

Contents

Syntool Portal Config	2
Table of Contents	2
Introduction	3
Timeline	3
Dataset (aka Granule)	3
Product (aka Collection)	3
Syntool Web	3
Products selection panel	3
Settings popup	3
Timeline	4
Pagination	5
Messages	8
Hotspots	8
Shapes (aka User shapes)	9
Configuration file	9
Example	9
Configuration Fields	12
Additional Layers	16
Types	16
URLShortenerCreator	16
URLShortener	17
ResponseType	18
ShortenerCallback	18
ShortenerErrorback	18
Message	18
MessageType	19
Hotspot	19
DrawableShape	20
Shape	21
ShapeType	25
ShapeArrow	25
ShapeStyleMap	26
ShapeStyle	27
Layer	28
TimeRange	28
Timespan	28
MarkerSymbols	29
MarkerSymbol	29
Product	30
ProductType	35
Group	36
Color	37
Fields	37
Palette	38
Marker	38
MarkerStyle	39
MarkerStateStyle	39
ImageMarkerStyle	39
PolygonMarkerStyle	40

Syntool Portal Config

Table of Contents

- Introduction
 - Timeline
 - Dataset (aka Granule)
 - Product (aka Collection)
- Syntool Web
 - Products selection panel
 - Settings popup
 - Timeline
 - Pagination
 - Messages
 - Hotspots
 - Shapes
- Configuration file
- Configuration Fields
 - Additional Layers
- Types
 - URLShortenerCreator
 - URLShortener
 - ResponseType
 - ShortenerCallback
 - ShortenerErrorback
 - Message
 - MessageType
 - Hotspot
 - DrawableShape
 - Shape
 - ShapeType
 - ShapeArrow
 - ShapeStyleMap
 - ShapeStyle
 - Layer
 - TimeRange
 - Timespan
 - MarkerSymbols
 - MarkerSymbol
 - Product
 - ProductType
 - Group
 - Color
 - Fields
 - Palette
 - Marker
 - MarkerStyle
 - ImageMarkerStyle
 - PolygonMarkerStyle

The following document is split into two sections. The first one is dedicated to the different functions available on Syntool web portal and the second one describes the configuration file.

Introduction

Timeline

The timeline consists of two concepts:

- the date and time (aka datetime),
- the date range of a certain width centered around the datetime (aka timespan).

The currently selected datetime is referred to as the “current datetime”. And the currently selected timespan is referred to as the “current timespan”.

The user can change both the current datetime and the current timespan.

An animation is just an automatic increment/decrement of the current datetime by a fixed step (controlled by the animation speed) every fixed delay.

Dataset (aka Granule)

A temporally and spatially referenced piece of data.

It can be a trajectory (e.g. drifters), an image (e.g. SAR), or a vector matrix (barbs, arrows, or streamlines) (e.g. surface current).

It must have:

- spatial attributes: a bounding box and an outlining polygon (aka outline).
- temporal attributes: a start date and an end date (a date range).

The dataset will be added to the map either:

- when it’s date range contains the current datetime, or
- when it’s date range intersects the current timespan.

Product (aka Collection)

A collection of datasets that share the same configuration.

Syntool Web

In this section, the different components and functionalities of Syntool web portals are described.

Products selection panel

Products are listed in the left panel. This panel allows the user to select and deselect them.

Note: A product that is not selected is ignored completely.

Settings popup

Selected products are listed in the settings popup (accessible by clicking on the “Settings” button in the top bar).

This popup allows the user to reorder the products (a low product on the list is displayed below a higher one), change their transparency, and filter their datasets using a pattern (matched against the dataset’s ID).

This popup allows the user to see the colorbar of each product (when applicable).

Timeline

The timeline is located in the bottom of the page. It's initially collapsed (the detailed view is hidden).

