los módulos **urllib.parse** y **requests**.

Paso 10) Crea la variable que construye la URL.

```
main_api = "http://api.coronatracker.com/v5/analytics/dailyNewStats?limit="
```

Paso 11) Utiliza la función input para datos de entrada de usuario

Paso 12) Crea la variable que realiza la solicitud

```
json_data = requests.get(url).json()
```

Paso 13) Analiza el dato regresado para obtener el código de estatus regresado e imprímelo.

Paso 14) Utiliza un ciclo infinito con **while True:** para probar tu API; además utiliza una condicional if donde cheque la entrada del usuario **Salir** o **S** para terminar el programa.

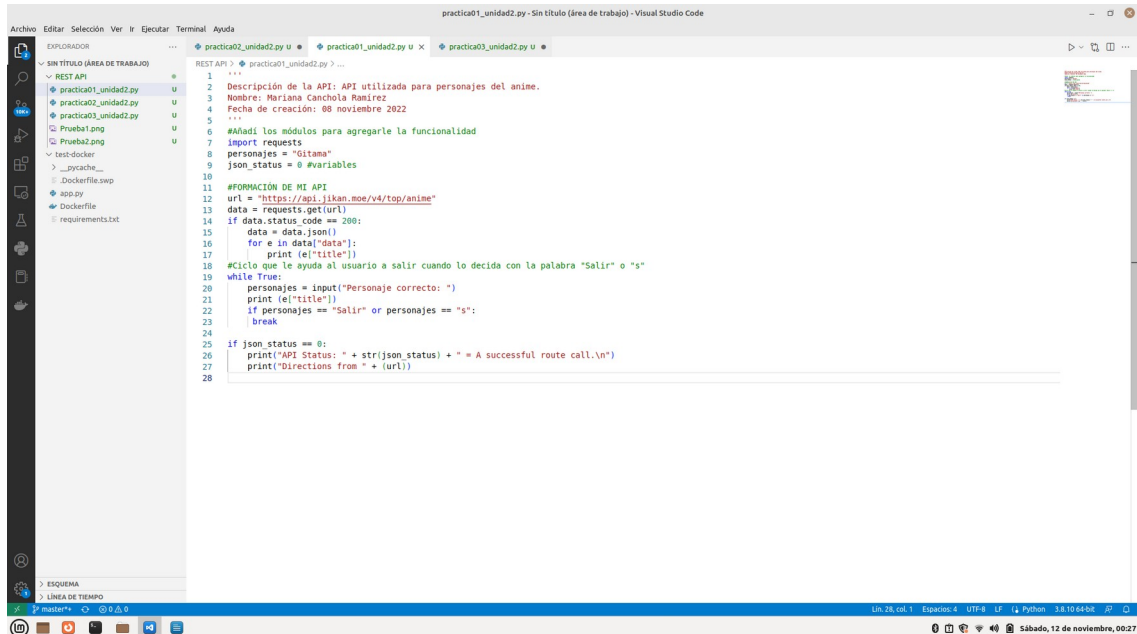
Paso 15) Prueba la funcionalidad de salir y con una impresión de pantalla

