HOW-TO: HARVESTING LARGE QUANTITY OF DATA FROM ARCGIS REST SERVICES USING A NEW TOOL!

If you have ever worked with ArcGIS REST Services, either through ArcGIS Online or ArcGIS Server, you might have faced a limitation of 1000 features by response, and you wouldn't be the only one! Googling "1000 limit on ArcGIS REST service" would already give you a good idea of how much this has been a recurring issue for users for quite some time.

This limitation is defined by the maxRecordCount **service property and it is set to 1000 by default**. As mentioned by Esri, this implies that only that specific number of "records can be returned by the server to a client for any given query operation". Therefore, **as users, we are unable to access a whole dataset with a single query**, making us search for workarounds which could allow to perform either a global analysis of the dataset or operations needing access to all records at once, like a "Find Closest Facility".

I. Why are services de facto limited?

There are some valid reasons explaining such default value. The main one is to ensure an acceptable performance level for both clients but also servers. As it is explained by Esri, the data we are handling is quite heavy and take a lot of memory space in one's browser, especially if we are talking about polygons. Without the 1,000 graphics limit, we could easily reach the capacity of the server to process and send requests, and the user's browser to smoothly display a map including so many features.

Having said that, some of the complains on the matter are dated from more than five years ago, so one could wonder if the technology hasn't advanced enough to increase the 1,000 graphics limit. Besides, the limitation was built around the idea of a worst-case scenario, i.e. using a polygon layer, when one often only needs points (a couple of numbers) or simple tabular data. In any case, if you are the data publisher, it is hard to say what could be a reasonable number today, as it depends on each environment and use case.

II. Yes… but give me all!

One shouldn't worry! Even if we cannot increase the limitation of the service we are using, either because of technical constraints or because we are not the server maintainers, **there are still several methods to query all the data at once**.

II.1. Grouped object IDs approach

One of them, which is mentioned by Esri, has to do with **performing sequential queries to groups of 1000 Objectids** (or as many Objectids as the limitation is set).

"While there is a limit to the number of features included in the feature set response, there is no limit to the number of object IDs returned in the ID array response. Clients can exploit this to get all the query conforming object IDs by specifying *returnIdsOnly=true* and subsequently requesting feature sets for subsets of object IDs." (Esri).

II.2. The resultOffset approach

A **second method** that we have been using at CartONG is to **take advantage of the** _resultOffset_ **query parameter** (available since 10.3), which allows to "skip the specified number of records and [to return response] starting from the next record" (Esri). If we iterate queries, while each time adding the number of returned records (which should match with the service limitation) to the _resultOffset_ parameter, we then end up obtaining the full dataset. **This is exactly what is done by the ArcGIS REST Service export tool created by CartONG and which is introduced in the next section.**

Anyone interested on using or consulting the source code of the tool can have a look at its GitHub repository. Special mention to the _CartONG.ArcGISService.js_ file, where the implementation of the method explained above is coded, (_CartONG.ArcgisService.prototype.query_ function).

III. Introducing the ArcGIS REST Service export tool

Here below is a short introduction on how to best use the tool developed by CartONG to help users extract datasets from ArcGIS services. As of today, the layout is quite simple and limited in terms of options, with only three input options for the user. Yet we hope to keep developing the tool to add further functionalities in the future.

1- If needed, change the default print format given to any of the available ones (Esri JSON, GeoJSON and CSV as of today). This is the format in which the results will be showing in the grey box below ("Print results").

2- Optionally, if you don't need geometries, set Return geometry to "No", as it will slow down the response time. Keep in mind that polygon services are quite heavy and response could take a long time.

3- Enter the URL of the Map Service layer of interest, including its layer index (the number at the end of the URL below), and click on the search button on the right or press "Enter" on your keyboard.

- Careful: make sure you don't add a trailing slash (/).
- Also, if you don't know the index of the layer, access the service URL and check the index of available layers.

World_Cities (FeatureServer)

View In: ArcGIS.com Map

Service Description:

Service ItemId: 6996f03a1b364dbab4008d99380370ed

Has Versioned Data: false

Max Record Count: 2000

Supported query Formats: JSON

Supports applyEdits with GlobalIds: False

All Layers and Tables

Layers:

- World_Cities (0)

Description:

Copyright Text:

4- After a while, if the query is successful, the results will be inputted in the grey box.

Print results   Success! | Number of features: 2540

```
}
},
{
"attributes":{
"FID": 3,
"ObjectID": 501,
"CITY_NAME": "Nashville",
"GMI_ADMIN": "USA-TNN",
"ADMIN_NAME": "Tennessee",
```

5- Finally, if necessary, an export option is available in two formats, the elected one at step 1 (or Esri JSON if CSV was chosen) and CSV.

Export ▾

As CSV

As Esri JSON
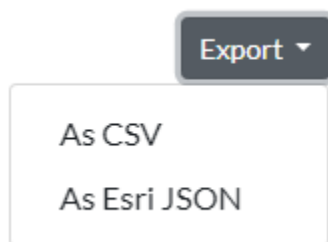
ArcGIS services are used in all sectors, be it the private, public or humanitarian one. Overcoming the service limits does thus imply a big advantage for any use case. This is why the ArcGIS REST Service Export tool is a truly cross-cutting support tool for data consumers.