The timeline is composed of three parts:

1. the top bar
2. the availability calendar
3. the detailed timeline

Top bar

Item	Description
Animation controls	Allow to start and stop the animation, and control it's speed.
Timespan selection	Allow to change the current timespan by picking one of the pre-defined ones.
Find nearest button	Allow to find and go to the nearest datetime that contains at least one dataset.
Collocation toggle	Toggles the collocation feature which changes the behavior of the availability calendar. For more info see the availability calendar
Number of datasets in the map	Is the number of datasets in the map, visible or not (see pagination).
Pagination controls	Allow to change the current page, and shows the number of datasets currently visible. For more info see pagination.
ID of selected dataset	Shows the ID of the selected dataset and the ID of it's product. (Only visible when one or more datasets are selected)
The current date and time	Shows the current current datetime. (Only visible when no dataset is selected)
Position of the mouse	Shows the coordinates of the mouse pointer in degrees longitude and latitude.

Availability calendar

Composed of three rows:

1. Contains all the possible years.
2. Contains all the months of the selected year.
3. Contains all the days of the selected month of the selected year.

Each cell (year/month/day) of the calendar can be:

- Grey: none of the selected products have any datasets in that year/month/day.
- White: at least one of the selected products have at least one dataset in that year/month/day.
- Yellow: that year/month contains at datasets from all of the selected products. But not on the same day.
- Red: at least one day of that year/month contains datasets from all of the selected products.

Note: The yellow and red states are only used if the Collocation is active (The collocation toggle in the top bar of the timeline is red).

Detailed timeline

Hidden by default, can be shown by clicking on the arrow pointing upward just above the timeline.

The full width represents the current timespan. And the vertical line in the center represents the current datetime.

The dots and horizontal lines represent the datasets. The horizontal lines represent the datetime range of the dataset. The dots represent datasets with small datetime range. A datetime range is considered small when it's length is smaller than 10 minutes.

The dots and horizontal lines can be:

- Grey: if their datasets are not in the map.
- Blue: if their datasets are in the map.
- Yellow: if their datasets are selected.

The dots and horizontal lines are clickable (if their datasets are in the map). When clicked it's respective dataset is selected in the map.

They also have a tooltip which contains the ID of their respective dataset.

Pagination

Syntool is rendered unresponsive or unusable when the number of datasets in the map gets to big. To solve this the datasets are split into pages, and shown one page of datasets at a time.

Smart pagination

The `maxDatasetsWeight` is the rendering budget.

The `datasetWeight` of a product is the average cost of rendering one of it's datasets.

The `minPageSize` of a product is the minimum number of it's datasets to be rendered (even if we go over-budget).

The pagination system tries to maximize the number of rendered datasets. While distributing as uniformly as possible the budget on all selected products and respecting the `minPageSize` of all products. If a product is over-budgeted then the remaining budget is redistributed.

The `priority` of a product comes in play in the case that the budget can't be distributed uniformly (the number of rendered datasets must be an integer). In which case the products with the highest priorities (biggest `priority`) will have their datasets rendered.

Examples

1. Normal

Product	priority	datasetWeight	minPageSize	Number of datasets
Product A	0	1	0	1000
Product B	0	1	0	1000
Product C	0	1	0	1000

1st iteration

- `maxDatasetsWeight` is 10
- `BudgetPerProduct` = `maxDatasetsWeight` / 3 = 10 / 3 = 3

We distribute the budget uniformly.

Product	Page size	Number remaining of datasets
Product A	3	997
Product B	3	997
Product C	3	997

2nd iteration

- `maxDatasetsWeight` is $10 - 3 - 3 - 3 = 1$
- `BudgetPerProduct` = `maxDatasetsWeight` / 3 = $1 / 3 = 0$

Since `maxDatasetsWeight` is 1 greater than 0 and `BudgetPerProduct` is 0, then this means that we can't distribute the budget uniformly. So starting from the product with the highest priority down to the one with lowest, giving 1 from the budget to each one until we run-out.