Agrega tu pantalla de salida con varias corridas

Paso 16) Crear una carpeta llamada practica en el repositorio de la unidad II

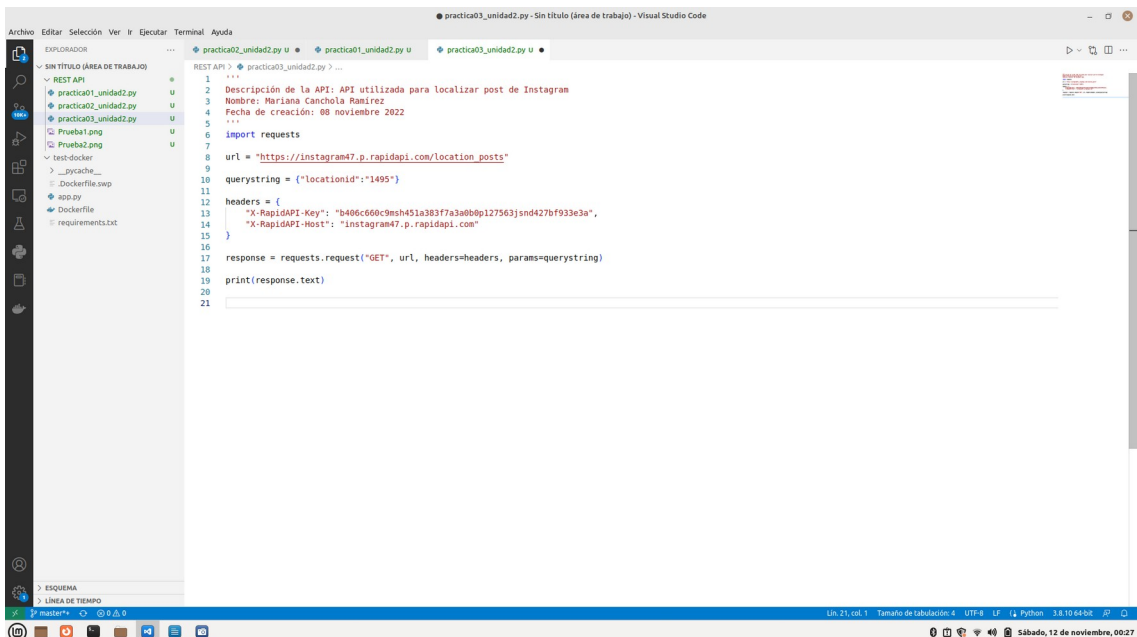
Paso 17) Agrega tu código fuente documentado en GitHub y el archivo resultado de esta

```
1 ***
2 Descripción de la API: Esta api nos da información acerca de una bebida
3 Nombre: Mariana Canchola Ramirez
4 Fecha de creación: 08 noviembre 2022
5 ***
6
7 #Añadi los módulos para agregarle la funcionalidad
8 import urllib.parse
9 import requests
10
11 #FORMACIÓN DE MI API
12 url = "https://api.nutritionix.com/v1_1/search/pizza?"
13 appId = "f7d3f36d"
14 appKey = "076f7595c3dd563a0aa6d66df275f41f"
15 query = "Medium pizza"
16
17 #Ciclo que le ayuda al usuario a salir cuando lo decida con la palabra "Salir" o "s"
18 while True:
19     query= input("Tamaño: ")
20     if query == "Salir" or query == "s":
21         break
22     #Funcionalidad del codigo de solicitud
23     url = url + urllib.parse.urlencode({"query":query,"search": query,"appId": appId,"appKey":appKey})
24     print("URL: " + url)
25
26     json_data = requests.get(url).json()
27     json_status = json_data['hits'][0]['fields']['item_name']
28
29     for elem in json_data["hits"]:
30         if query in elem["fields"]["item name"]:
31             print("Tamaño pizza: " + str(elem["fields"]["item name"]) )
32             print("ID: " + str(elem["_id"]))
33
34 if json_status == 0:
35     print("API Status: " + str(json_status) + " = A successful route call.\n")
36     print("Directions from " + url)
```



```
practica01_unidad2.py - Sin título (área de trabajo) - Visual Studio Code
Archivo  Editar  Selección  Ver  Ejecutar  Terminal  Ayuda
EXPLORADOR
Sin título (ÁREA DE TRABAJO)
  REST API
    practica01_unidad2.py
    practica02_unidad2.py
    practica03_unidad2.py
  Prueba1.png
  Prueba2.png
  test-docker
  ..._pycache_...
  Dockerfile.swp
  app.py
  Dockerfile
  requirements.txt
ESQUEMA
  LINEA DE TIEMPO
  Python 3.8.10
  Python 3.8.10.6458
  Sábado, 12 de noviembre, 00:27

