ARLAS Exploration API

Table of Contents

Chapter 1. Overview

Explore the content of ARLAS collections

1.1. Version information

Version : V0.1.0

1.2. Contact information

Contact : Gisaia
Contact Email : contact@gisaia.com

1.3. License information

License : Apache 2.0
License URL : https://www.apache.org/licenses/LICENSE-2.0.html
Terms of service : null

1.4. URI scheme

BasePath : /arlas
Schemes : HTTP

1.5. Tags

- collections
- explore

Chapter 2. Resources

2.1. Collections

2.1.1. Get all collection references

```
GET /collections
```

Description

Get all collection references in ARLAS

Responses

HTTP Code	Description	Schema
200	Successful operation	< CollectionReference > array
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.1.2. Get a collection reference

```
GET /collections/{collection}
```

Description

Get a collection reference in ARLAS

Parameters

Type	Name	Description	Schema
Path	**collection** *required*	collection	string

Responses

HTTP Code	Description	Schema
200	Successful operation	CollectionReference
404	Collection not found.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.1.3. Add a collection reference

```
PUT /collections/{collection}
```

Description

Add a collection reference in ARLAS

Parameters

Type	Name	Description	Schema
Path	**collection** *required*	collection	string
Body	**collectionParams** *required*	collectionParams	CollectionReferenceParameters

Responses

HTTP Code	Description	Schema
200	Successful operation	CollectionReference
400	JSON parameter malformed.	ArlasError
404	Not Found Error.	ArlasError

HTTP Code	Description	Schema
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.1.4. Delete a collection reference

```
DELETE /collections/{collection}
```

Description

Delete a collection reference in ARLAS

Parameters

Type	Name	Description	Schema
Path	**collection** *required*	collection	string

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasSuccess
404	Collection not found.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2. Explore

2.2.1. List

```
GET /explore/_list
```

Description

List the collections configured in ARLAS.

Parameters

Type	Name	Description	Schema
Query	**max-age-cache** *optional*	max-age-cache	integer(int32)

Responses

HTTP Code	Description	Schema
200	Successful operation	CollectionReferenceDescription
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.2. Suggest

```
GET /explore/{collections}/_suggest
```

Description

Suggest the the n (n=size) most relevant terms given the filters

Parameters

Type	Name	Description	Schema	Default
Path	**collections** *required*	collections, comma separated	string	
Query	**after** *optional*	Any element having its point in time reference after the given timestamp	integer(int64)	
Query	**before** *optional*	Any element having its point in time reference before the given timestamp	integer(int64)	

Type	Name	Description	Schema	Default
Query	**f** *optional*	* 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 possible values of the comparison operator are : ---- Operator | Description | value type : | {fieldName} equals {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 ---- * The AND operator is applied between filters having different fieldNames. * The OR operator is applied on filters having the same fieldName. * If the fieldName starts with - then a must not filter is used * If the fieldName starts with - then a must not filter is used For more details, check https://gitlab.com/GISAIA.ARLAS/ARLAS-server/blob/master/doc/api/API-definition.md	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**field** *optional*	Name of the field to be used for retrieving the most relevant terms	string	`"_all"`
Query	**from** *optional*	From index to start the search from. Defaults to 0.	integer(int32)	`"0"`
Query	**gintersect** *optional*	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	**gwithin** *optional*	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	**human** *optional*	Human readable print	boolean	`"false"`
Query	**max-age-cache** *optional*	max-age-cache	integer(int32)	
Query	**notgintersect** *optional*	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	**notgwithin** *optional*	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	**notpwithin** *optional*	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	**q** *optional*	A full text search	string	
Query	**size** *optional*	The maximum number of entries or sub-entries to be returned. The default value is 10	integer(int32)	`"10"`

Responses

HTTP Code	Description	Schema
200	Successful operation	No Content

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.3. Aggregate

```
POST /explore/{collection}/_aggregate
```

Description

Aggregate the elements in the collection(s), given the filters and the aggregation parameters

Parameters

Type	Name	Description	Schema
Path	**collection** *required*	collection	string
Body	**body** *optional*		AggregationRequest

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasAggregation
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.4. Aggregate

```
GET /explore/{collection}/_aggregate
```

Description

Aggregate the elements in the collection(s), given the filters and the aggregation parameters

Parameters

Type	Name	Description	Schema	Default
Path	**collection** *required*	collection	string	
Query	**after** *optional*	Any element having its point in time reference after the given timestamp	string	

