prog_datasci_5_api_entrega

November 17, 2020

0.1 Unidad 5: Adquisición de datos en Python

1 Fundamentos de Programación

En este Notebook se encontraréis el conjunto de actividades evaluables como PEC de la asignatura. Veréis que cada una de ellas tiene asociada una puntuación, que indica el peso que tiene la actividad sobre la nota final de la PEC. Adicionalmente, hay un ejercicio opcional, que no tiene puntuación dentro de la PEC, pero que se valora al final del semestre de cara a conceder las matrículas de honor y redondear las notas finales. Podréis sacar la máxima nota de la PEC sin necesidad de hacer este ejercicio. El objetivo de este ejercicio es que sirva como pequeño reto para los estudiantes que quieran profundizar en el contenido de la asignatura.

Veréis que todas las actividades de la PEC tienen una etiqueta, que indica los recursos necesarios para llevarla a cabo. Hay tres posibles etiquetas:

- NM **Sólo materiales**: las herramientas necesarias para realizar la actividad se pueden encontrar en los materiales de la asignatura.
- EG **Consulta externa guiada**: la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, pero el enunciado contiene indicaciones de dónde o cómo encontrar la información adicional necesaria para resolver la actividad.
- EI **Consulta externa independiente**: la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, y el enunciado puede no incluir la descripción de dónde o cómo encontrar esta información adicional. Será necesario que el estudiante busque esta información utilizando los recursos que se han explicado en la asignatura.

Es importante notar que estas etiquetas no indican el nivel de dificultad del ejercicio, sino únicamente la necesidad de consulta de documentación externa para su resolución. Además, recordad que las **etiquetas son informativas**, pero podréis consultar referencias externas en cualquier momento (aunque no se indique explícitamente) o puede ser que podáis hacer una actividad sin consultar ningún tipo de documentación. Por ejemplo, para resolver una actividad que sólo requiera los materiales de la asignatura, puedéis consultar referencias externas si queréis, ya sea tanto para ayudaros en la resolución como para ampliar el conocimiento!

En cuanto a la consulta de documentación externa en la resolución de los ejercicios, recordad **citar siempre la bibliografía utilizada** para resolver cada actividad.

1.0.1 Ejercicio 1

Hemos visto el uso de la libería Requests para realizar peticiones a web API de manera manual.

Mediante esta librería podemos realizar solicitudes como en el ejemplo que hemos visto de post-codes.io.

```
response = requests.get('http://api.postcodes.io/postcodes/E98%201TT')
```

Hemos visto que, en realizar una petición a una web API http, recuperamos un objeto que contiene, entre otros, los siguientes atributos: **status.code**, **content** y **headers**. Busca la información sobre los códigos de **status.code** y completa la siguiente tabla sobre los códigos de error http.

(1 punto) EG

Respuesta

Descripción de los principales códigos de error http:

- 200:
- 301:
- 400:
- 401:
- 403:
- 404:
- 505:
- 501:

1.0.2 Ejercicio 2

En este ejercicio intentaremos hacer una solicitud a tres paginas web diferentes vía el protocolo http mediante el método GET implementado en `requests.get`.

Obtén mediante `requests.get`, el contenido y el correspondiente `status.code` de las siguentes pàginas web:

- http://google.com
- http://wikipedia.org
- https://mikemai.net/
- http://google.com/noexisto

Para cada web, muestra:

- Los primeros 80 carácteres del contenido de la web
- El código de `status.code`.

(1.5 puntos) NM

[ ]:
```
# Respuesta
```

1.0.3 Ejercicio 3

En este ejercicio vamos a hacer un poco de *Fun with cats*. Existe una API para *cat-facts* (hechos sobre gatos) en la base de https://cat-fact.herokuapp.com. Esta API tiene dos puntos de acceso:

- **/facts**
- **/users**

Según la documentación, el modelo en el punto de entrada de un **fact** es tal y como se indica a continuación:

Key	Type	Description
_id	ObjectId	Unique ID for the Fact
__v	Number	Version number of the Fact
user	ObjectId	ID of the User who added the Fact
text	String	The Fact itself
updatedAt	Timestamp	Date in which Fact was last modified
sendDate	Timestamp	If the Fact is meant for one time use, this is the date that it is used
deleted	Boolean	Whether or not the Fact has been deleted (Soft deletes are used)
source	String (enum)	Can be 'user' or 'api', indicates who added the fact to the DB
used	Boolean	Whether or not the Fact has been sent by the CatBot. This value is reset each time every Fact is used
type	String	Type of animal the Fact describes (e.g. 'cat', 'dog', 'horse')

Así, para obtener el **fact** número *58e0086f0aac31001185ed02*, debemos construir una solicitud a la url:

- *https://cat-fact.herokuapp.com/facts/58e0086f0aac31001185ed02*

El objecto que se nos devolverá, contendrá la información indicada en la tabla en formato *json* serializado.

a) Contruye la solicitud, convierte el resultado a un diccionario y muestra por pantalla el resultado de los valores de la tabla anterior para el fact id *58e0086f0aac31001185ed02*.

