Actividad Mongodb

Jaime García Lozano

10 de abril de 2023

1. Introducción

En esta actividad vamos a poner en práctica las consultas en Mongodb. Para ello, utilizaremos información disponible sobre el Inventario de Servicios Públicos Innovadores [1].

2. Creación del dataset

El primer paso es copiar el json guardado en un directorio local, en alguna de las carpetas del container donde tenemos Mongodb (tmp, por ejemplo):

```
docker cp cases.json mimongo:/tmp/.
```

Cabe destacar que la línea de código anterior ha de ser ejecutada dentro de la carpeta donde se encuentra el json. De lo contrario se ha de escribir su ruta completa.

Ahora entramos en el container *mimongo*:

```
docker exec -it mimongo bash
```

Una vez dentro, construimos la base de datos:

```
mongoimport --db trabajo --collection cases  --file /tmp/cases.json --jsonArray
```

Donde *db* es el nombre de la base de datos, *collection* el nombre de la colección (conjunto de documentos o elementos dentro del json), *file* el archivo a partir del cual generamos la colección y *jsonArray* especifica que el archivo tiene la forma de *array* de diccionarios.

Establecemos la conexión con el *host* local:

[4]:
```
from pymongo import MongoClient

client = MongoClient('127.0.0.1:27017')
```

Definimos el cursor, el cual nos permitirá movernos por la colección *cases*:

[5]:
```
cursor = client.trabajo.cases
```

Importamos la librería pprint para tener salidas visualmente más claras:

[6]:
```
from pprint import pprint as pp
```

Mostramos un ítem aleatorio para ver su estructura:

[7]:
```
item = cursor.find_one({})
pp(item)
```

```
{'_id': ObjectId('624add4bd497af65a293e883'),
 'active': 'Yes',
 'cid': 10001,
 'contact': 'mailto:supporto@blockchainregionelombardia.it',
 'cross_border': 'No',
 'cross_sector': 'Yes',
 'description': 'A public service, using blockchain technologies, which offers '
                'families from deprived backgrounds free access to childcare '
                'for children up to the age of 3.',
 'end_date': '',
 'entry_type': 'case',
 'geocoverage': ['Italy'],
 'geocoverage_codes': ['IT'],
 'geoextent': 'Regional',
 'id': 10001,
 'lead_organisation': {'category': 'Governmental', 'name': 'Regione Lombardia'},
 'lead_organisation_category': 'Governmental',
 'lead_organisation_name': 'Regione Lombardia',
 'name': 'Nidi gratis',
 'primary_sector': 'Social protection',
 'rtype': 'Service',
 'secondary_sector': 'Financial support',
 'start_date': '2019',
 'status': 'Pilot',
 'technology': 'Blockchain',
 'type': 'Process digitisation',
 'uptake': 'Small',
 'url': 'https://nidigratis.blockchainregionelombardia.it/'}
```

3. Consultas

1. **¿Qué país tiene un mayor número de servicios publicados? ¿En qué posición se encuentra España?**