Type	Name	Description	Schema	Default
Query	**agg** *required*	* The agg parameter should be given in the following formats: {type}:{field}:interval-{interval}:format-{format}:collect_field-{collect_field}:collect_fct-{function}:order-{order}:on-{on}:size-{size} Where : * **{type}:{field}** part is mandatory. * **interval** must be specified only when aggregation type is datehistogram, histogram and geohash. * **format** is optional for datehistogram, and must not be specified for the other types. * (**collect_field**,**collect_fct**) couple is optional for all aggregation types. * (**order**,**on**) couple is optional for all aggregation types. * **size** is optional for term and geohash, and must not be specified for the other types. * {type} possible values are : datehistogram, histogram, geohash and term. * {interval} possible values depends on {type}. If {type} = datehistogram, then {interval} = {size}(year,quarter,month,week,day,hour,minute,second). If {type} = histogram, then {interval} = {size}. If {type} = geohash, then {interval} = {size}. It's an integer between 1 and 12. Lower the length, greater is the surface of aggregation. If {type} = term, then interval-{interval} is not needed. * format-{format} is the date format for key	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**before** *optional*	Any element having its point in time reference before the given timestamp	string	

Type	Name	Description	Schema	Default
Query	**f** *optional*	* 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 possible values of the comparison operator are : ---- Operator | Description | value type :eq: | {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 :lt: | {fieldName} is less than {value} | numeric ---- * The AND operator is applied between filters having different fieldNames. * The OR operator is applied on filters having the same fieldName. * If the fieldName starts with - then a must not filter is used	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**gintersect** *optional*	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	**gwithin** *optional*	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	**human** *optional*	Human readable print	boolean	`"false"`
Query	**max-age-cache** *optional*	max-age-cache	integer(int32)	
Query	**notgintersect** *optional*	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	**notgwithin** *optional*	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	**notpwithin** *optional*	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	**q** *optional*	A full text search. Optionally, it's possible to search on a field using this syntax: {fieldname}:{text}	string	

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasAggregation
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.5. Count

```
POST /explore/{collection}/_count
```

Description

Count the number of elements found in the collection(s), given the filters

Parameters

Type	Name	Description	Schema
Path	**collection** *required*	collections	string
Body	**body** *optional*		Count

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasHits
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.6. Count

```
GET /explore/{collection}/_count
```

Description

Count the number of elements found in the collection(s), given the filters

Parameters

Type	Name	Description	Schema	Default
Path	**collection** *required*	collections	string	
Query	**after** *optional*	Any element having its point in time reference after the given timestamp	string	
Query	**before** *optional*	Any element having its point in time reference before the given timestamp	string	

Type	Name	Description	Schema	Default
Query	**f** *optional*	* 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 possible values of the comparison operator are : ---- Operator | Description | value type :eq: | {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 :lt: | {fieldName} is less than {value} | numeric ---- * The AND operator is applied between filters having different fieldNames. * The OR operator is applied on filters having the same fieldName. * If the fieldName starts with - then a must not filter is used	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**gintersect** *optional*	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	**gwithin** *optional*	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	**human** *optional*	Human readable print	boolean	`"false"`
Query	**max-age-cache** *optional*	max-age-cache	integer(int32)	
Query	**notgintersect** *optional*	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	**notgwithin** *optional*	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	**notpwithin** *optional*	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	**q** *optional*	A full text search. Optionally, it's possible to search on a field using this syntax: {fieldname}:{text}	string	

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasHits
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.7. Describe

```
GET /explore/{collection}/_describe
```

Description

Describe the structure and the content of the given collection.

Parameters

Type	Name	Description	Schema	Default
Path	**collection** *required*	collection	string	
Query	**human** *optional*	Human readable print	boolean	`"false"`
Query	**max-age-cache** *optional*	max-age-cache	integer(int32)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`

Responses

HTTP Code	Description	Schema
200	Successful operation	CollectionReferenceDescription
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.8. GeoAggregate

```
POST /explore/{collection}/_geoaggregate
```

Description

Aggregate the elements in the collection(s), given the filters and the aggregation parameters

Parameters

Type	Name	Description	Schema
Path	**collection** *required*	collection	string
Body	**body** *optional*		AggregationRequest

Responses

HTTP Code	Description	Schema
200	Successful operation	FeatureCollection
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError
501	Not implemented functionality.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.9. GeoAggregate

```
GET /explore/{collection}/_geoaggregate
```

