

Módulo de adaptación

Master en Business Intelligence y Big Data

PROFESOR

Antonio Sarasa Cabezuelo

Manipulando JSON usando Python

Documentos JSON

- Para gestionar documentos JSON desde Python se usa el modulo JSON que permite la traducción de datos JSON en valores de Python.
- JSON no puede almacenar cualquier tipo de valor Python, únicamente cadenas, enteros, reales, booleanos, listas, diccionarios y el tipo None.
- JSON no puede representar objetos específicos de Python tales como Ficheros, expresiones regulares,...

Documentos JSON

- Para traducir una cadena que contiene datos JSON en un valor de Python se utiliza el método `json.loads()`.

```
JsonDatos='{"nombre": "Sofia", "matriculado":true, "asignaturas":34, "ID":null}'  
import json  
PythonDatos=json.loads(JsonDatos)  
print PythonDatos  
  
{u'matriculado': True, u'nombre': u'Sofia', u'asignaturas': 34, u'ID': None}
```

Documentos JSON

- La llamada al método loads() del módulo json permite cargar una cadena de datos JSON en valores de Python, retornando como resultado una lista dónde cada elemento es un diccionario. Si se quiere acceder a los distintos elementos del diccionario se usan los índices. La cadena JSON utiliza dobles comillas para las claves.

Documentos JSON

- Observar que los valores en los diccionarios no están ordenados, por lo que los pares clave-valor pueden aparecer en orden diferente a como aparecían en la cadena original.

Documentos JSON

- La tabla de correspondencia entre JSON y valores Python:

JSON	Python
object	dict
array	list
string	unicode
number(int)	int,long
number(real)	float
true	True
false	False
null	None

Documentos JSON

- Para escribir un valor de Python como una cadena de datos JSON se usa el método `json.dumps()`.

```
PythonDatos={'nombre': 'Sofia', 'matriculado':True, 'asignaturas':34, 'ID':None}  
import json  
JSONDatos=json.dumps(PythonDatos)  
print JSONDatos
```

```
{"matriculado": true, "nombre": "Sofia", "asignaturas": 34, "ID": null}
```


Documentos JSON

- La tabla de correspondencia entre los valores de Python y JSON:

Python	JSON
dict	object
list, tuple	array
str,unicode	string
int,long,float	number
True	true
False	false
None	null

Documentos JSON

Ejemplo 1

Considerar un programa que permita:

- Leer desde teclado una ciudad
- Llamar a la API de geocodificación de Google
- Extraer la información en formato JSON que nos devuelve.

Documentos JSON

Para implementarlo:

- El programa toma la cadena de búsqueda que el usuario introduce por teclado y se construye una URL tomando la cadena introducida. Mediante urllib se recupera el texto en JSON que la API de geocodificación de Google devuelve.

```
import urllib
import json

serviceurl = 'http://maps.googleapis.com/maps/api/geocode/json?'

while True:
    address = raw_input('Entrar ciudad: ')
    if len(address) < 1 : break

    url = serviceurl + urllib.urlencode({'sensor':'false', 'address': address})
    print 'Recuperando', url
    uh = urllib.urlopen(url)
    data = uh.read()
    print 'Recuperados',len(data), 'characters'
```

Documentos JSON

- Una vez recuperados los datos JSON se analizan y se muestran.

```
try: js = json.loads(str(data))
except: js = None
if 'status' not in js or js['status'] != 'OK':
    print '==== Fallo de recuperación ==== '
    print data
    continue

print json.dumps(js, indent=4)

lat = js["results"][0]["geometry"]["location"]["lat"]
lng = js["results"][0]["geometry"]["location"]["lng"]
print 'lat',lat,'lng',lng
location = js['results'][0]['formatted_address']
print location
```

Documentos JSON

- El programa completo sería:

```
import urllib
import json

serviceurl = 'http://maps.googleapis.com/maps/api/geocode/json?'

while True:
    address = raw_input('Entrar ciudad: ')
    if len(address) < 1 : break

    url = serviceurl + urllib.urlencode({'sensor':'false', 'address': address})
    print 'Recuperando', url
    uh = urllib.urlopen(url)
    data = uh.read()
    print 'Recuperados',len(data),'characters'

    try: js = json.loads(str(data))
    except: js = None
    if 'status' not in js or js['status'] != 'OK':
        print '==== Fallo de recuperación ====='
        print data
        continue

    print json.dumps(js, indent=4)

    lat = js["results"][0]["geometry"]["location"]["lat"]
    lng = js["results"][0]["geometry"]["location"]["lng"]
    print 'lat',lat,'lng',lng
    location = js['results'][0]['formatted_address']
    print location
```