Product	Page size	Number remaining of datasets
Product A	4	996
Product B	3	997
Product C	3	997

2. Priority

Product	priority	datasetWeight	minPageSize	Number of datasets
Product A	0	1	0	1000
Product B	0	1	0	1000
Product C	1	1	0	1000

1st iteration

- `maxDatasetsWeight` is 10
- `BudgetPerProduct` = `maxDatasetsWeight` / 3 = $10 / 3 = 3$

We distribute the budget uniformly.

Product	Page size	Number remaining of datasets
Product A	3	997
Product B	3	997
Product C	3	997

2nd iteration

- `maxDatasetsWeight` is $10 - 3 - 3 - 3 = 1$
- `BudgetPerProduct` = `maxDatasetsWeight` / 3 = $1 / 3 = 0$

Since `maxDatasetsWeight` is 1 greater than 0 and `BudgetPerProduct` is 0, then this means that we can't distribute the budget uniformly. So starting from the product with the highest priority down to the one with lowest, giving 1 from the budget to each one until we run-out.

Product	Page size	Number remaining of datasets
Product A	3	997

Product	Page size	Number remaining of datasets
Product B	3	997
Product C	4	996

3. Weight

Product	priority	datasetWeight	minPageSize	Number of datasets
Product A	0	2	0	1000
Product B	0	1	0	1000
Product C	0	1	0	1000

1st iteration

- `maxDatasetsWeight` is 10
- `BudgetPerProduct` = `maxDatasetsWeight` / 3 = 10 / 3 = 3

We distribute the budget uniformly.

Product	Page size	Number remaining of datasets
Product A	3 / 2 = 1	999
Product B	3	997
Product C	3	997

2nd iteration

- `maxDatasetsWeight` is 10 - (1 * 2) - 3 - 3 = 2
- `BudgetPerProduct` = `maxDatasetsWeight` / 3 = 2 / 3 = 0

Since `maxDatasetsWeight` is 2 greater than 0 and `BudgetPerProduct` is 0, then this means that we can't distribute the budget uniformly. So starting from the product with the highest priority down to the one with lowest, giving 1 from the budget to each one until we run-out.

Product	Page size	Number remaining of datasets
Product A	3 + 1 / 2 = 3	999
Product B	3 + 1 = 4	996
Product C	3 + 1 = 4	996

4. minPageSize

Product	priority	datasetWeight	minPageSize	Number of datasets
Product A	0	1	0	1000
Product B	0	1	4	1000
Product C	0	1	0	1000

before iterating (applying `minPageSize` constraint)

- `maxDatasetsWeight` is 10

Product	Page size	Number remaining of datasets
Product A	0	1000
Product B	4	996
Product C	0	1000

1st iteration

- `maxDatasetsWeight` is $10 - 4 = 6$
- `BudgetPerProduct` = $\text{maxDatasetsWeight} / 3 = 6 / 3 = 2$

We distribute the budget uniformly.

Product	Page size	Number remaining of datasets
Product A	2	998
Product B	$4 + 2 = 6$	994
Product C	2	996

5. `minPageSize` with going over-budget

Product	priority	datasetWeight	minPageSize	Number of datasets
Product A	0	2	4	1000
Product B	0	1	4	1000
Product C	0	1	0	1000

before iterating (applying `minPageSize` constraint)

- `maxDatasetsWeight` is 10

Product	Page size	Number remaining of datasets
Product A	4	996
Product B	4	996
Product C	0	1000

1st iteration

- `maxDatasetsWeight` is $10 - (4 * 2) - 4 = -2$

Since `maxDatasetsWeight` is negative (-2) then we have blown our budget. So we stop.

Messages

A message is like a notification, it conveys some information that the user should see.

Hotspots

A hotspot is a bookmark of the current configuration (datetime, timespan, map viewport and the selected products with their respective transparency levels and order), with and associated name or title and an optional description link (a link to an external web page that describes it, like an article).

Hotspots can be created, viewed, shared, and deleted from the “Hotspots” popup (accessible by clicking on the “Hotspots” button in the top bar).

