**LLAMADAS A SERVICIOS API**

* Un Servicio Api es un origen de datos en la nube.
* Nos permite poder acceder a datos desde diferentes dispositivos, ya que no nos comunicamos con una base de datos directamente, leemos contenido Web de tipo String en formato JSON.
* Para acceder, debemos instalar la librería **requests** dentro de Python
* Debemos tener en cuenta que vamos a consumir servicios Api gratuitos y la primera vez tardarán en responder a las peticiones.

**Acceso a un JSON en un servicio Api**

Primero instalamos el módulo de Request en nuestro equipo

python -m pip install requests

**Después a ejecutar!!!!**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

import requests

response = requests.get('https://api.covid19api.com/summary').*text*

print(response)

**Ejemplo leer API empleado ID**



import requests

import json

api\_url = "https://apiempleadosspgs.azurewebsites.net/api/Empleados/7322"

#capturamos la respuesta

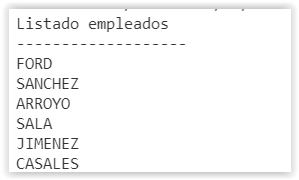
response = requests.get(api\_url)

#convertimos la respuesta a objeto diccionario

empleado = response.json()

print(empleado)

print(empleado["apellido"])



import requests

import json

api\_url = "https://apiempleadosspgs.azurewebsites.net/api/Empleados"

#capturamos la respuesta

response = requests.get(api\_url)

#convertimos la respuesta a objeto diccionario

empleados = response.json()

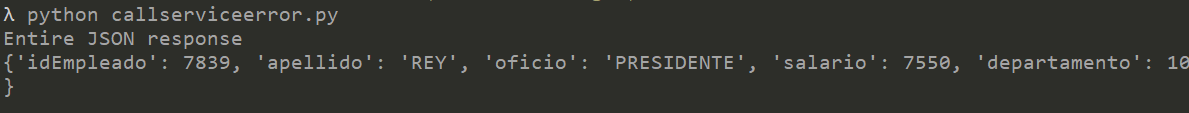
print("Listado empleados")

print("-------------------")

for i in empleados:

    print(i["apellido"])

**LLAMADA A UN ELEMENTO API EMPLEADO CON ERROR**



**PYTHON**

import requests

from requests.exceptions import HTTPError

apiurl = "https://apiempleadosspgs.azurewebsites.net/api/Empleados/7839"

try:

    response = requests.get(apiurl)

    response.raise\_for\_status()

    # access JSOn content

    jsonResponse = response.json()

    print("Entire JSON response")

    print(jsonResponse)

except HTTPError as http\_err:

    print(*f*'HTTP error occurred: {http\_err}')

except *Exception* as err:

    print(*f*'Other error occurred: {err}')

**RECORRER KEY-VALUE JSON**

Texto

Descripción generada automáticamente

import requests

from requests.exceptions import HTTPError

apiurl = "https://apiempleadosspgs.azurewebsites.net/api/Empleados/7839"

try:

    response = requests.get(apiurl)

    response.raise\_for\_status()

    # access JSOn content

    jsonResponse = response.json()

    for key, value in jsonResponse.items():

        print(key, ":", value)

except HTTPError as http\_err:

    print(*f*'HTTP error occurred: {http\_err}')

except *Exception* as err:

    print(*f*'Other error occurred: {err}')

**json.dumps()** convierte un objeto json en String

**EJEMPLO POST (INSERTAR)**

Texto

Descripción generada automáticamente

import requests

apiurl = "https://apicruddepartamentoscore.azurewebsites.net"

request = "/api/departamentos"

departamento = {"numero": 40, "nombre": "PRODUCCION", "localidad": "SEVILLA"}

response = requests.post(apiurl + request, *json*=departamento)

#convertimos la respuesta a objeto diccionario

#print(response.json())

print("Status: " + *str*(response.*status\_code*))

**EJEMPLO PUT (MODIFICAR)**

import requests

apiurl = "https://apicruddepartamentoscore.azurewebsites.net"

request = "/api/departamentos"

departamento = {"numero": 20, "nombre": "INVESTIGACION", "localidad": "MADRID"}

response = requests.put(apiurl + request, *json*=departamento)

#convertimos la respuesta a objeto diccionario

#print(response.json())

print("Status: " + *str*(response.*status\_code*))

**EJEMPLO DELETE (ELIMINAR)**

import requests

apiurl = "https://apicruddepartamentoscore.azurewebsites.net"

request = "/api/departamentos/122234"

#departamento = {"numero": 10, "nombre": "CONTABILIDAD", "localidad": "ELCHE"}

response = requests.delete(apiurl + request)

#response = requests.put(apiurl + request, json=departamento)

#convertimos la respuesta a objeto diccionario

#print(response.json())

print("Status: " + *str*(response.*status\_code*))

Ahora vamos a realizar un ejemplo completo trabajando con clases

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

Creamos un nuevo archivo llamado **Departamento.py**

**DEPARTAMENTO.PY**

*class* Departamento:

    numero = 0

    nombre = ""

    localidad = ""

*def* \_\_str\_\_(*self*):

        return (*str*(*self*.*numero*) + " - " + *self*.*nombre* + " - " + *self*.*localidad*)

A continuación, vamos a crear una clase que se encargará de consumir los servicios y darnos respuestas en consultas de acción y selección sobre departamentos.

