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
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	No Content

Consumes

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

Produces

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

2.2.2. Aggregate

```
GET /explore/{collections}/_aggregate
```

Description

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

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)	

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} Where the {type}:{field} part is mandatory AND interval, format, collect_field, collect_fct, order AND on are optional sub-parameters. * {type} possible values are : datehistogram, histogram, 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} = term, then interval-{interval} is not needed. * format-{format} is to be specified when {type} = datehistogram. It's the date format for key aggregation. * {collect_fct} is the aggregation function to apply to collections on the specified {collect_field}. {collect_fct} possible values are : ---- avg,cardinality,max,min,sum ---- * {order} is set to sort the aggregation result on the field name or on the result itself. It's values are 'asc' or 'desc'. * {on} is set to specify whether the {order} is on the field name or the result. It's values are	< string > array(multi)	

Type	Name	Description	Schema	Default
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	**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"`
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. * For aggregation, provide the `agg` keyword as the `{field}`.	< string > array(multi)	

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

```
GET /explore/{collections}/_geoaggregate
```

Description

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

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)	

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} Where the {type}:{field} part is mandatory AND interval, format, collect_field, collect_fct, order AND on are optional sub-parameters. * {type} possible value is : geohash. * {interval} must be a numeric value. * format-{format} is to be specified when {type} = datehistogram. It's the date format for key aggregation. * {collect_fct} is the aggregation function to apply to collections on the specified {collect_field}. {collect_fct} possible values are : ---- avg,cardinality,max,min,sum ---- * {order} is set to sort the aggregation result on the field name or on the result itself. It's values are 'asc' or 'desc'. * {on} is set to specify whether the {order} is on the field name or the result. It's values are 'field' or 'result'. agg parameter in this case is not multiple. For more details, check https://gitlab.com/GISAIA.ARLAS/ARLAS-server/blob/master/doc/api/API-definition.md	< string > array(multi)	
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	**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"`
Query	**sort** *optional*	Sort the result on a given field, ascending or descending : '{fieldName}:(ASC, DESC)' . The parameter can be provided several times. The order matters. For aggregation, provide the 'agg' keyword as the {fieldName}.	< string > array(multi)	

Responses

HTTP Code	Description	Schema
200	Successful operation	No Content

Consumes

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

Produces

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

2.2.4. 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`

2.2.5. 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	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	**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 BBOX (top,left,bottom,right)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given BBOX (top,left,bottom,right)	< string > array(multi)	
Query	**q** *optional*	A full text search	string	

Responses

HTTP Code	Description	Schema
200	Successful operation	No Content

Consumes

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

Produces

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

2.2.6. 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	No Content

Consumes

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

Produces

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

2.2.7. Geoearch

```
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	integer(int64)	
Query	**before** *optional*	Any element having its point in time reference before the given timestamp	integer(int64)	
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 : | {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	**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	**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 BBOX (top,left,bottom,right)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given BBOX (top,left,bottom,right)	< 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"`

Type	Name	Description	Schema	Default
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. * For aggregation, provide the `agg` keyword as the `{field}`.	< string > array(multi)	

Responses

HTTP Code	Description	Schema
200	Successful operation	No Content

Consumes

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

Produces

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

2.2.8. 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	integer(int64)	

Type	Name	Description	Schema	Default
Query	**before** *optional*	Any element having its point in time reference before the given timestamp	integer(int64)	
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 : | {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	**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	**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 BBOX (top,left,bottom,right)	< string > array(multi)	
Query	**pretty** *optional*	Pretty print	boolean	`"false"`
Query	**pwithin** *optional*	Any element having its centroid contained within the given BBOX (top,left,bottom,right)	< 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"`

Type	Name	Description	Schema	Default
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. * For aggregation, provide the `agg` keyword as the `{field}`.	string	

Responses

HTTP Code	Description	Schema
200	Successful operation	ArlasHits

Consumes

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

Produces

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

Chapter 3. Definitions

3.1. ArlasAggregation

Type : object

3.2. ArlasError

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

3.3. 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

3.4. 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)

3.5. ArlasMD

Metadata of the ARLAS hit

Name	Description	Schema
centroid *optional*	The centroid of the hit	Point
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)

3.6. ArlasSuccess

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

3.7. 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

3.8. 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
geometry_path *optional*	Path to the collection's geometry **Example** : `"geometry"`	string
id_path *optional*	Path to the collection's id **Example** : `"id"`	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

3.9. Crs

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

3.10. Feature

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

3.11. FeatureCollection

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

3.12. GeoJsonObject

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

3.13. GeometryCollection

Polymorphism : Inheritance
Discriminator : type

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

3.14. LineString

Polymorphism : Inheritance
Discriminator : type

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

3.15. LngLatAlt

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

3.16. MultiLineString

Polymorphism : Inheritance
Discriminator : type

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

3.17. MultiPoint

Polymorphism : Inheritance
Discriminator : type

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

3.18. MultiPolygon

Polymorphism : Inheritance
Discriminator : type

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

3.19. Point

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

3.20. Polygon

Polymorphism : Inheritance
Discriminator : type

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