Shapes (aka User shapes)

Shapes are special datasets. They may have an author and a label (which can hold any text value). They are filtered based on their label not their ID.

Shapes can be created by the user. The user created ones are always in the product of type “USER_SHAPES”. A shape is a geometry (e.g. point, line, polygon, ...) or a text label positioned geographically. A shape can be imported and exported. The exported shapes are in the same format accepted in the config (see Shape).

Configuration file

A portal must be configured using is a JavaScript file having the form of a commonjs module that exports an object with the fields described in the configuration table below.

The file will have the form:

```
// any code here...
module.exports = {
  // fields here...
};
```

The different fields of the configuration file are going to be explained in the following section. The example given below may be a good starting point for the creation of a new configuration file.

Example

```
// Use global extent for web mercator
var maxExtent = [
  -20037508.342789, // left (smallest longitude)
  -20037508.342789, // bottom (smallest latitude)
  20037508.342789, // right (biggest longitude)
  20037508.342789, // top (biggest latitude)
];

function argoMarker(color) {
  return {
    graphicName: 'left_arrow_tip',
    pointRadius: 4,
    strokeWidth: 1,
    strokeColor: color,
    fillColor: color,
  };
}

module.exports = {
  version: '1.1.0',

  /* Services */
  serviceHost: 'http://syntooldata.oceandatalab.com/server',
  dataServers: [
    'http://syntooldata.oceandatalab.com/'
  ]
};
```

```

    'http://syntooldata1.oceandatalab.com/',
    'http://syntooldata2.oceandatalab.com/',
    'http://syntooldata3.oceandatalab.com/',
    'http://syntooldata4.oceandatalab.com/',
    'http://syntooldata5.oceandatalab.com/',
    'http://syntooldata6.oceandatalab.com/',
    'http://syntooldata7.oceandatalab.com/',
    'http://syntooldata8.oceandatalab.com/',
    'http://syntooldata9.oceandatalab.com/',
  ],

  /* Map */
  projection: 'EPSG:3857', // web mercator projection
  maxExtent: maxExtent,
  baseLayers: [
    new OpenLayers.Layer.Google('Google Satellite', {
      type: google.maps.MapTypeId.SATELLITE,
      numZoomLevels: 20,
      maxExtent: new OpenLayers.Bounds(maxExtent),
    }),
    new OpenLayers.Layer.Google('Google Hybrid', {
      type: google.maps.MapTypeId.HYBRID,
      numZoomLevels: 20,
      maxExtent: new OpenLayers.Bounds(maxExtent),
    }),
    new OpenLayers.Layer.Google('Google Physical', {
      type: google.maps.MapTypeId.TERRAIN,
      numZoomLevels: 22,
      maxExtent: new OpenLayers.Bounds(maxExtent),
    }),
    new OpenLayers.Layer.Google('Google Streets', {
      type: google.maps.MapTypeId.ROADMAP,
      numZoomLevels: 22,
      maxExtent: new OpenLayers.Bounds(maxExtent),
    }),
  ],

  /* Timeline */
  yearsRange: [2010, 2018],
  timeRanges: [
    ['6-Hour', '6h'], // 6 hours timespan with 1 day overview
    ['Daily', '1d', true], // 1 day timespan with 1 week overview and selected by default
    ['3-Day', '3d'], // 3 day timespan with 1 week overview
    ['Weekly', '1w'], // 1 week timespan with 1 month overview
    ['Bi-weekly', '2w'], // 2 weeks timespan with 1 month overview
  ],

  /* Misc */
  storageId: 'example',
  markerSymbols: {
    left_arrow_tip: [-8,4, -8,-4, 0,0, -8,4],
  },
  showGroups: false,
  collapsibleGroups: false,