Documentos JSON

```
>>>
Entrar ciudad: Madrid
Recuperando http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Madrid
Recuperados 2167 characters
{
  "status": "OK",
  "results": [
    {
      "geometry": {
        "location_type": "APPROXIMATE",
        "bounds": {
          "northeast": {
            "lat": 40.5638447,
            "lng": -3.5249115
          },
          "southwest": {
            "lat": 40.3120639,
            "lng": -3.8341618
          }
        },
        "viewport": {
          "northeast": {
            "lat": 40.5638447,
            "lng": -3.5249115
          }
        }
      }
    }
  ]
}
```

Documentos JSON

Ejemplo 2

Twitter tiene una disponible una API con servicios para los usuarios. Para poder utilizar dicha API es necesario el uso de firmas OAuth(es una tecnología para firmar peticiones en Internet) en cada solicitud.

Documentos JSON

- Para obtener datos de twitter es necesario seguir los siguientes pasos:
 - Si no se dispone de cuenta en twitter, entonces hay que crearse una cuenta en <https://twitter.com/>
 - A continuación se navega a la dirección <https://apps.twitter.com/>, dónde habrá que autenticarse. Esta página da acceso a la api de twitter para desarrolladores.

Documentos JSON

Inicia sesión en Twitter

Iniciar sesión

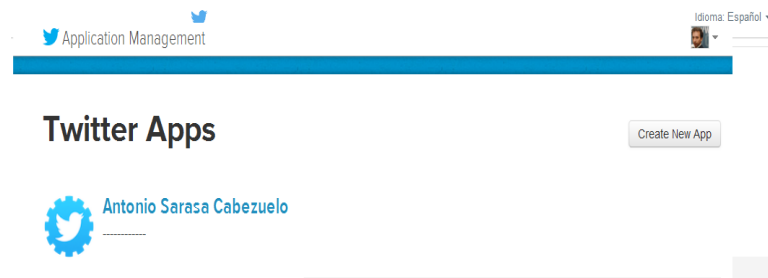
☒ Recordar mis datos · [¿Olvidaste tu contraseña?](#)

[¿Eres nuevo en Twitter? Regístrate ahora »](#)

[¿Ya usas Twitter por SMS? Activa tu cuenta »](#)

Documentos JSON

- Una vez autenticados, se entra en la página principal:



- Se selecciona “Create New App”, lo que abre un formulario que hay que rellenar.

Documentos JSON

Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information.

Documentos JSON

- Cuando se pulsa sobre “Create your Twitter application” aparece una nueva página con la información de la aplicación.

Prueba-19092015

Details Settings Keys and Access Tokens Permissions



Prueba-19092015

<http://www.Pruebagalia.es>

Organization

Information about the organization or company associated with your application. This information is optional.

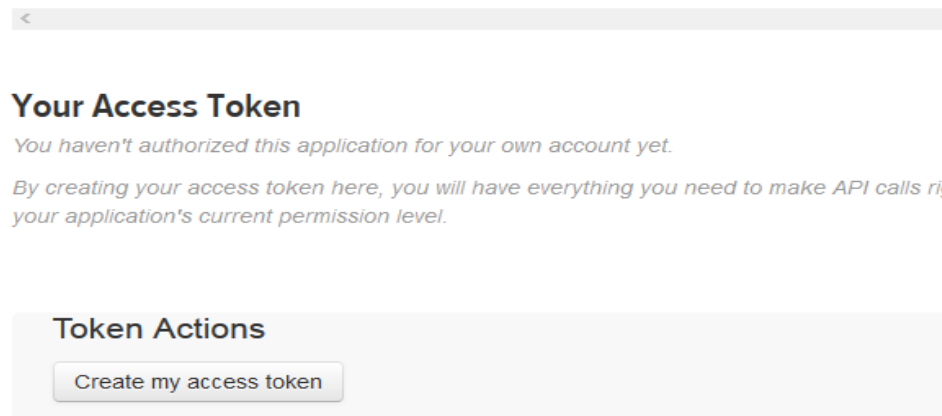
Organization None

Organization website None

< >

Documentos JSON

- Se pulsa sobre la solapa “Keys and Access content”, y en dicha página se busca la sección “Your access token” dónde se pulsa sobre la opción “create my access token”.



Documentos JSON

- Después de pulsar sobre la opción “create my access token” se han generado un conjunto de valores en la página.

