Cas pràctic: accés a una API

PID_00271956

Àngel Ollé Blázquez

Temps mínim de dedicació recomanat: 1 hora

Universitat Oberta de Catalunya

Àngel Ollé Blázquez

Enginyer tècnic en Informàtica de gestió per la Universitat Rovira i Virgili (URV). Enginyer en Informàtica i màster en Programari Lliure per la Universitat Oberta de Catalunya (UOC). Actualment treballa com a enginyer de programari i participa en projectes *open source*. Anteriorment ha desenvolupat la seva activitat professional en el sector de serveis i consultoria IT per EMEA.

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Julià Minguillón Alfonso (2020)

Primera edició: febrer 2020

Av. Tibidabo, 39-43, 08035 Barcelona
Realització editorial: FUOC

Índex

1. Cas pràctic: accés a una API 5
1.1. Inici del cas pràctic 5
1.2. Tipologia de les dades del *dataset* 6
1.3. Cas pràctic fent servir SoQL 7

1. Cas pràctic: accés a una API

En aquest cas pràctic s'aprenen els aspectes fonamentals per a poder obtenir, transformar i treballar amb dades, i també extreure i interpretar els resultats.

D'entrada, s'explica com obtenir i filtrar les dades utilitzant l'API que proporciona el portal de transparència de la Generalitat de Catalunya referent a les contractacions públiques. Veurem com fer les consultes i obtenir les dades del servei de la Generalitat amb la nostra *shell*, i també com obtenir un primer conjunt de dades desitjades, ja filtrades des de l'inici via un model de consultes SQL suportat per l'API. Aquest procediment ens donarà una perspectiva delegant les tasques de cribratge de les dades al servidor o *back-end*.

Com que les dades seran obtingudes en format JSON, s'utilitzarà un processador JSON per línia de comandes en la segona fase de filtratge, tractament i transformació. Per acabar, s'extreuen conclusions i es fa una representació gràfica amb una aplicació per a fer traces.

1.1. Inici del cas pràctic

API SODA

Farem servir l'API SODA del portal de transparència de la Generalitat de Catalunya com a font de dades i consultes. La interacció amb l'API es farà mitjançant cURL i les consultes es passaran com a paràmetres GET a l'*endpoint*.

- https://analisi.transparenciacatalunya.cat/Sector-P-blic/Contractaci-de-Catalunya/hb6v-jcbf
- https://analisi.transparenciacatalunya.cat/resource/hb6v-jcbf.json

Manual de l'API

- https://dev.socrata.com/foundry/analisi.transparenciacatalunya.cat/hb6v-jcbf
- https://dev.socrata.com/consumers/getting-started.html

App Token

L'API té un nombre limitat de consultes anònimes. En els casos que es vulgui treballar realitzant un volum alt de consultes cal disposar d'una *app* Token que s'ha d'enviar com a 'header X-App-Token' en la petició.

Per a aconseguir una *app* Token ens hem de registrar a: https://analisi.transparenciacatalunya.cat/login seguint les instruccions, ens demana l'adreça de correu electrònic, el nom i la paraula de pas.

Un cop registrats, creem la nova clau de l'aplicació, al menú «Configuració del Desenvolupador»:

El *token* el podrem veure fent clic a «Mostrar el Símbol Secret».

El *token* s'ha d'enviar amb la petició https via el *header* X-App-Token. Amb cURL és:

```
$ curl -H "X-App-Token" <url>
```

Per a reproduir només aquest cas pràctic no cal disposar de *token*, ja que les consultes que es fan són poques. Però si volem experimentar amb l'API, es recomana disposar de *token* per a tenir una restricció més relaxada de les quotes.

1.2. Tipologia de les dades del *dataset*

- Cada fila és un contracte.
- Columnes: https://analisi.transparenciacatalunya.cat/Sector-P-blic/Contractaci-de-Catalunya/hb6v-jcbf
- Apartat columnes. Es pot veure la descripció de cadascuna.

