

ARLAS Exploration API

Table of Contents

1. Overview	1
1.1. Version information	1
1.2. Contact information.....	1
1.3. License information.....	1
1.4. URI scheme.....	1
1.5. Tags	1
2. Resources	2
2.1. Collections	2
2.1.1. Get all collection references	2
2.1.2. Get a collection reference	2
2.1.3. Add a collection reference	3
2.1.4. Delete a collection reference	4
2.2. Explore	5
2.2.1. List	5
2.2.2. Suggest.....	5
2.2.3. Aggregate	9
2.2.4. Aggregate	9
2.2.5. Count	15
2.2.6. Count	15
2.2.7. Describe	19
2.2.8. GeoAggregate.....	19
2.2.9. GeoAggregate.....	20
2.2.10. GeoSearch.....	26
2.2.11. GeoSearch.....	26
2.2.12. Search	30
2.2.13. Search	31
2.2.14. Get an Arlas document	34
3. Definitions	36
3.1. AggregationModel	36
3.2. AggregationRequest.....	36
3.3. Aggregations	36
3.4. ArlasAggregation	37
3.5. ArlasError	37
3.6. ArlasHit	38
3.7. ArlasHits.....	38
3.8. ArlasMD	38

3.9. ArlasMetric	38
3.10. ArlasSuccess	39
3.11. CollectionReference	39
3.12. CollectionReferenceDescription	39
3.13. CollectionReferenceDescriptionProperty	40
3.14. CollectionReferenceParameters	40
3.15. Count	41
3.16. Crs	41
3.17. Feature	41
3.18. FeatureCollection	42
3.19. Filter	42
3.20. Form	43
3.21. GeoJsonObject	43
3.22. GeometryCollection	43
3.23. LineString	44
3.24. LngLatAlt	44
3.25. MultiLineString	44
3.26. MultiPoint	45
3.27. MultiPolygon	45
3.28. Point	45
3.29. Polygon	46
3.30. Search	46
3.31. Size	46
3.32. Sort	46

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 <i>required</i>	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 <i>required</i>	collection	string
Body	collectionParams <i>required</i>	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 <i>required</i>	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 <i>optional</i>	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 <i>required</i>	collections, comma separated	string	
Query	after <i>optional</i>	Any element having its point in time reference after the given timestamp	integer(int64)	
Query	before <i>optional</i>	Any element having its point in time reference before the given timestamp	integer(int64)	

Type	Name	Description	Schema	Default														
Query	f optional	<p>* A triplet for filtering the result. Multiple filter can be provided. The order does not matter.</p> <p>* A triplet is composed of a field name, a comparison operator and a value.</p> <p>The possible values of the comparison operator are :</p> <p>----</p> <table><thead><tr><th>Operator</th><th>Description</th></tr></thead><tbody><tr><td> value type</td><td></td></tr><tr><td>:</td><td> {fieldName} equals {value} numeric or strings</td></tr><tr><td>:gte:</td><td> {fieldName} is greater than or equal to {value} numeric</td></tr><tr><td>:gt:</td><td> {fieldName} is greater than {value} numeric</td></tr><tr><td>:lte:</td><td> {fieldName} is less than or equal to {value} numeric</td></tr><tr><td>:lt:</td><td> {fieldName} is less than {value} numeric</td></tr></tbody></table> <p>----</p> <p>* The AND operator is applied between filters having different fieldNames.</p> <p>* The OR operator is applied on filters having the same fieldName.</p> <p>* If the fieldName starts with - then a must not filter is used</p> <p>* If the fieldName starts with ~ then a must not filter is used</p> <p>For more details, check https://gitlab.com/GISAIA.ARLAS/ARLAS-server/blob/master/doc/api/API-definition.md</p>	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	< string > array(multi)	
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																	

Type	Name	Description	Schema	Default
Query	field <i>optional</i>	Name of the field to be used for retrieving the most relevant terms	string	"_all"
Query	from <i>optional</i>	From index to start the search from. Defaults to 0.	integer(int32)	"0"
Query	gintersect <i>optional</i>	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	gwithin <i>optional</i>	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	human <i>optional</i>	Human readable print	boolean	"false"
Query	max-age-cache <i>optional</i>	max-age-cache	integer(int32)	
Query	notgintersect <i>optional</i>	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	notgwithin <i>optional</i>	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	notpwithin <i>optional</i>	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	pretty <i>optional</i>	Pretty print	boolean	"false"
Query	pwithin <i>optional</i>	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	q <i>optional</i>	A full text search	string	
Query	size <i>optional</i>	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 <i>required</i>	collection	string
Body	body <i>optional</i>		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 <i>required</i>	collection	string	
Query	after <i>optional</i>	Any element having its point in time reference after the given timestamp	integer	