Application Settings

Keep the “Consumer Secret” a secret. This key should never be human-readable in your applic

Consumer Key (API Key)	fm/	I9OvZ1IW
Consumer Secret (API Secret)	vor62ll0gl	lkqYqYhDoe8
Access Level	Read and write (modify app permissions)	
Owner	asarasa	
Owner ID	12400272	

Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not sha

Access Token	12400272- MCVfE	5dtmUJM5
Access Token Secret	nGcWNDSvi	GcjqlmDKTYnqWxhU
Access Level	Read and write	
Owner	asarasa	
Owner ID	12400272	

Documentos JSON

- Los valores que son necesarios utilizar para acceder a los datos de twitter son:
 - `api_key` = “Valor de la api key”
 - `api_secret` = “Valor de la api secret”
 - `access_token_key` = “Valor de la access token”
 - `access_token_secret` = “Valor de la access token secret”
- Una vez que se disponen de estos valores, se crea un programa en python para recuperar datos de Twitter.

Documentos JSON

- Se van a necesitar 4 programas:
 - oauth.py
 - twurl.py
 - hidden.py
 - twitter.py

Documentos JSON

- El programa `oauth.py` contiene una implementación del protocolo de firmas oauth.

Documentos JSON

- El programa hidden.py contiene los parámetros para acceder a la API de twitter que se han debido copiar de la app creada en la API en la sección de “Keys and Access content”.

```
# Keep this file separate

def oauth() :
    return { "consumer_key" : "Yfvmgqe Lqk2",
            "consumer_secret" : "D8Yj3VhkvK GpavQGapFw",
            "token_key" : "12400272-WPzOMPs I4j2vFGqIz",
            "token_secret" : "SwmzyRKOhH ChuDFEg" }
```

Documentos JSON

- El programa twurl es un programa auxiliar necesario para realizar la conexión.

```
import urllib
import oauth
import hidden

def augment(url, parameters) :
    secrets = hidden.oauth()
    consumer = oauth.OAuthConsumer(secrets['consumer_key'], secrets['consumer_secret'])
    token = oauth.OAuthToken(secrets['token_key'], secrets['token_secret'])

    oauth_request = oauth.OAuthRequest.from_consumer_and_token(consumer,
        token=token, http_method='GET', http_url=url, parameters=parameters)
    oauth_request.sign_request(oauth.OAuthSignatureMethod_HMAC_SHA1(), consumer, token)
    return oauth_request.to_url()
```

Documentos JSON

- El programa `twitter.py` contiene el procesamiento buscado. En este caso se recuperan los amigos de un usuario en Twitter, se analiza el JSON devuelto y se extrae parte de la información sobre esos amigos. También recupera información de las cabeceras de respuesta HTTP. En particular de `x-rate-limit-remaining` que informa sobre cuántas peticiones se pueden hacer antes de que ser bloqueados por un corto periodo de tiempo.

Documentos JSON

```
import urllib
import twurl
import json

TWITTER_URL = 'https://api.twitter.com/1.1/friends/list.j

while True:
    print ''
    acct = raw_input('Entrar cuenta de twitter:')
    if ( len(acct) < 1 ) : break
    url = twurl.augment(TWITTER_URL,
        {'screen_name': acct, 'count': '5'})
    print 'Recuperando', url
    connection = urllib.urlopen(url)
    data = connection.read()
    headers = connection.info().dict
    print 'Restante', headers['x-rate-limit-remaining']
    js = json.loads(data)
    print json.dumps(js, indent=4)
```

Documentos JSON

```
Entrar cuenta de twitter:asarasa
Recuperando https://api.twitter.com/1.1/friends/list.json?count=5&oauth_version=1.0&oauth_token=1240027
2-WPzOMPstfiQ0lxyS9ggQCCPAX6lORvgI4j2vFGqIz&screen_name=asarasa&oauth_nonce=33705127&oauth_timestamp=14
42684210&oauth_signature=sA2GHd0JvbbDj5xPHt8hDoRnmRw%3D&oauth_consumer_key=YfympqePkAEkAuo7joskXLqk2&oa
uth_signature_method=HMAC-SHA1
Restante 11
{
  "previous_cursor": 0,
  "previous_cursor_str": "0",
  "next_cursor": 1494817415528471951,
  "users": [
    {
      "follow_request_sent": false,
      "has_extended_profile": false,
      "profile_use_background_image": true,
      "default_profile_image": false,
      "id": 3001506986,
      "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png",
      "verified": false,
      "blocked_by": false,
      "profile_text_color": "333333",
      "muting": false,
      "profile_image_url_https": "https://pbs.twimg.com/profile_images/560822979178037248/JUnrE14
D_normal.jpeg",
      "profile_sidebar_fill_color": "DDEEF6",
      "entities": {
        "description": {
          "urls": []
        }
      }
    },
  ],
}
```