```

```

/* Predefined Data */
hotspots: [
  '?name=Chlorophyll%20A%20and%20Globcurrent%20%40%20Agulhas%20region&date=1412426592000&timespan=1d%',
],
drawableShapes: [
  /* ---|> */
  {
    title: 'Right Arrow',
    cssClassNames: 'olControlDrawFeatureRightArrow',
    wkt: 'MULTILINESTRING((0 0, 100 0),(90 10, 100 0, 90 -10, 90 10))',
  },
],
products: [
  {
    /* General */
    label: 'Shapes', // the label shown in the left products pan
    id: '900913_Shapes', // the id of this product in the index
    type: 'SERVER_SHAPES', // indicate that this product defines shape

    /* Timeline */
    mustBeCurrent: false, // the datasets of this product don't need
                          // intersect the current date/time, only in
                          // the current timespan

    /* Pagination */
    datasetWeight: 0.1, // the datasets of this products take a 10t
                       // of time to render compared to normal dat

    /* General Rendering */
    stackLevel: 150,

    /* Shapes */
    shapes: [
      {
        type: 'WKT',
        text: 'text',
        start: 1149163200000,
        end: 1149163200000,
        author: 'anonymous',
        wkt: 'POLYGON((77.51953122494571 59.26588064106204,40.95703122494571 44.339565266015946,64.68'
      },
    ],
  },
],
{
  /* General */
  label: 'Drifters 1000m North Atlantic speed (ANDRO)', // the label shown in the left products pan
  id: '3857_ARGO_Deep_NATL1000', // the id of this product in the index
  type: 'TRAJECTORIES', // render the datasets of this product as t

  /* Timeline */
  mustBeCurrent: false, // the datasets of this product don't need
                        // intersect the current date/time, only in
                        // the current timespan
}

```

```

    /* Pagination */
    datasetWeight: 0.1, // the datasets of this products take a 10t
                        // of time to render compared to normal dat

    /* General Rendering */
    mapMinZoom: 2, // this product is only rendered starting f
                 // zoom level 2

    stackLevel: 90,

    /* Dynamic Rendering */
    fields: {
      // the "speed" field (which representes the channel "speed" of the dataset)
      speed: {
        min: 0, // The minimum value.
        max: 0.35, // The maximum value.
      },
    },
    palette: {
      field: 'speed', // render the "speed" field of the datasets
      colormap: 'images/palettes/undefined', // using the default "yellow" to "red" colo
    },

    /* Trajectory Rendering */
    marker: argoMarker, // use the `argoMarker` function defined ab
                       // to render the trajectory marker
    markerPostion: 1, // the trajectory marker should always be a
                     // the end of the trajectory
  },
],
};

```

Configuration Fields

Field	Type	Default	Description
version	string	Required	MUST be '1.1.x' where x can be any positive integer.
<i>Services</i> seashotHost	URL	Optional	The URL to the server running a seashot services server (or a compatible one). If not given, the seashot feature will be deactivated.
serviceHost	URL	Required	The URL to the server running a Syntool services server (or a compatible one).

Field	Type	Default	Description
<code>dataServers</code>	<code>URL []</code>	Required	A list of URLs to use for requesting data (tiles, geoJSON, ...). All URLs MUST be equivalent, a list is given in order to circumvent the browser's limitation on the number of simultaneous requests to one domain.
<code>shortenerForURL</code>	<code>URLShortenerCreator</code>	Optional	If provided, will be called to know how to shorten a given URL. Used, for example, when user clicks on the "Share" button.
<i>Map</i> <code>projection</code>	<code>string</code>	Required	A string identifying the Well Known Identifier for the projection. Specifies the projection of the map.
<code>maxExtent</code>	<code>number [4]</code>	Required	The maximum extent of the map defined as [<code>left</code> , <code>bottom</code> , <code>right</code> , <code>top</code>] all in the projection <code>projection</code> .
<code>zoomOffset</code>	<code>integer >= 0</code>	0	Some map providers (like Bing maps) skip some zoom levels (e.g. Bing maps considers the zoom level 0 as level 1), if its the case you can use this field to set the offset (e.g. set it to 1).
<code>defaultZoom</code>	<code>integer >= 0</code>	Optional	If provided, will be used to set the zoom level of the map by default (if the <code>extent</code> is not set in the URL of the page).