Description

Aggregate the elements in the collection(s), given the filters and the aggregation parameters

Parameters

Type	Name	Description	Schema	Default
Path	**collection** *required*	collection	string	

Type	Name	Description	Schema	Default
Query	**after** *optional*	Any element having its point in time reference after the given timestamp	string	

Type	Name	Description	Schema	Default
Query	**agg** *required*	* The agg parameter should be given in the following formats: {type}:{field}:interval-{interval}:format-{format}:collect_field-{collect_field}:collect_fct-{function}:order-{order}:on-{on}:size-{size} Where : * **{type}:{field}** part is mandatory. * **interval** must be specified only when aggregation type is datehistogram, histogram and geohash. * **format** is optional for datehistogram, and must not be specified for the other types. * (**collect_field**,**collect_fct**) couple is optional for all aggregation types. * (**order**,**on**) couple is optional for all aggregation types. * **size** is optional for term and geohash, and must not be specified for the other types. * {type} possible values are : geohash, datehistogram, histogram and term. geohash must be the main aggregation. * {interval} possible values depends on {type}. If {type} = datehistogram, then {interval} = {size}(year,quarter,month,week,day,hour,minute,second). If {type} = histogram, then {interval} = {size}. If {type} = geohash, then {interval} = {size}. It's an integer between 1 and 12. Lower the length, greater is the surface of aggregation. If {type} = term, then interval-{interval} is not needed.	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**before** *optional*	Any element having its point in time reference before the given timestamp	string	

Type	Name	Description	Schema	Default
Query	**f** *optional*	* 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 possible values of the comparison operator are : ---- Operator | Description | value type :eq: | {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 :lt: | {fieldName} is less than {value} | numeric ---- * The AND operator is applied between filters having different fieldNames. * The OR operator is applied on filters having the same fieldName. * If the fieldName starts with - then a must not filter is used	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**gintersect** *optional*	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	**gwithin** *optional*	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	**human** *optional*	Human readable print	boolean	`"false"`
Query	**max-age-cache** *optional*	max-age-cache	integer(int32)	
Query	**notgintersect** *optional*	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	**notgwithin** *optional*	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	**notpwithin** *optional*	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	**q** *optional*	A full text search. Optionally, it's possible to search on a field using this syntax: {fieldname}:{text}	string	

Responses

HTTP Code	Description	Schema
200	Successful operation	FeatureCollection
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError
501	Not implemented functionality.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.10. GeoSearch

```
POST /explore/{collection}/_geosearch
```

Description

Search and return the elements found in the collection(s) as features, given the filters

Parameters

Type	Name	Description	Schema
Path	**collection** *required*	collection	string
Body	**body** *optional*		Search

Responses

HTTP Code	Description	Schema
200	Successful operation	FeatureCollection
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.11. GeoSearch

```
GET /explore/{collection}/_geosearch
```

Description

Search and return the elements found in the collection(s) as features, given the filters

Parameters

Type	Name	Description	Schema	Default
Path	**collection** *required*	collection	string	
Query	**after** *optional*	Any element having its point in time reference after the given timestamp	string	
Query	**before** *optional*	Any element having its point in time reference before the given timestamp	string	
Query	**exclude** *optional*	List the name patterns of the field to be excluded in the result. Seperate patterns with a comma.	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**f** *optional*	* 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 possible values of the comparison operator are : ---- Operator | Description | value type :eq: | {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 :lt: | {fieldName} is less than {value} | numeric ---- * The AND operator is applied between filters having different fieldNames. * The OR operator is applied on filters having the same fieldName. * If the fieldName starts with - then a must not filter is used	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**from** *optional*	From index to start the search from. Defaults to 0.	string	`"0"`
Query	**gintersect** *optional*	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	**gwithin** *optional*	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	**human** *optional*	Human readable print	boolean	`"false"`
Query	**include** *optional*	List the name patterns of the field to be included in the result. Seperate patterns with a comma.	< string > array(multi)	
Query	**max-age-cache** *optional*	max-age-cache	integer(int32)	
Query	**notgintersect** *optional*	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	**notgwithin** *optional*	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	**notpwithin** *optional*	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	**q** *optional*	A full text search. Optionally, it's possible to search on a field using this syntax: {fieldname}:{text}	string	
Query	**size** *optional*	The maximum number of entries or sub-entries to be returned. The default value is 10	string	`"10"`
Query	**sort** *optional*	* 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.	< string > array(multi)	