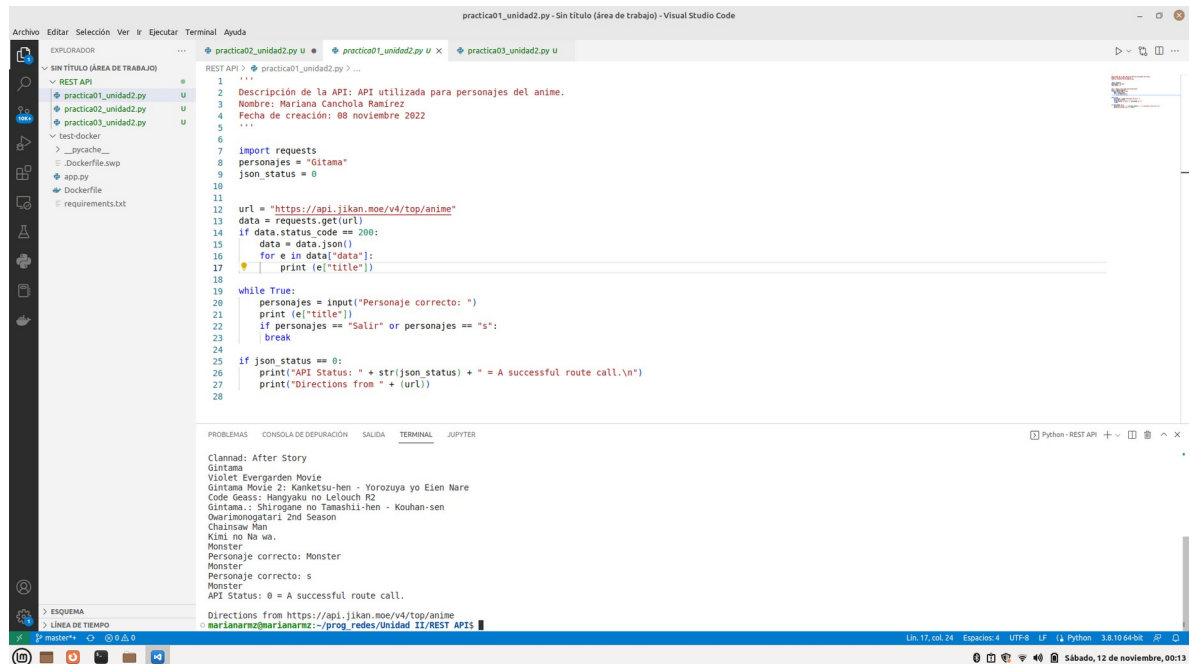
practica01_unidad2.py
1  ...
2  Descripción de la API: API utilizada para personajes del anime.
3  Nombre: Mariana Canciola Ramirez
4  Fecha de creación: 08 noviembre 2022
5  ...
6  #Añadi los módulos para agregarle la funcionalidad
7  import requests
8  personajes = "Gitana"
9  json_status = 0 #Variables
10
11 #FORMACIÓN DE MI API
12 url = "https://api.jikan.moe/v4/top/anime"
13 data = requests.get(url)
14 if data.status_code == 200:
15     data = data.json()
16     for e in data["data"]:
17         print(e["title"])
18 #Ciclo que le ayuda al usuario a salir cuando lo decida con la palabra "Salir" o "s"
19 while True:
20     personajes = input("Personaje correcto: ")
21     print(e["title"])
22     if personajes == "Salir" or personajes == "s":
23         break
24
25 if json_status == 0:
26     print("API Status: " + str(json_status) + " = A successful route call.\n")
27     print("Directions from " + (url))
28
```



```
practica01_unidad2.py - Sin título (área de trabajo) - Visual Studio Code
Archivo  Editar  Selección  Ver  Ejecutar  Terminal  Ayuda
EXPLORADOR
Sin título (ÁREA DE TRABAJO)
  REST API
    practica01_unidad2.py
    practica02_unidad2.py
    practica03_unidad2.py
  Prueba1.png
  Prueba2.png
  test-docker
  ..._pycache_...
  Dockerfile.swp
  app.py
  Dockerfile
  requirements.txt
ESQUEMA
  LINEA DE TIEMPO
  Python 3.8.10
  Python 3.8.10.6458
  Sábado, 12 de noviembre, 00:27

practica01_unidad2.py
1  ...
2  Descripción de la API: API utilizada para localizar post de Instagram
3  Nombre: Mariana Canciola Ramirez
4  Fecha de creación: 08 noviembre 2022
5  ...
6  import requests
7
8  url = "https://instagram47.p.rapidapi.com/location_posts"
9
10 querystring = {"locationid": "1495"}
11
12 headers = {
13     "X-RapidAPI-Key": "b406c60c9msh451a383f7a3a8b0p127563jsnd427bf933e3a",
14     "X-RapidAPI-Host": "instagram47.p.rapidapi.com"
15 }
16
17 response = requests.request("GET", url, headers=headers, params=querystring)
18
19 print(response.text)
20
21
```

