
Caso práctico: acceso a una API

PID_00271958

Àngel Ollé Blázquez

Tiempo mínimo de dedicación recomendado: 1 hora



**Àngel Ollé Blázquez**

Ingeniero técnico en Informática de gestión por la Universidad Rovira i Virgili (URV). Ingeniero en Informática y máster en Software Libre por la Universitat Oberta de Catalunya (UOC). Actualmente trabaja como ingeniero de software y participa en proyectos *open source*. Anteriormente desarrolló su actividad profesional en el sector de servicios y consultoría IT por EMEA.

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Julià Minguillón Alfonso (2020)

Primera edición: febrero 2020
© Àngel Ollé Blázquez
Todos los derechos reservados
© de esta edición, FUOC, 2020
Av. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea este eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares de los derechos.

Índice

1. Caso práctico: acceso a una API.....	5
1.1. Inicio del caso práctico	5
1.2. Tipología de los datos del <i>dataset</i>	6
1.3. Caso práctico utilizando SoQL	7

1. Caso práctico: acceso a una API

En este caso práctico se van a aprender los aspectos fundamentales para poder obtener, transformar y trabajar con datos, y también extraer e interpretar los resultados.

De entrada, se explica cómo obtener y filtrar los datos utilizando la API que proporciona el portal de transparencia de la Generalitat de Cataluña referente a las contrataciones públicas. Veremos cómo realizar las consultas y obtener los datos del servicio de la Generalitat con nuestra *shell*, y también cómo obtener un primer conjunto de datos deseados, ya filtrados desde el inicio a través de un modelo de consultas SQL soportado por la API. Este procedimiento nos proporcionará una perspectiva, delegando las tareas de cribado de los datos al servidor o *back-end*.

Dado que los datos se obtendrán en formato JSON, se utilizará un procesador JSON para línea de comandos en la segunda fase de filtrado, tratamiento y transformación. Para acabar, se extraerán conclusiones y se realizará una representación gráfica con una aplicación para hacer gráficos.

1.1. Inicio del caso práctico

API SODA

Utilizaremos la API SODA del portal de transparencia de la Generalitat de Cataluña como fuente de datos y consultas. La interacción con la API se realizará mediante cURL y las consultas se pasarán como parámetros GET al *endpoint*.

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

Manual de la API

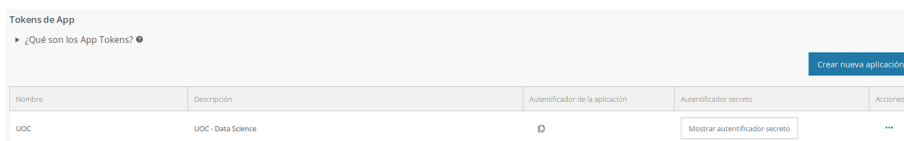
- <https://dev.socrata.com/foundry/analisi.transparenciacatalunya.cat/hb6v-jcbf>
- <https://dev.socrata.com/consumers/getting-started.html>

App Token

La API dispone de un número limitado de consultas anónimas. En los casos en los que se desee trabajar realizando un volumen alto de consultas, es necesario disponer de una *app* Token, que tiene que ser enviada como 'header X-App-Token' en la petición.

Para obtener una *app* Token, nos hemos de registrar en: <https://analisi.transparenciacatalunya.cat/es/login>. Siguiendo las instrucciones, nos pide la dirección de correo electrónico, el nombre y la contraseña.

Una vez registrados, creamos la nueva contraseña de la aplicación en el menú «Configuración del Desarrollador»:



Nombre	Descripción	Autentificador de la aplicación	Autentificador secreto	Acciones
UOC	UOC - Data Science	D	Mostrar autentificador secreto	...

Podremos ver el *token* clicando en «Mostrar Autentificador Secreto».

El *token* tiene que ser enviado con la petición https a través del *header* X-App-Token. Con cURL es:

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

Para reproducir solo este caso práctico, no hace falta disponer de *token*, ya que las consultas que se van a realizar son escasas. Pero si queremos experimentar con la API, se recomienda disponer de *token* para tener una restricción más relajada de las cuotas.

1.2. Tipología de los datos del *dataset*

- Cada fila es un contrato.
- Columnas: <https://analisi.transparenciacatalunya.cat/es/Sector-P-blic/Contractaci-de-Catalunya/hb6v-jcbf>
- Apartado columnas. Se puede ver la descripción de cada una de ellas.

Ejemplo de dato (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) y DataTypes

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

Herramientas adicionales

Para el desarrollo de este caso práctico, debemos contar con el JSON procesador (*jq*) y con Gnuplot para hacer gráficos. En distribuciones basadas en Debian, los podemos instalar con *apt*:

```
# apt install gnuplot jq
```

1.3. Caso práctico utilizando SoQL

Con las consultas a la API obtendremos el gasto total por organismo del año 2018 y visualizaremos los datos con Gnuplot.

Podemos realizar la consulta directamente usando el SoQL (*Socrata Query Language*). Este lenguaje de consulta SQL nos permite recuperar los datos deseados directamente del *dataset*.

SoQL Import total de los contratos del año 2018 por organismo:

- ejercicio = 2018
- importe total = SUM(import_adjudicacio)
- organismo = agrupar por agrupacio_organisme

Consulta:

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

Se puede ejecutar la consulta proporcionando todos los argumentos uno a uno o con la *query* entera. En este ejemplo, utilizaremos la *query* entera.

```
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
```

El *endpoint* SODA no gestiona las peticiones con espacios en blanco sin codificar. Si enviamos una petición vía cURL en la que la *query* tiene espacios en blanco sin codificar, recibiremos una respuesta del tipo: «HTTP Bad Request status 400».

Por este motivo, utilizaremos las expansiones de Bash para reemplazar en la consulta todas las ocurrencias de espacios en blanco por su codificación, que es *%20* según <https://tools.ietf.org/html/rfc3986>.

Otra manera de enviar la consulta codificada es utilizando la opción *data-urlencode* del cURL para que sea él mismo el que haga la codificación.

```
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 ejecutamos cURL con la opción *-v* veremos que la petición que se envía tiene la forma siguiente:

```
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
```

Se han dejado los resultados de la consulta en el fichero *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 resultado está ordenado alfabéticamente por nombre del organismo.

Aunque podemos averiguarlo lanzando varias consultas contra el *endpoint* de la API SODA, en nuestro caso sacaremos algunas conclusiones con las herramientas de que disponemos vía *shell*.

El organismo con más gasto en el año 2018 fue el Departamento de Salud, con 1.046.317.033,91 €:

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


Con el JSON processor (*jq*) sacamos el máximo para al campo *import_total*. Para hacer la comparación interna con la idea de encontrar este máximo, transformamos el *string* a número, ya que *import_total* es una *string* (el valor está entre comillas) y así hacemos que la comparación sea numérica.

El organismo con menos gasto en el año 2018 fue el Ayuntamiento de Horta de Sant Joan, con 1.164,42 €:

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

Gasto total en contrataciones:

```
$ 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€
```

Finalmente, generaremos una representación gráfica del total de los gastos solo para los departamentos de la Administración de la Generalitat, por lo que los datos deben filtrarse.

Generamos el CSV con los datos deseados:

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

datos.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"
```

Ejecutamos el Gnuplot y lo configuramos para crear un gráfico de barras:

```
$ gnuplot
<..>
Terminal type is now 'qt'
gnuplot> set output 'grafica.png'
gnuplot> set title 'Importe de las contrataciones por Departamento'
gnuplot> set datafile separator comma
gnuplot> set xtics rotate
gnuplot> set xtics font ",5.0"
```

