ARLAS

The ARLAS API makes the ARLAS catalog available for exploration and browsing. The catalog contains collections of geo-referenced elements. Every element has a geometry, a centroid, a timestamp and a set of fields specific to the collection.

URL Schema

The table below lists the URL endpoints and their optional "parts". A part is composed of optional parameters. The parameters are seperated with the character &.

PATH Template	Description
/arlas/explore/**_list**	List the collections configured in ARLAS
/arlas/explore/{collection}/**_describe**?form	Describe the structure and the content of the given collection
/arlas/explore/{collection}/**_count**?filter & form	Count the number of elements found in the collection, given the filters
/arlas/explore/{collection}/**_search**?filter & form & projection & size & sort	Search and return the elements found in the collection, given the filters
/arlas/explore/{collection}/**_geosearch**?filter & form & projection & size & sort	Search and return the elements found in the collection as features, given the filters
/arlas/explore/{collections}/**_aggregate**?aggregation &filter & form	Aggregate the elements in the collection(s), given the filters and the aggregation parameters
/arlas/explore/{collections}/**_geoaggregate**?aggregation &filter & form & size & sort	Aggregate the elements in the collection(s) as features, given the filters and the aggregation parameters
/arlas/explore/{collections}/**_suggest**?filter & form & size & suggest	Suggest the the n (n=size) most relevant terms given the filters

When multiple collections are permitted ({collections}), the comma is used for separating the collection names.

Examples
https://api.gisaia.com/demo/arlas/explore/_describe
https://api.gisaia.com/demo/arlas/explore/city,state,country/_describe
https://api.gisaia.com/demo/arlas/explore/city,state,country/_count?q=bord*&f=country:France&pretty=true&human=true
https://api.gisaia.com/demo/arlas/explore/election/_search?f=country:France&after=1490613808&format=geojson&pretty=true&human=true&size=1000&include=id,name
https://api.gisaia.com/demo/arlas/explore/election/_aggregate?f=country:France&after=1490613808&format=geojson&pretty=true&human=true&size=1000&include=id,name&agg=geohash&agg_interval=4

URL Parts

Part: `aggregation`

The [`aggregation`] url part allows the following parameters to be specified:

Parameter	Default value	Description	Multiple
agg	None	Gathers a set of sub-parameters indicating the type of aggregation, the field used as the aggregation key and possibly the interval for numeric values	true for _aggregate only

The agg parameter should be given in the following format :

- {type}:{field}:interval-{interval}:format-{format}:collect_field-{collect_field}:collect_fct-{function}:order-{order}:on-{on}:size-{size}

Where the `{type}:{field}` part is mandatory

The other parts must be specified or not depending on the aggregation type. All the cases are sum up in the following table.

Parameter	Aggregation type	Description
interval	`datehistogram, histogram, geohash`	mandatory
format	`datehistogram`	optional (default value : `yyyy-MM-dd-HH:mm:ss`)
(**collect_field**,**collect_fct**)	All types	optional
(**order**,**on**)	`term, histogram, datehistogram`	optional
size	`term, geohash`	optional

Example: `agg=datehistogram:date:interval-20day:format-dd.MM.yyyy&agg=term:sexe:collect_field-age:collect_fct-avg:order-asc:on-result:size-5`

The sub-parameters possible values are:

Parameter	Values	Description
{type}	`datehistogram`, `histogram`, `geohash`, `term`	Type of aggregation
{field}	{field}	Aggregates on {field}
interval	{interval}	Size of the intervals.(1)
format	Date format (https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket-daterange-aggregation.html#date-format-pattern) for key aggregation	Date format for key aggregation.
collect_field	`{collect_field}`	The field used to aggregate collections.
collect_fct	`avg,cardinality,max,min,sum`	The aggregation function to apply to collections on the specified **collect_field**.
order	`asc,desc`	Sorts the aggregation buckets on the field name, on the count of the buckets or on the the result of a metric sub-aggregation,

		ascending or descending.
on	`field,count,result` (2)	{on} is set to specify whether the **order** is on the field name, on the count of the aggregation or the result of a metric subaggregation.
size	{size}	Defines how many buckets should be returned.

(1) Each aggregation type ({type}) has its own type of interval. The table below lists the semantic of the interval sub-parameter.