Exemple de dada (json):


```
{
"situaci_contractual":"menor",
"exercici":"2016",
"subjecte_ambit":"Departaments i Sector Públic de la Generalitat de Catalunya",
"id_agrupacio_organisme":"1500",
"agrupacio_organisme":"DEPARTAMENT DE SALUT",
"id_organisme_contractant":"1539",
"organisme_contractant":"Institut Català de la Salut (ICS) Costa de Ponent Hospitalària",
"codi_expedient":"11006163031AH02",
"procediment_adjudicacio":"Menor",
"tipus_contracte":"3. SUBMINISTRAMENTS",
"descripcio_expedient":"Altres recanvis equipaments serveis generals",
"numero_lot":"1",
"codi_cpv":"44510000-8",
"adjudicatari":"DEXTROMEDICA, SL",
"import_adjudicacio":"575",
"data_adjudicacio":"2016-10-28T00:00:00.000",
"contracte":"Altres recanvis equipaments serveis generals",
"lot_desert":"N",
"dies_durada":"4",
```

```
"mesos_durada":"2",
"anys_durada":"0"
}
```

SoQL (*Socrata Query Language*) i DataTypes

- https://dev.socrata.com/docs/queries/index.html
- https://dev.socrata.com/docs/datatypes/

Eines addicionals

Per al desenvolupament d'aquest cas pràctic hem de disposar del JSON processor *-jq-* i de Gnuplot per a fer traces gràfiques. En distribucions basades en Debian els podem instal·lar amb *apt*:

```
# apt install gnuplot jq
```

1.3. Cas pràctic fent servir SoQL

Amb les consultes a l'API extraurem la despesa total per organisme de l'any 2018 i visualitzarem les dades amb Gnuplot.

Podem fer directament la consulta fent servir l'SoQL (*Socrata Query Language*). Aquest llenguatge de consulta SQL ens permet recuperar les dades desitjades directament del *dataset*.

SoQL Import total dels contractes de l'any 2018 per organisme:

- exercici = 2018
- import total = sum(import_adjudicacio)
- organisme = agrupar per agrupacio_organisme

Consulta:

```
$select=agrupacio_organisme,SUM(import_adjudicacio) as import_total
$where=exercici = 2018
$group=agrupacio_organisme
$order=agrupacio_organisme
```

La consulta es pot executar proporcionant tots els arguments un a un o amb la *query* sencera. En aquest exemple, utilitzarem la *query* sencera.

```
query="SELECT agrupacio_organisme,SUM(import_adjudicacio) as import_total WHERE exercici = 2018
GROUP BY agrupacio_organisme ORDER BY agrupacio_organisme"

query=${query// /%20}

$ curl "https://analisi.transparenciacatalunya.cat/resource/hb6v-jcbf.json?\$query=$query"
-o resultat.json
```

L'*endpoint* SODA no gestiona les peticions amb espais en blanc sense codificar. Si enviem una petició via cURL en què la *query* té espais en blanc sense codificar, rebrem una resposta del tipus: «HTTP Bad Request estat 400».

Per aquest motiu, utilitzem les expansions de Bash per a reemplaçar de la consulta totes les ocurrències d'espais en blanc per la seva codificació que és *%20* segons https://tools.ietf.org/html/rfc3986.

Una altra manera d'enviar la consulta codificada és utilitzant l'opció *data-urlencode* del cURL perquè sigui ell mateix el que faci la codificació.

```
query="SELECT agrupacio_organisme,SUM(import_adjudicacio) as import_total WHERE exercici = 2018
GROUP BY agrupacio_organisme ORDER BY agrupacio_organisme"

$ curl -G "https://analisi.transparenciacatalunya.cat/resource/hb6v-jcbf.json" --data-urlencode
"\$query=$query" -o resultat.json
```

Si executem cURL amb l'opció *-v* veurem que la petició que s'envia té la forma següent:

```
GET /resource/hb6v-jcbf.json?$query=SELECT%20agrupacio_organisme%2CSUM%28import_adjudicacio
%29%20as%20import_total%20WHERE%20exercici%20%3D%202018%20GROUP%20BY%20agrupacio_organisme
%20ORDER%20BY%20agrupacio_organisme HTTP/1.1
```