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

Type	Name	Description	Schema	Default
Query	before <i>optional</i>	Any element having its point in time reference before the given timestamp	integer	

Type	Name	Description	Schema	Default						
Query	f optional	<p>* A triplet for filtering the result. Multiple filter can be provided. The order does not matter.</p> <p>* A triplet is composed of a field name, a comparison operator and a value.</p> <p>The possible values of the comparison operator are :</p> <p>----</p> <table><tr><th>Operator </th><th>Description</th><th></th></tr><tr><td>value type</td><td></td><td></td></tr></table> <p>:eq: {fieldName} equals {comma separated values}. OR operation is applied for the specified values numeric or strings</p> <p>:ne: {fieldName} must not equal {comma separated values }. AND operation is applied for the specified values numeric or strings</p> <p>:like: {fieldName} is like {value} numeric or strings</p> <p>:gte: {fieldName} is greater than or equal to {value} numeric</p> <p>:gt: {fieldName} is greater than {value} numeric</p> <p>:lte: {fieldName} is less than or equal to {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>----</p> <p>* The AND operator is applied between filters having different fieldNames.</p> <p>* The OR operator is applied on filters having the same fieldName.</p> <p>* If the fieldName starts with - then a must not filter is used</p>	Operator	Description		value type			< string > array(multi)	
Operator	Description									
value type										

Type	Name	Description	Schema	Default
Query	gintersect <i>optional</i>	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	gwithin <i>optional</i>	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	human <i>optional</i>	Human readable print	boolean	"false"
Query	max-age-cache <i>optional</i>	max-age-cache	integer(int32)	
Query	notgintersect <i>optional</i>	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	notgwithin <i>optional</i>	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	notpwithin <i>optional</i>	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	pretty <i>optional</i>	Pretty print	boolean	"false"
Query	pwithin <i>optional</i>	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	q <i>optional</i>	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 <i>required</i>	collections	string
Body	body <i>optional</i>		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 <i>required</i>	collections	string	
Query	after <i>optional</i>	Any element having its point in time reference after the given timestamp	integer	
Query	before <i>optional</i>	Any element having its point in time reference before the given timestamp	integer	

Type	Name	Description	Schema	Default						
Query	f optional	<p>* A triplet for filtering the result. Multiple filter can be provided. The order does not matter.</p> <p>* A triplet is composed of a field name, a comparison operator and a value.</p> <p>The possible values of the comparison operator are :</p> <p>----</p> <table><tr><th>Operator </th><th>Description</th><th></th></tr><tr><td>value type</td><td></td><td></td></tr></table> <p>:eq: {fieldName} equals {comma separated values}. OR operation is applied for the specified values numeric or strings</p> <p>:ne: {fieldName} must not equal {comma separated values }. AND operation is applied for the specified values numeric or strings</p> <p>:like: {fieldName} is like {value} numeric or strings</p> <p>:gte: {fieldName} is greater than or equal to {value} numeric</p> <p>:gt: {fieldName} is greater than {value} numeric</p> <p>:lte: {fieldName} is less than or equal to {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>----</p> <p>* The AND operator is applied between filters having different fieldNames.</p> <p>* The OR operator is applied on filters having the same fieldName.</p> <p>* If the fieldName starts with - then a must not filter is used</p>	Operator	Description		value type			< string > array(multi)	
Operator	Description									
value type										

Type	Name	Description	Schema	Default
Query	gintersect <i>optional</i>	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	gwithin <i>optional</i>	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	human <i>optional</i>	Human readable print	boolean	"false"
Query	max-age-cache <i>optional</i>	max-age-cache	integer(int32)	
Query	notgintersect <i>optional</i>	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	notgwithin <i>optional</i>	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	notpwithin <i>optional</i>	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	pretty <i>optional</i>	Pretty print	boolean	"false"
Query	pwithin <i>optional</i>	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	q <i>optional</i>	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 <i>required</i>	collection	string	
Query	human <i>optional</i>	Human readable print	boolean	"false"
Query	max-age-cache <i>optional</i>	max-age-cache	integer(int32)	
Query	pretty <i>optional</i>	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 <i>required</i>	collection	string
Body	body <i>optional</i>		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 <i>required</i>	collection	string	