(2) When **on** is `result`, then (**collect_field**,**collect_fct**) should be specified

Service	Aggregation type	Interval	Description
_aggregate	***datehistogram***	{size} (year,quarter,month,week,day,hour,minute,second)	Size of a time interval with the given unit (no space between number and unit). Size must be equal to 1 for year, quarter and month
_geoaggregate	***geohash***	{length}	The geohash length: lower the length, greater is the surface of aggregation. See table below.
_aggregate	***histogram***	{size}	The interval size of the numeric aggregation
_aggregate	***term***	None	None

The table below shows the metric dimensions for cells covered by various string lengths of geohash. Cell dimensions vary with latitude and so the table is for the worst-case scenario at the equator.

GeoHash length	Area width x height

1	5,009.4km x 4,992.6km
2	1,252.3km x 624.1km
3	156.5km x 156km
4	39.1km x 19.5km
5	4.9km x 4.9km
6	1.2km x 609.4m
7	152.9m x 152.4m
8	38.2m x 19m
9	4.8m x 4.8m
10	1.2m x 59.5cm
11	14.9cm x 14.9cm
12	3.7cm x 1.9cm

agg parameter is multiple. Every agg parameter specified is a subaggregation of the previous one : the order matters.

For **_geoaggregate** service, the first (main) aggregation must be geohash.

Part: `filter`

The `filter` url part allows the following parameters to be specified:

Parameter	Default value	Values	Description	Multiple
f	None	`{fieldName}{operator}`	A triplet for filtering the result. Multiple filter can be provided. The order does not matter. A triplet is composed of a field name, a comparison operator and a value. The **AND** operator is applied between filters. For the **:** (equal) filter, values can be comma separated ({field}:{v1},{v2}) which stands for an **OR**.	true

		{value}	For the **:ne:** (not equal) filter, values can be comma separated ({field}:ne:{v1},{v2}) which stands for an **AND**	
q	None	text	A full text search	false
before	None	timestamp	Any element having its point in time reference before the given timestamp	false
after	None	timestamp	Any element having its point in time reference after the given timestamp	false
pwithin	None	geometry	Any element having its centroid contained within the given BBOX	false
gwithin	None	geometry	Any element having its geometry contained within the given geometry	false
gintersect	None	geometry	Any element having its geometry intersecting the given geometry (WKT)	false

Operator	Description	Value type
:	{fieldName} equals {comma separated values}. **OR** operation is applied for the specified values	numeric or strings
:ne:	{fieldName} must not equal {comma separated values }. **AND** operation is applied for the specified values	numeric or strings
:like:	{fieldName} is like {value}.	numeric or strings
:gte:	{fieldName} is greater than or equal to {value}	numeric
:gt:	{fieldName} is greater than {value}	numeric
:lte:	{fieldName} is less than or equal to {value}	numeric
:lt:	{fieldName} is less than {value}	numeric

Example: f=city:Toulouse&f=city:Bordeaux&after=1490613808&

Part: `form`

The `form` url part allows the following parameters to be specified:

Parameter	Default value	Values	Description	Multiple
pretty	`false`	`true,false`	Pretty print	false
human	`false`	`true,false`	Human readable print	false

Example: `pretty=true&human=true`

Part: `format`

The `format` url part allows the following parameters to be specified:

Parameter	Default value	Values	Description	Multiple
format	`false`	`json,geojson`	JSON or GeoJSON format	false

Example: `format=geojson`

Part: `projection`

The `projection` url part allows the following parameters to be specified:

Parameter	Default value	Values	Description	Multiple

include	*	`{fieldNamePattern}`	List the name patterns of the field to be included in the result. Seperate patterns with a comma.	true
exclude	*	`{fieldNamePattern}`	List the name patterns of the field to be excluded in the result. Seperate patterns with a comma.	true

Example: `include=*&exclude=city,state`

Part: `suggest`

The `suggest` url part allows the following parameters to be specified:

Parameter	Default value	Values	Description	Multiple
field	`_all`	`{fieldName}`	Name of the field to be used for retrieving the most relevant terms	false

Example: `field=recommended`

Part: `size`

The `size` url part allows the following parameters to be specified:

Parameter	Default value	Values	Description	Multiple
size	10	>0	The maximum number of entries or sub-entries to be returned.	false
from	0	>0	From index to start the search from. Defaults to 0.	false

Example: `size=1000`

Part: `sort`

The `sort` url part allows the following parameters to be specified:

Parameter	Default value	Values	Description	Multiple
sort	None	`((-?){field})(,(-?){field})*`	Sort the result on the given fields ascending or descending. Fields can be provided several times by separating them with a comma. The order matters. For a descending sort, precede the field with '-'. The sort will be ascending otherwise. For aggregation, provide the agg keyword as the `{field}`.	false (separate fields with comma in the same parameter)

Example: `sort=-country,city`