Field	Type	Default	Description
<code>baseLayers</code>	<code>Layer []</code>	Required	A list of layers. It MUST contain at least one layer. One of these layers (by default the first one) will be used as the background layer of the map. These layers will be displayed in a list which and be toggled using the globe button in the top bar of Syntool allowing the user to switch between them.
<code>additionalLayers</code>	<code>Layer []</code>	<code>[]</code>	A list of layers that are always visible and placed above or below all products. For more info see Additional Layers.
<i>Timeline</i> <code>currentDate</code>	<code>Date</code>	Optional	The default current date of the portal. If not given it will be set to the client's local date and time and then a find nearest will be executed. (example for 01/06/2006 12:00 use <code>new Date(Date.UTC(2006, 5, 1, 12))</code>), for more info see MDN's <code>Date.UTC</code>)
<code>animationSpeed</code>	<code>number > 0</code>	20 hours / s	The base speed of the animations in <i>milliseconds of virtual time per second of real time</i> .
<code>timeRanges</code>	<code>TimeRange []</code>	<i>see below</i>	A list of predefined timespans that the users can select.
<code>yearsRange</code>	<code>integer[2] > 1969</code>	<i>see below</i>	Timeline years range from January 1st to January 1st (which means the upper bound will not appear in the timeline)
<i>Misc</i>			

Field	Type	Default	Description
<code>seashotTag</code>	<code>string</code>	Optional	If provided, will be used to pre-fill the seashot creation form.
<code>storageId</code>	<code>string</code>	Required	Portals that have the same <code>storageId</code> and the same domain will share the same local storage (local hotspots and user shapes).
<code>maxDatasetsWeight</code>	<code>number >= 0</code>	100	The maximum cumulated weight of datasets rendered at the same time. For more info see Pagination .
<code>markerSymbols</code>	<code>MarkerSymbols</code>	<code>{}</code>	A hash map of polygons to be referenced in the product's <code>graphicName</code> .
<code>showGroups</code>	<code>boolean</code>	<code>false</code>	If set to <code>true</code> , the products will be ordered by groups and the group label will be visible in the products panel.
<code>collapsibleGroups</code>	<code>boolean</code>	<code>false</code>	If set to <code>true</code> , clicking on a group label will show/hide its contents in the products panel. Only applicable if <code>showGroups</code> is <code>true</code> .
<i>Predefined Data</i> <code>messages</code>	<code>Message []</code>	<code>[]</code>	A list of predefined ordered messages. For more info see Messages .
<code>hotspots</code>	<code>Hotspot []</code>	<code>[]</code>	A list of predefined ordered hotspots. For more info see Hotspots .
<code>drawableShapes</code>	<code>DrawableShape []</code>	<code>[]</code>	A list of predefined drawable shapes that the users can use to draw user shapes. (Some shapes are defined in Syntool and can't be removed like: polygon, polyline, and point).

Field	Type	Default	Description
<code>products</code>	<code>Product []</code>	<code>[]</code>	An ordered list of available products. For more info see Products.
<code>groups</code>	<code>Group []</code>	<code>[]</code>	An ordered list of groups. For more info see Groups.

Note: If no product of type 'USER_SHAPES' is found in the `products` field, one will be added automatically at the top of the list.

Note: The `products` list must at most contain one 'USER_SHAPES'.

Note: The product with the type 'USER_SHAPES' will be used for the user's local shapes.

Note: All product IDs MUST be unique (no two products can have the same `id`).

Note: All message IDs MUST be unique (no two messages can have the same `id`).

`timeRanges` default value

```
[
  ['6-Hour', '6h'],
  ['Daily', '1d', true],
  ['3-Day', '3d'],
  ['Weekly', '1w'],
  ['Bi-weekly', '2w'],
]
```

`yearsRange` default value

The default value for the `yearsRange` property is dynamically generated so that the timeline displays the last 15 years (+ current year).