Type	Name	Description	Schema	Default
Query	after <i>optional</i>	Any element having its point in time reference after the given timestamp	integer	

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

Type	Name	Description	Schema	Default
Query	before <i>optional</i>	Any element having its point in time reference before the given timestamp	integer	

Type	Name	Description	Schema	Default						
Query	f optional	<p>* A triplet for filtering the result. Multiple filter can be provided. The order does not matter.</p> <p>* A triplet is composed of a field name, a comparison operator and a value.</p> <p>The possible values of the comparison operator are :</p> <p>----</p> <table><tr><th>Operator </th><th>Description</th><th></th></tr><tr><td>value type</td><td></td><td></td></tr></table> <p>:eq: {fieldName} equals {comma separated values}. OR operation is applied for the specified values numeric or strings</p> <p>:ne: {fieldName} must not equal {comma separated values }. AND operation is applied for the specified values numeric or strings</p> <p>:like: {fieldName} is like {value} numeric or strings</p> <p>:gte: {fieldName} is greater than or equal to {value} numeric</p> <p>:gt: {fieldName} is greater than {value} numeric</p> <p>:lte: {fieldName} is less than or equal to {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>----</p> <p>* The AND operator is applied between filters having different fieldNames.</p> <p>* The OR operator is applied on filters having the same fieldName.</p> <p>* If the fieldName starts with - then a must not filter is used</p>	Operator	Description		value type			< string > array(multi)	
Operator	Description									
value type										

Type	Name	Description	Schema	Default
Query	gintersect <i>optional</i>	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	gwithin <i>optional</i>	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	human <i>optional</i>	Human readable print	boolean	"false"
Query	max-age-cache <i>optional</i>	max-age-cache	integer(int32)	
Query	notgintersect <i>optional</i>	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	notgwithin <i>optional</i>	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	notpwithin <i>optional</i>	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	pretty <i>optional</i>	Pretty print	boolean	"false"
Query	pwithin <i>optional</i>	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	q <i>optional</i>	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 <i>required</i>	collection	string
Body	body <i>optional</i>		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 <i>required</i>	collection	string	
Query	after <i>optional</i>	Any element having its point in time reference after the given timestamp	integer	
Query	before <i>optional</i>	Any element having its point in time reference before the given timestamp	integer	
Query	exclude <i>optional</i>	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	<p>* A triplet for filtering the result. Multiple filter can be provided. The order does not matter.</p> <p>* A triplet is composed of a field name, a comparison operator and a value.</p> <p>The possible values of the comparison operator are :</p> <p>----</p> <table><tr><th>Operator </th><th>Description</th><th></th></tr><tr><td>value type</td><td></td><td></td></tr></table> <p>:eq: {fieldName} equals {comma separated values}. OR operation is applied for the specified values numeric or strings</p> <p>:ne: {fieldName} must not equal {comma separated values }. AND operation is applied for the specified values numeric or strings</p> <p>:like: {fieldName} is like {value} numeric or strings</p> <p>:gte: {fieldName} is greater than or equal to {value} numeric</p> <p>:gt: {fieldName} is greater than {value} numeric</p> <p>:lte: {fieldName} is less than or equal to {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>----</p> <p>* The AND operator is applied between filters having different fieldNames.</p> <p>* The OR operator is applied on filters having the same fieldName.</p> <p>* If the fieldName starts with - then a must not filter is used</p>	Operator	Description		value type			< string > array(multi)	
Operator	Description									
value type										