Responses

HTTP Code	Description	Schema
200	Successful operation	FeatureCollection
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.12. Search

```
POST /explore/{collection}/_search
```

Description

Search and return the elements found in the collection, given the filters

Parameters

Type	Name	Description	Schema
Path	**collection** *required*	collection	string
Body	**body** *optional*		Search

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasHits
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

2.2.13. Search

```
GET /explore/{collection}/_search
```

Description

Search and return the elements found in the collection, given the filters

Parameters

Type	Name	Description	Schema	Default
Path	**collection** *required*	collection	string	
Query	**after** *optional*	Any element having its point in time reference after the given timestamp	string	
Query	**before** *optional*	Any element having its point in time reference before the given timestamp	string	
Query	**exclude** *optional*	List the name patterns of the field to be excluded in the result. Seperate patterns with a comma.	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**f** *optional*	* 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 possible values of the comparison operator are : ---- Operator | Description | value type :eq: | {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 :lt: | {fieldName} is less than {value} | numeric ---- * The AND operator is applied between filters having different fieldNames. * The OR operator is applied on filters having the same fieldName. * If the fieldName starts with - then a must not filter is used	< string > array(multi)	

Type	Name	Description	Schema	Default
Query	**from** *optional*	From index to start the search from. Defaults to 0.	string	`"0"`
Query	**gintersect** *optional*	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	**gwithin** *optional*	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	**human** *optional*	Human readable print	boolean	`"false"`
Query	**include** *optional*	List the name patterns of the field to be included in the result. Seperate patterns with a comma.	< string > array(multi)	
Query	**max-age-cache** *optional*	max-age-cache	integer(int32)	
Query	**notgintersect** *optional*	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	**notgwithin** *optional*	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	**notpwithin** *optional*	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	**q** *optional*	A full text search. Optionally, it's possible to search on a field using this syntax: {fieldname}:{text}	string	
Query	**size** *optional*	The maximum number of entries or sub-entries to be returned. The default value is 10	string	`"10"`
Query	**sort** *optional*	* 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.	string	

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasHits
400	Bad request.	ArlasError
500	Arlas Server Error.	ArlasError

Consumes

- `application/json;charset=utf-8`

Produces

- `application/json;charset=utf-8`

Chapter 3. Definitions

3.1. AggregationModel

Name	Schema
collectFct *optional*	string
collectField *optional*	string
field *optional*	string
format *optional*	string
interval *optional*	string
order *optional*	string
size *optional*	string
true *optional*	string
type *optional*	string

3.2. AggregationRequest

Name	Schema
aggregations *optional*	Aggregations
filter *optional*	Filter
form *optional*	Form

3.3. Aggregations

Name	Schema
aggregations *optional*	< AggregationModel > array

3.4. ArlasAggregation

Aggregation result

Name	Description	Schema
arlastime *optional*	All the aggregation execution time. It includes query time.	integer(int64)
count *optional*	Count	integer(int64)
elements *optional*	Sub-aggregations	< ArlasAggregation > array
key *optional*	Key	object
key_as_string *optional*	Key as string	object
metric aggregation *optional*	Metric aggregation	ArlasMetric
name *optional*	Name	string
querytime *optional*	Query execution time	integer(int64)
totalnb *optional*	Total number of hits matching the query	integer(int64)

3.5. ArlasError

Name	Schema
error *optional*	string
message *optional*	string
status *optional*	integer(int32)

3.6. ArlasHit

A hit retrieved from an ARLAS Collection

Name	Description	Schema
data *optional*	The hit's data	object
md *optional*	The hit's metadata	ArlasMD

3.7. ArlasHits

A collection of hits retrieved from ARLAS Collections

Name	Description	Schema
hits *optional*	ARLAS hits	< ArlasHit > array
nbhits *optional*	Number of hits contained in hits	integer(int64)
totalnb *optional*	Total number of hits matching the query	integer(int64)

3.8. ArlasMD

Metadata of the ARLAS hit

Name	Description	Schema
centroid *optional*	The centroid of the hit	object
geometry *optional*	The geometry of the hit	object
id *optional*	The unique identifier of the hit	string
timestamp *optional*	The timestamp of the hit	integer(int64)

3.9. ArlasMetric

Metric agg

Name	Description	Schema
field *optional*	field of the metric aggregation	string
type *optional*	Name of the metric aggregation	string
value *optional*	Value of the metric aggregation	number(double)

3.10. ArlasSuccess

Name	Schema
message *optional*	string
status *optional*	integer(int32)

3.11. CollectionReference

The reference to ARLAS collection that embed elasticsearch index description.