Els resultats de la consulta s'han desat al fitxer *result.json*:

```
$ head -10 resultat.json
[{"agrupacio_organisme":"Agència de Desenvolupament del Berguedà","import_total":"318621.49"}
,{"agrupacio_organisme":"Ajuntament d'Abrera","import_total":"5172356.77"}
,{"agrupacio_organisme":"Ajuntament d'Agramunt","import_total":"907731.31"}
,{"agrupacio_organisme":"Ajuntament d'Aiguaviva","import_total":"120255.30"}
,{"agrupacio_organisme":"Ajuntament d'Aitona","import_total":"1013262.56"}
,{"agrupacio_organisme":"Ajuntament d'Albinyana","import_total":"197522.61"}
,{"agrupacio_organisme":"Ajuntament d'Alcanar","import_total":"182454"}
,{"agrupacio_organisme":"Ajuntament d'Alcanó","import_total":"12850.14"}
,{"agrupacio_organisme":"Ajuntament d'Alcover","import_total":"602221.21"}
,{"agrupacio_organisme":"Ajuntament d'Alella","import_total":"1392624.05"}
```

El resultat està ordenat alfabèticament per nom de l'organisme.

Tot i que podem esbrinar-ho llençant diverses consultes contra l'*endpoint* de l'API SODA, en el nostre cas extraurem algunes conclusions amb les eines que disposem via *shell*.

L'organisme amb més despesa l'any 2018 ha estat el Departament de Salut amb 1.046.317.033,91 €:

```
$ jq -c 'max_by(.import_total | tonumber)' resultat.json
{"agrupacio_organisme":"DEPARTAMENT DE SALUT","import_total":"1046317033.91"}
```

Amb el JSON processor (*jq*) traiem el màxim pel camp *import_total*. Per a fer la comparació interna per a trobar el màxim, transformen l'*string* a número, ja que *import_total* és una *string* (el valor està entre cometes) i així fem que la comparació sigui numèrica.

L'organisme amb menys despesa de l'any 2018 ha estat l'Ajuntament d'Horta de Sant Joan amb 1.164,42 €:

```
$ jq -c 'min_by(.import_total | tonumber)' resultat.json
{"agrupacio_organisme":"Ajuntament d'Horta de Sant Joan","import_total":"1164.42"}
```

Despesa total en contractacions:

```
$ printf %.2f€ $(jq -c '[.[] | .import_total | tonumber] | reduce .[] as $imp(0; .+$imp)'
resultat.json)
3738977991.29€
```

o

```
$ jq -c '[.[] | .import_total | tonumber] | reduce .[] as $imp(0; .+$imp)'
resultat.json | xargs printf %.2f€
3738977991.29€
```

Finalment, generarem una representació gràfica del total de les despeses només per als departaments de l'Administració de la Generalitat, per la qual cosa s'han de filtrar les dades.

Generem el CSV amb les dades desitjades:

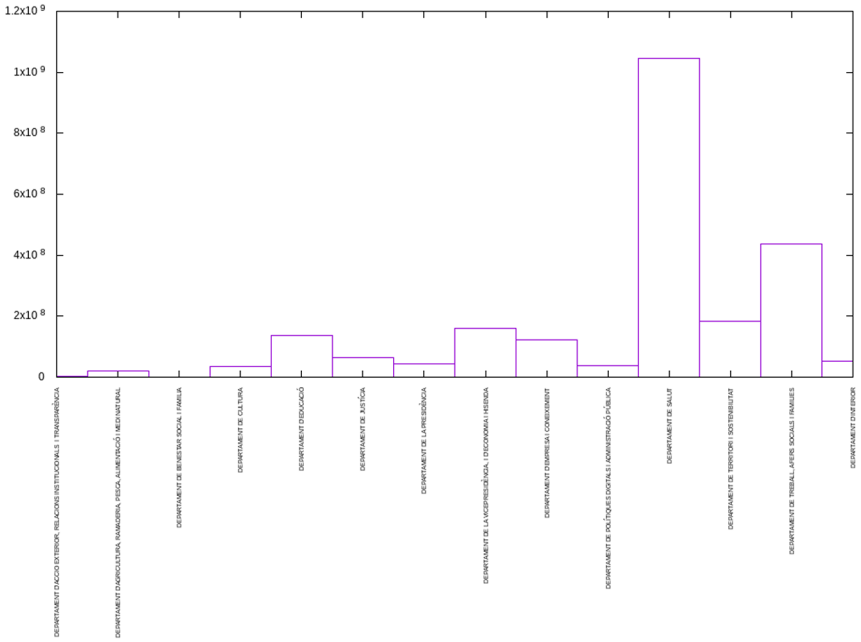
```
$ jq -r '[.[] | select(.agrupacio_organisme | startswith("DEPARTAMENT"))] |
map([.agrupacio_organisme,.import_total])[] | @csv' resultat.json > dades.csv
```