Type	Name	Description	Schema	Default
Query	from <i>optional</i>	From index to start the search from. Defaults to 0.	integer	"0"
Query	gintersect <i>optional</i>	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	gwithin <i>optional</i>	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	human <i>optional</i>	Human readable print	boolean	"false"
Query	include <i>optional</i>	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 <i>optional</i>	max-age-cache	integer(int32)	
Query	notgintersect <i>optional</i>	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	notgwithin <i>optional</i>	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	notpwithin <i>optional</i>	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	pretty <i>optional</i>	Pretty print	boolean	"false"
Query	pwithin <i>optional</i>	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	q <i>optional</i>	A full text search. Optionally, it's possible to search on a field using this syntax: {fieldname}:{text}	string	
Query	size <i>optional</i>	The maximum number of entries or sub-entries to be returned. The default value is 10	integer	"10"
Query	sort <i>optional</i>	<p>* Sort the result on the given fields ascending or descending.</p> <p>* Fields can be provided several times by separating them with a comma. The order matters.</p> <p>* For a descending sort, precede the field with '-'. The sort will be ascending otherwise.</p>	< 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 <i>required</i>	collection	string
Body	body <i>optional</i>		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 <i>required</i>	collection	string	
Query	after <i>optional</i>	Any element having its point in time reference after the given timestamp	integer	
Query	before <i>optional</i>	Any element having its point in time reference before the given timestamp	integer	
Query	exclude <i>optional</i>	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	<p>* A triplet for filtering the result. Multiple filter can be provided. The order does not matter.</p> <p>* A triplet is composed of a field name, a comparison operator and a value.</p> <p>The possible values of the comparison operator are :</p> <p>----</p> <table><tr><th>Operator </th><th>Description</th><th></th></tr><tr><td>value type</td><td></td><td></td></tr></table> <p>:eq: {fieldName} equals {comma separated values}. OR operation is applied for the specified values numeric or strings</p> <p>:ne: {fieldName} must not equal {comma separated values }. AND operation is applied for the specified values numeric or strings</p> <p>:like: {fieldName} is like {value} numeric or strings</p> <p>:gte: {fieldName} is greater than or equal to {value} numeric</p> <p>:gt: {fieldName} is greater than {value} numeric</p> <p>:lte: {fieldName} is less than or equal to {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>:lt: {fieldName} is less than {value} numeric</p> <p>----</p> <p>* The AND operator is applied between filters having different fieldNames.</p> <p>* The OR operator is applied on filters having the same fieldName.</p> <p>* If the fieldName starts with - then a must not filter is used</p>	Operator	Description		value type			< string > array(multi)	
Operator	Description									
value type										

Type	Name	Description	Schema	Default
Query	from <i>optional</i>	From index to start the search from. Defaults to 0.	integer	"0"
Query	gintersect <i>optional</i>	Any element having its geometry intersecting the given geometry (WKT)	< string > array(multi)	
Query	gwithin <i>optional</i>	Any element having its geometry contained within the given geometry (WKT)	< string > array(multi)	
Query	human <i>optional</i>	Human readable print	boolean	"false"
Query	include <i>optional</i>	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 <i>optional</i>	max-age-cache	integer(int32)	
Query	notgintersect <i>optional</i>	Any element having its geometry not intersecting the given geometry (WKT)	< string > array(multi)	
Query	notgwithin <i>optional</i>	Any element having its geometry outside the given geometry (WKT)	< string > array(multi)	
Query	notpwithin <i>optional</i>	Any element having its centroid outside the given geometry (WKT)	< string > array(multi)	
Query	pretty <i>optional</i>	Pretty print	boolean	"false"
Query	pwithin <i>optional</i>	Any element having its centroid contained within the given geometry (WKT)	< string > array(multi)	
Query	q <i>optional</i>	A full text search. Optionally, it's possible to search on a field using this syntax: {fieldname}:{text}	string	
Query	size <i>optional</i>	The maximum number of entries or sub-entries to be returned. The default value is 10	integer	"10"
Query	sort <i>optional</i>	<p>* Sort the result on the given fields ascending or descending.</p> <p>* Fields can be provided several times by separating them with a comma. The order matters.</p> <p>* For a descending sort, precede the field with '-'. The sort will be ascending otherwise.</p>	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`

2.2.14. Get an Arlas document

```
GET /explore/{collection}/{identifier}
```

Description

Returns a raw indexed document.