Name	Description	Schema
collection_name *optional*	The collection name	string
params *optional*	The collection parameters	CollectionReferenceParameters

3.12. CollectionReferenceDescription

Describe the structure and the content of the given collection.

Name	Description	Schema
collection_name *optional*	The collection name	string
params *optional*	The collection parameters	CollectionReferenceParameters

Name	Description	Schema
properties *optional*	The collection fields	object

3.13. CollectionReferenceParameters

The description of the elasticsearch index and the way ARLAS API will serve it.

Name	Description	Schema
centroid_path *optional*	Path to the collection's centroid **Example** : `"centroid"`	string
exclude_fields *optional*	List the name patterns of the fields to be excluded in the result. Seperate patterns with a comma. **Example** : `"fieldname"`	string
geometry_path *optional*	Path to the collection's geometry **Example** : `"geometry"`	string
id_path *optional*	Path to the collection's id **Example** : `"id"`	string
include_fields *optional*	List the name patterns of the fields to be included in the result. Seperate patterns with a comma. **Example** : `"*"`	string
index_name *optional*	The collection's index name	string
timestamp_path *optional*	Path to the collection's timestamp **Example** : `"timestamp"`	string
type_name *optional*	The collection's type name	string

3.14. Count

Name	Schema
filter *optional*	Filter
form *optional*	Form

3.15. Crs

Name	Schema
properties *optional*	< string, object > map
type *optional*	enum (name, link)

3.16. Feature

Name	Schema
bbox *optional*	< number(double) > array
crs *optional*	Crs
geometry *optional*	GeoJsonObject
id *optional*	string
properties *optional*	< string, object > map

3.17. FeatureCollection

Name	Schema
bbox *optional*	< number(double) > array
crs *optional*	Crs
features *optional*	< Feature > array

3.18. Filter

Name	Schema
after *optional*	integer(int64)

Name	Schema
before *optional*	integer(int64)
f *optional*	< string > array
gintersect *optional*	string
gwithin *optional*	string
notgintersect *optional*	string
notgwithin *optional*	string
notpwithin *optional*	string
pwithin *optional*	string
q *optional*	string

3.19. Form

Name	Schema
human *optional*	boolean
pretty *optional*	boolean

3.20. GeoJsonObject

Name	Schema
bbox *optional*	< number(double) > array
crs *optional*	Crs

3.21. GeometryCollection

Polymorphism : Inheritance
Discriminator : type

Name	Schema
bbox *optional*	< number(double) > array
crs *optional*	Crs
geometries *optional*	< GeoJsonObject > array

3.22. LineString

Polymorphism : Inheritance
Discriminator : type

Name	Schema
bbox *optional*	< number(double) > array
coordinates *optional*	< LngLatAlt > array
crs *optional*	Crs

3.23. LngLatAlt

Name	Schema
additionalElements *optional*	< number(double) > array
altitude *optional*	number(double)
latitude *optional*	number(double)
longitude *optional*	number(double)

3.24. MultiLineString

Polymorphism : Inheritance
Discriminator : type

Name	Schema
bbox *optional*	< number(double) > array
coordinates *optional*	< < LngLatAlt > array > array
crs *optional*	Crs

3.25. MultiPoint

Polymorphism : Inheritance
Discriminator : type

Name	Schema
bbox *optional*	< number(double) > array
coordinates *optional*	< LngLatAlt > array
crs *optional*	Crs

3.26. MultiPolygon

Polymorphism : Inheritance
Discriminator : type

Name	Schema
bbox *optional*	< number(double) > array
coordinates *optional*	< < < LngLatAlt > array > array > array
crs *optional*	Crs

3.27. Point

Polymorphism : Inheritance
Discriminator : type

Name	Schema
bbox *optional*	< number(double) > array
coordinates *optional*	LngLatAlt
crs *optional*	Crs

3.28. Polygon

Polymorphism : Inheritance
Discriminator : type

Name	Schema
bbox *optional*	< number(double) > array
coordinates *optional*	< < LngLatAlt > array > array
crs *optional*	Crs

3.29. Search

Name	Schema
filter *optional*	Filter
form *optional*	Form
size *optional*	Size
sort *optional*	Sort

3.30. Size

Name	Schema
from *optional*	integer(int32)
size *optional*	integer(int32)

3.31. Sort

Name	Schema
sort *optional*	string