(1.5 puntos) NM

[ ]:
```
# Respuesta
```

b) Para ara los fact ids:

- *5d38bdab0f1c57001592f156*
- *5ed11e643c15f700172e3856*
- *5ef556dff61f300017030d4c*
- *5d9d4ae168a764001553b388*

Obtén campos *type*, *user*, *user*, *source*, *used*, *text* y imprímelos siguiendo el siguiente formato:

```
Type: cat  User: 58e007480aac31001185ecef Used: True  Id:
58e0086f0aac31001185ed02 Source: https://www.scientificamerican.com/article/strange-but-true-ca
Text: Cats can't taste sweetness.
```

(2 puntos) NM

[ ]:
```
# Respuesta
```

1.1 Ejercicio 4

En los ejercicios anteriores, usamos directamente una API para hacer la solicitud que requiramos, y nos encargamos directamente de la gestión de los datos de salida.

No obstante, hemos visto ya el uso de librerías que facilitan el accesso a una API. La mayoría de estas librerías (y APIs de proyectos populares) requieren de un registro en la web de desarolladores.

Sigue la documentación proporcionada en clase para conseguir un registro en el panel de desarolladores de Twitter. Obtendrás 4 códigos para autenticar tu aplicación.

Usa la librería **tweepy** para programar dos funciones.

- La primera función, se autentica en la API de twitter usando los 4 códigos proporcionados por el registro. A partir de un nombre de usuario en twitter proporcionado en el argumento de la función, esta retorna una tupla `(user, api)` con el objeto `twepy.models.User`, correspondiente a ese usuario y el descriptor de la API ya inicializada.
- La segunda funcion, aceptará un objeto `twepy.models.User` de entrada y imprimirá:

1. El número de tweets del usuario.
2. El número de amigos del usuario.
3. El número de seguidores del usuario.
4. Los nombres de pantalla de los primeros 10 amigos del usuario (`screen_name`), sus nombres (`name`) junto con sus descripciones.

Ejecuta las dos funciones sobre el usuario de twitter **Space_Station**.

(2 puntos) EG

[ ]:
```
# Respuesta
```

1.1.1 Ejercicio 5

congreso.es es la página web del Congreso de los Diputados en España. En ella se guarda una relación de todos los diputados elegidos en cada una de las legislaturas.

En una de las páginas se puede observar un mapa del hemiciclo, junto con la posición de cada uno de los diputados, su fotografía, su representación territorial y el partido político al que esté adscrito. Esta url se encuentra en Hemiciclo.

Usad `scrappy` para extraer la siguiente información:

Nombre, *Territorio*, *Partido*, *URL Imagen*, en el formato de un diccionario, como por ejemplo:

```
{'Nombre': 'Callejas Cano, Juan Antonio ', 'Territorio': 'Diputado
por Ciudad Real', 'Partido': 'G.P. Popular en el Congreso', 'url':
'/wc/htdocs/web/img/diputados/peq/35_14.jpg'}
```

Para Ello:

- Utilizad el tutorial de scrappy para encontrar un `xpath` que contenga la información requerida
- Extraed la información requerida en forma de diccionario.

Nota: si la ejecución del *crawler* os devuelve un error `ReactorNotRestartable`, reiniciad el núcleo del Notebook (en el menú: `Kernel` - `Restart`)

(2 puntos) EI

[4]:
```
# Respuesta
```

1.1.2 Ejercicio opcional

Consultad la pagína web de Open Notify, indicando la información sobre los humanos residentes fuera de la tierra (es decir, en el espacio). Dirección url en Open Notify.

Codificad una función que imprima por pantalla el número total de astronautas en el espacio, numero de naves tripuladas actualmente en órbita, así como el nombre de los astronautas que habitan para cada una de estas naves.

EI

[ ]:
```
# Respuesta
```