Parameters

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

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasHit
400	Bad request.	ArlasError
404	Not Found Error.	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 <i>optional</i>	string
collectField <i>optional</i>	string
field <i>optional</i>	string
format <i>optional</i>	string
interval <i>optional</i>	string
order <i>optional</i>	string
size <i>optional</i>	string
true <i>optional</i>	string
type <i>optional</i>	string

3.2. AggregationRequest

Name	Schema
aggregations <i>optional</i>	Aggregations
filter <i>optional</i>	Filter
form <i>optional</i>	Form

3.3. Aggregations

Name	Schema
aggregations <i>optional</i>	< AggregationModel > array

3.4. ArlasAggregation

Aggregation result

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

3.5. ArlasError

Name	Schema
error <i>optional</i>	string
message <i>optional</i>	string
status <i>optional</i>	integer(int32)

3.6. ArlasHit

A hit retrieved from an ARLAS Collection

Name	Description	Schema
data <i>optional</i>	The hit's data	object
md <i>optional</i>	The hit's metadata	ArlasMD

3.7. ArlasHits

A collection of hits retrieved from ARLAS Collections

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

3.8. ArlasMD

Metadata of the ARLAS hit

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

3.9. ArlasMetric

Metric agg

Name	Description	Schema
field <i>optional</i>	field of the metric aggregation	string
type <i>optional</i>	Name of the metric aggregation	string
value <i>optional</i>	Value of the metric aggregation	number(double)

3.10. ArlasSuccess

Name	Schema
message <i>optional</i>	string
status <i>optional</i>	integer(int32)

3.11. CollectionReference

The reference to ARLAS collection that embed elasticsearch index description.

Name	Description	Schema
collection_name <i>optional</i>	The collection name	string
params <i>optional</i>	The collection parameters	CollectionReferenceParameters

3.12. CollectionReferenceDescription

Describe the structure and the content of the given collection.

Name	Description	Schema
collection_name <i>optional</i>	The collection name	string
params <i>optional</i>	The collection parameters	CollectionReferenceParameters

Name	Description	Schema
properties <i>optional</i>	The collection fields	< CollectionReferenceDescriptionProperty > array

3.13. CollectionReferenceDescriptionProperty

Name	Description	Schema
format <i>optional</i>	The collection field format	string
name <i>optional</i>	The collection field name	string
type <i>optional</i>	The collection field type	enum (TEXT, KEYWORD, LONG, INTEGER, SHORT, BYTE, DOUBLE, FLOAT, DATE, BOOLEAN, BINARY, INT_RANGE, FLOAT_RANGE, LONG_RANGE, DOUBLE_RANGE, DATE_RANGE, OBJECT, NESTED, GEO_POINT, GEO_SHAPE, IP, COMPLETION, TOKEN_COUNT, MAPPER_MURMUR3, UNKNOWN)

3.14. CollectionReferenceParameters

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

Name	Description	Schema
centroid_path <i>optional</i>	Path to the collection's centroid Example : "centroid"	string
custom_params <i>optional</i>	Calculated params.	< string, string > map

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

3.15. Count

Name	Schema
filter <i>optional</i>	Filter
form <i>optional</i>	Form

3.16. Crs

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

3.17. Feature

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

3.18. FeatureCollection

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

3.19. Filter

Name	Schema
after <i>optional</i>	integer(int64)
before <i>optional</i>	integer(int64)
f <i>optional</i>	< string > array
gintersect <i>optional</i>	string
gwithin <i>optional</i>	string
notgintersect <i>optional</i>	string

Name	Schema
notgwithin <i>optional</i>	string
notpwithin <i>optional</i>	string
pwithin <i>optional</i>	string
q <i>optional</i>	string

3.20. Form

Name	Schema
human <i>optional</i>	boolean
pretty <i>optional</i>	boolean

3.21. GeoJsonObject

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

3.22. GeometryCollection

Polymorphism : Inheritance

Discriminator : type

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

3.23. LineString

Polymorphism : Inheritance

Discriminator : type

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

3.24. LngLatAlt

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

3.25. MultiLineString

Polymorphism : Inheritance

Discriminator : type

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

3.26. MultiPoint

Polymorphism : Inheritance

Discriminator : type

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

3.27. MultiPolygon

Polymorphism : Inheritance

Discriminator : type

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

3.28. Point

Polymorphism : Inheritance

Discriminator : type

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

3.29. Polygon

Polymorphism : Inheritance

Discriminator : type

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

3.30. Search

Name	Schema
filter <i>optional</i>	Filter
form <i>optional</i>	Form
size <i>optional</i>	Size
sort <i>optional</i>	Sort

3.31. Size

Name	Schema
from <i>optional</i>	integer(int32)
size <i>optional</i>	integer(int32)

3.32. Sort

Name	Schema
sort <i>optional</i>	string