dades.csv:

```
$ cat dades.csv
"DEPARTAMENT D'ACCIO EXTERIOR, RELACIONS INSTITUCIONALS I TRANSPARÈNCIA","3193313.61"
"DEPARTAMENT D'AGRICULTURA, RAMADERIA, PESCA, ALIMENTACIÓ I MEDI NATURAL","20061323.39"
"DEPARTAMENT DE BENESTAR SOCIAL I FAMILIA","1449344.00"
"DEPARTAMENT DE CULTURA","35754603.95"
"DEPARTAMENT D'EDUCACIÓ","137935559.52"
"DEPARTAMENT DE JUSTÍCIA","63723085.14"
"DEPARTAMENT DE LA PRESIDÈNCIA","43174240.05"
"DEPARTAMENT DE LA VICEPRESIDÈNCIA, I D'ECONOMIA I HISENDA","160223625.48"
"DEPARTAMENT D'EMPRESA I CONEIXEMENT","122659833.41"
"DEPARTAMENT DE POLÍTIQUES DIGITALS I ADMINISTRACIÓ PÚBLICA","37554632.60"
"DEPARTAMENT DE SALUT","1046317033.91"
"DEPARTAMENT DE TERRITORI I SOSTENIBILITAT","182531973.78"
"DEPARTAMENT DE TREBALL, AFERS SOCIALS I FAMILIES","438234273.50"
"DEPARTAMENT D'INTERIOR","51877601.73"
```

Executem el Gnuplot i configurem per a crear un gràfic de barres:

```
$ gnuplot
<..>
Terminal type is now 'qt'
gnuplot> set output 'grafica.png'
gnuplot> set title 'Import de les contractacions per Departament'
gnuplot> set datafile separator comma
gnuplot> set xtics rotate
gnuplot> set xtics font ",5.0"
gnuplot> set ytics font ",9.0"
gnuplot> set size 1, 0.5
gnuplot> set term png size 1024, 768
gnuplot> plot 'dades.csv' using 2:xtic(1) notitle with boxes
```


1.2x10 9
1x10 9
8x10 8
6x10 8
4x10 8
2x10 8
0
DEPARTAMENT D'ACCIÓ EXTERIOR, RELACIONS INSTITUCIONALS I TRANSPARÈNCIA
DEPARTAMENT D'AGRICULTURA, RAMADERIA, PESCA, ALIMENTACIÓ I MEDI NATURAL
DEPARTAMENT DE BENESTAR SOCIAL I FAMÍLIA
DEPARTAMENT DE CULTURA
DEPARTAMENT D'EDUCACIÓ
DEPARTAMENT DE JUSTÍCIA
DEPARTAMENT DE LA PRESIDÈNCIA
DEPARTAMENT DE LA VICEPRESIDÈNCIA, I D'ECONOMIA I HISENDA
DEPARTAMENT D'EMPRESA I CONEIXEMENT
DEPARTAMENT DE POLÍTIQUES DIGITALS I D'ADMINISTRACIÓ PÚBLICA
DEPARTAMENT DE SALUT
DEPARTAMENT DE TERRITORI I SOSTENIBILITAT
DEPARTAMENT DE TREBALL, AFERS SOCIALS I FAMÍLIES
DEPARTAMENT D'INTERIOR