Creamos una clase llamada **ConexionApiDepartamentos.py**

**CONEXIONAPIDEPARTAMENTOS.PY**

import requests

from Departamento import Departamento

*class* ConexionApiDepartamentos:

*def* \_\_init\_\_(*self*):

*self*.*apiurl* = "https://apicruddepartamentoscore.azurewebsites.net/";

*def* insertarDepartamento(*self*, *numero*, *nombre*, *localidad*):

        request = "/api/departamentos"

        #CREAMOS UN OBJETO DEPARTAMENTO JSON

        departamento = {"numero": numero, "nombre": nombre, "localidad": localidad}

        #CREAMOS NUESTRA URL CON LA PETICION

        url = *self*.*apiurl* + request;

        response = requests.post(url, *json*=departamento)

        return response.*status\_code*;

*def* modificarDepartamento(*self*, *numero*, *nombre*, *localidad*):

        request = "/api/departamentos"

        #CREAMOS UN OBJETO DEPARTAMENTO JSON PARA MODIFICAR

        departamento = {"numero": numero, "nombre": nombre, "localidad": localidad}

        #CREAMOS NUESTRA URL CON LA PETICION

        url = *self*.*apiurl* + request;

        response = requests.put(url, *json*=departamento)

        return response.*status\_code*;

*def* eliminarDepartamento(*self*, *numero*):

        request = "/api/departamentos/" + *str*(numero)

        #CREAMOS NUESTRA URL CON LA PETICION

        url = *self*.*apiurl* + request;

        response = requests.delete(url)

        return response.*status\_code*;

*def* buscarDepartamento(*self*, *numero*):

        request = "/api/departamentos/" + *str*(numero)

        url = *self*.*apiurl* + request;

        response = requests.get(url)

        if (response.*status\_code* == 200):

            #capturamos la respuesta en json

            departamentojson = response.json()

            #Creamos un objeto de la clase Departamento para

            #devolver la respuesta

            departamento = Departamento()

            departamento.*numero* = departamentojson["numero"]

            departamento.*nombre* = departamentojson["nombre"]

            departamento.*localidad* = departamentojson["localidad"]

            return departamento

        else:

            return None

*def* todosDepartamentos(*self*):

        request = "/api/departamentos"

        url = *self*.*apiurl* + request;

        response = requests.get(url)

        #Creamos una lista para devolver todos los departamentos

        departamentos = []

        #capturamos la respuesta en json

        departamentojson = response.json()

        for row in departamentojson:

            #Creamos un departamento por cada fila del bucle

            departamento = Departamento()

            #Asignamos los datos al nuevo objeto dept

            departamento.*numero* = row["numero"]

            departamento.*nombre* = row["nombre"]

            departamento.*localidad* = row["localidad"]

            #Añadimos a la lista cada nuevo objeto dept

            departamentos.append(departamento)

        return departamentos

Por último, creamos una clase **main** que será la encargada de realizar las peticiones. Creamos un nuevo archivo Python llamado **MainApiDepartamentos.py**

**MAINAPIDEPARTAMENTOS.PY**

from ConexionApiDepartamentos import ConexionApiDepartamentos

print("DEPARTAMENTOS API CON CLASES")

#Creamos el objeto conexion para trabajar

connection = ConexionApiDepartamentos()

opcion = -1

while (opcion != 6):

    print("------------MENU-----------")

    print("1.- Nuevo Departamento")

    print("2.- Eliminar departamento")

    print("3.- Modificar departamento")

    print("4.- Buscar departamento")

    print("5.- Mostrar departamentos")

    print("6.- Salir")

    print("Seleccione un opción")

    opcion = *int*(input())

    if (opcion == 1):

        print("-----INSERTANDO-----")

        print("Introduzca un número")

        numero = *int*(input())

        print("Introduzca un nombre")

        nombre = input()

        print("Introduzca una localidad")

        localidad = input()

        respuesta = connection.insertarDepartamento(numero, nombre, localidad)

        print("Departamentos insertados " + *str*(respuesta))

    elif (opcion == 2):

        print("-----ELIMINANDO-----")

        print("Número de departamento a eliminar")

        numero = *int*(input())

        respuesta = connection.eliminarDepartamento(numero)

        print("Departamentos eliminados: " + *str*(respuesta))

    elif (opcion == 3):

        #Llamamos a modificar

        print("-----Update------")

        print("Introduzca número de departamento")

        numero = *int*(input())

        print("Introduzca nuevo nombre")

        nombre = input()

        print("Introduzca nueva localidad")

        localidad = input()

        respuesta = connection.modificarDepartamento(

            numero, nombre, localidad)

        print("Departamentos modificados: " + *str*(respuesta))

    elif (opcion == 4):

        print("Introduzca número departamento")

        numero = *int*(input())

        #Se supone que nos tiene que devoler un departamento

        #o nada...

        departamento = connection.buscarDepartamento(numero)

        if (not departamento):

            print("No existe el departamento")

        else:

            print(departamento)

    elif (opcion == 5):

        #Recuperamos la lista de departamentos

        departamentos = connection.todosDepartamentos()

        for dept in departamentos:

            print(dept)

    elif (opcion == 6):

        print("Hasta luego")

    else:

        print("Opción incorrecta")

print("Fin de programa")