RESULTADOS

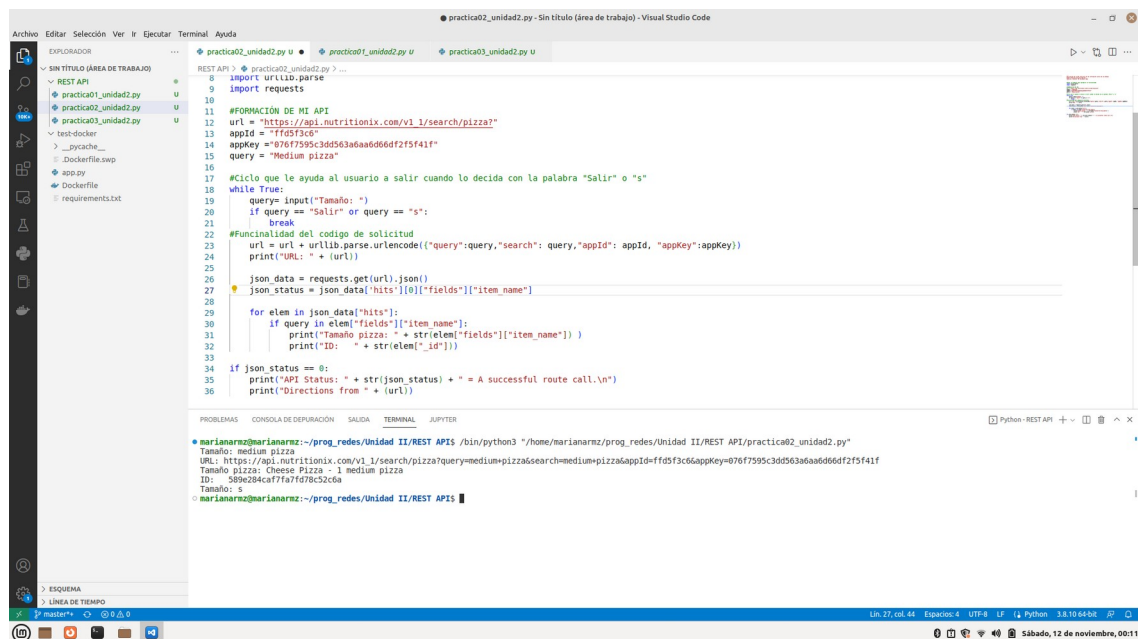


```
practica01_unidad2.py - Sin título (área de trabajo) - Visual Studio Code

EXPLORADOR
  SIN TÍTULO (ÁREA DE TRABAJO)
    REST API
      practica01_unidad2.py
      practica02_unidad2.py
      practica03_unidad2.py
    test-docker
    __pycache__
    .Dockerfile.swp
    app.py
    Dockerfile
    requirements.txt

practica01_unidad2.py
1  """
2  Descripción de la API: API utilizada para personajes del anime.
3  Nombre: Mariana Cancho Ramirez
4  Fecha de creación: 08 noviembre 2022
5  """
6
7  import requests
8  personajes = "Gintama"
9  json_status = 0
10
11
12  url = "https://api.jikan.moe/v4/top/anime"
13  data = requests.get(url)
14  if data.status_code == 200:
15      data = data.json()
16      for e in data["data"]:
17          print(e["title"])
18
19  while True:
20      personajes = input("Personaje correcto: ")
21      print(e["title"])
22      if personajes == "Salir" or personajes == "s":
23          break
24
25  if json_status == 0:
26      print("API Status: " + str(json_status) + " = A successful route call.\n")
27      print("Directions from " + url)
28
PROBLEMAS  CONSOLA DE DEPURACIÓN  SALIDA  TERMINAL  JUPYTER
Python - REST API

Clanad: After Story
Gintama
Violet Evergarden Movie
Gintama Movie 2: Kanketsu-hen - Yorozuya yo Eien Nare
Code Geass: Hangyaku no Lelouch R2
Gintama: Shirogane no Tamashii-hen - Kouhan-sen
Owarimonogatari 2nd Season
Chainsaw Man
Kim! no Na wa.
Monster
Personaje correcto: Monster
Monster
Personaje correcto: s
Monster
API Status: 0 = A successful route call.
Directions from https://api.jikan.moe/v4/top/anime
marianar@marianar:~/prog_redes/Unidad II/REST API$
```



```
practica02_unidad2.py - Sin título (área de trabajo) - Visual Studio Code

EXPLORADOR
  SIN TÍTULO (ÁREA DE TRABAJO)
    REST API
      practica01_unidad2.py
      practica02_unidad2.py