[11]:
```
items=cursor.aggregate(
[ { "$match" : { "rtype" : "Service"}}, #filtrado de la variable de interés
{ "$group" : { "_id":"$geocoverage", "count": { '$sum': 1 }}},# agrupamos
 ↪por país
{"$sort": { "count":-1} # ordenamos de manera descendente
}]
);

for item in items:
    print(item)
```

```
{'_id': ['European Union'], 'count': 43}
{'_id': ['Netherlands'], 'count': 36}
{'_id': ['United Kingdom'], 'count': 33}
{'_id': ['Italy'], 'count': 25}
{'_id': ['France'], 'count': 23}
{'_id': ['Finland'], 'count': 20}
{'_id': ['Belgium'], 'count': 19}
{'_id': ['Spain'], 'count': 18}
{'_id': ['Germany'], 'count': 12}
{'_id': ['Denmark'], 'count': 11}
{'_id': ['Estonia'], 'count': 11}
{'_id': ['Sweden'], 'count': 9}
{'_id': ['Czechia'], 'count': 7}
{'_id': ['Austria'], 'count': 6}
{'_id': ['Ukraine'], 'count': 5}
{'_id': ['Norway'], 'count': 5}
{'_id': ['Slovenia'], 'count': 5}
{'_id': ['Ireland'], 'count': 5}
{'_id': ['Luxembourg'], 'count': 4}
{'_id': ['Malta'], 'count': 4}
{'_id': ['Latvia'], 'count': 4}
{'_id': ['Switzerland'], 'count': 4}
{'_id': ['Portugal'], 'count': 4}
{'_id': ['Poland'], 'count': 4}
{'_id': ['Greece'], 'count': 4}
{'_id': ['Cyprus'], 'count': 2}
{'_id': ['Slovakia'], 'count': 2}
{'_id': ['Lithuania'], 'count': 2}
{'_id': ['Hungary'], 'count': 2}
{'_id': ['Bulgaria'], 'count': 2}
{'_id': ['Iceland'], 'count': 1}
{'_id': ['Albania'], 'count': 1}
{'_id': ['Croatia'], 'count': 1}
{'_id': ['Romania'], 'count': 1}
{'_id': ['UK'], 'count': 1}
{'_id': ['Liechenstein'], 'count': 1}
{'_id': ['Czech Republic'], 'count': 1}
{'_id': ['Moldova'], 'count': 1}
```

2. **¿Cuántos tratan sobre "health"?**

[18]:
```
items=cursor.aggregate(
[ { "$match" : { "$and": [
                        {"rtype" : "Service"},
                        {"primary_sector": "Health"}]}}, #filtrado de la␣
 ↪variable de interés
```

```
{ '$group' : { '_id':"null", "count": { '$sum': 1 }}},# agrupamos por país
{"$sort": { "count":-1} # ordenamos de manera descendente
}]
);

for item in items:
    print(item)
```

```
{'_id': 'null', 'count': 38}
```

3. **Lista y contabiliza la frecuencia de tecnologías ("technology")**

[16]:
```
items=cursor.aggregate(
[{ '$group' : { '_id':"$technology", "count": { '$sum': 1 }}},# agrupamos␣
 ↪por tecnología
{"$sort": { "count":-1} # ordenamos de manera descendente
}]
);

for item in items:
    print(item)
```

```
{'_id': 'Api', 'count': 208}
{'_id': 'Blockchain', 'count': 66}
{'_id': 'Artificial intelligence', 'count': 59}
{'_id': 'Internet of things', 'count': 33}
{'_id': 'Data analytics', 'count': 14}
{'_id': 'Augmented reality / virtual reality', 'count': 5}
```

4. **Lista y contabiliza la frecuencia de ítems por año ("start_date"), en este caso gestiona los ítems sin este campo**

[17]:
```
items=cursor.aggregate(
[{ '$group' : { '_id':"$start_date", "count": { '$sum': 1 }}},# agrupamos␣
 ↪por tecnología
{"$sort": { "count":-1} # ordenamos de manera descendente
}]
);

for item in items:
    print(item)
```

```
{'_id': '2016', 'count': 61}
{'_id': '2018', 'count': 56}
{'_id': '2012', 'count': 43}
```

```
{'_id': '2017', 'count': 42}
{'_id': '2020', 'count': 35}
{'_id': '2015', 'count': 32}
{'_id': '2011', 'count': 26}
{'_id': '2014', 'count': 26}
{'_id': '2019', 'count': 20}
{'_id': '2013', 'count': 15}
{'_id': '2010', 'count': 14}
{'_id': '2021', 'count': 5}
{'_id': '2007', 'count': 4}
{'_id': '2008', 'count': 3}
{'_id': '2009', 'count': 2}
{'_id': '2005', 'count': 1}
```

[21]:
```
items =cursor.find( { "start_date": "null" } ); #return if start_date is␣
 ↪missing or null

for item in items:
    pp(item)
```

No hay ítems sin ese campo.

Referencias

[1] *JSON*, https://data.europa.eu/euodp/es/data/dataset/8d783268-6a51-4d61-a378-21605c563395.