      practica03_unidad2.py
    test-docker
    __pycache__
    .Dockerfile.swp
    app.py
    Dockerfile
    requirements.txt

practica02_unidad2.py
1  """
2  Descripción de la API: API utilizada para personajes del anime.
3  Nombre: Mariana Cancho Ramirez
4  Fecha de creación: 08 noviembre 2022
5  """
6
7  import requests
8  import urllib.parse
9  import requests
10
11  #FORMACIÓN DE MI API
12  url = "https://api.nutritionix.com/v1_1/search/pizza?"
13  appId = "ff05f3c8"
14  apiKey = "076f7595c3d563adaad66d72f5f41f"
15  query = "Medium pizza"
16
17  #Ciclo que le ayuda al usuario a salir cuando lo decida con la palabra "Salir" o "s"
18  while True:
19      query = input("Tamaño: ")
20      if query == "Salir" or query == "s":
21          break
22
23  #Funcionalidad del código de solicitud
24  url = url + urllib.parse.urlencode({"query": query, "search": query, "appId": appId, "apiKey": apiKey})
25  print("URL: " + url)
26
27  json_data = requests.get(url).json()
28  json_status = json_data["hits"][0]["fields"]["item_name"]
29
30  for elem in json_data["hits"]:
31      if query in elem["fields"]["item_name"]:
32          print("Tamaño pizza: " + str(elem["fields"]["item_name"]))
33          print("ID: " + str(elem["id"]))
34
35  if json_status == 0:
36      print("API Status: " + str(json_status) + " = A successful route call.\n")
37      print("Directions from " + url)
38
PROBLEMAS  CONSOLA DE DEPURACIÓN  SALIDA  TERMINAL  JUPYTER
Python - REST API

marianar@marianar:~/prog_redes/Unidad II/REST API$ /bin/python3 ~/home/marianar/prog_redes/Unidad II/REST API/practica02_unidad2.py
Tamaño: medium pizza
URL: https://api.nutritionix.com/v1_1/search/pizza?query=medium+pizza&search=medium+pizza&appId=ff05f3c8&apiKey=076f7595c3d563adaad66d72f5f41f
ID: 50bc83ca77a7678c28ca
Tamaño: s
marianar@marianar:~/prog_redes/Unidad II/REST API$
```