Additional Layers

Additional layers are normal OpenLayers layers with an optional additional custom option `syntoolIsOverlay`. `syntoolIsOverlay` defaults to `false` and when set to `true` the layer is displayed above all products. Otherwise it's displayed under all products.

The order of the layer specifies the order in which they get rendered: * The first element of the list is closest to the base layer and displayed below all other layers of the same group (with the same value `syntoolIsOverlay`). * The last element of the list is farthest from the base layer and displayed above all other layers of the same group (with the same value `syntoolIsOverlay`).

Types

URLShortenerCreator

Type: (`url: URL`) -> URLShortener

URLShortener

Type: object

Field	Type	Default	Description
url	URL	Required	The URL to send the HTTP request to.
method	string	'GET'	The request method (example 'GET', 'POST').
data	object	Optional	The data to be sent with the request. If provided, data will be encoded and added to the URL for 'HEAD' and 'GET' requests, or sent as the request's form data body for all other request methods.
responseType	ResponseType	'text'	How the response should be interpreted. For more info see MDN's XMLHttpRequest.responseType.
callback	ShortenerCallback	Optional	If provided, will be called with the received response as its only argument and should return the shortened URL or an Error object. Otherwise the response will be treated as the shortened URL.
errorback	ShortenerErrorback	Optional	If provided, will be called with the error that occurred as its only argument and should return the shortened URL or an Error object. Otherwise the error will be handled as is.

Example

```
function createBitlyShortener(url) {  
  // For more info see http://dev.bitly.com/authentication.html#apikey  
  var login = '<your bitly login>';  
  var apiKey = '<your bitly api-key>';  
  
  return {  
    method: 'GET',  
    url: (  
      window.location.protocol === 'http:'  
      ? 'http://api.bit.ly/v3/shorten'
```

```

    : 'https://api-ssl.bit.ly/v3/shorten'
  ),
  data: {longUrl: url, apiKey: apiKey, login: login},
  responseType: 'json',
  callback: function(response) {
    if (response.status_code === 200) {
      return response.data.url;
    } else {
      return new Error(response.status_txt);
    }
  },
};
}

```

ResponseType

Type: string **Possible values:** * 'arraybuffer' * 'blob' * 'json' * 'text'

For more info see MDN's XMLHttpRequest.responseType.

ShortenerCallback

Type: (response) -> URL | Error

ShortenerErrorback

Type: (error: Error) -> URL | Error

Message

Type: object

Field	Type	Required?	Description
id	string	Required	The ID of the message used to mainly to remember if it was shown before.
type	MessageType	Required	The type of the message, used to style it. For more info see MessageType
text	string	Required	The body of the message.
showOnce	boolean	false	When set to true , once the user closes the message it will not be shown again.

Examples

```

{
  id: 'odyssea reprocessing from 01/07/2016 to 03/02/2017',
  type: 'warning',
  text: 'All Odyssea products are currently being reprocessed from July 2016 until the 3<sup>rd</sup> F',
  showOnce: true,
}
{

```

```

    id: 'test_info',
    type: 'info',
    text: 'This is an info message.',
  }
  {
    id: 'test_warning',
    type: 'warning',
    text: 'This is a warning message.',
  }
  {
    id: 'test_error',
    type: 'error',
    text: 'This is an error message.',
  }
  {
    id: 'test_long',
    type: 'info',
    text: 'This is a loooooooooooooooooooooong loooooooooooooooooooooong loooooooooooooong long long loooooooooooooong lo
  }
  {
    id: 'test_multi_lines',
    type: 'info',
    text: 'This is a message<br>on multiple<br>lines.',
  }
  {
    id: 'test_multi_spaces',
    type: 'info',
    text: 'This is a message      with      lots of      spaces.',
  }
}

```

MessageType

Type: string Possible values: * 'info' * 'warning' * 'error'

Hotspot

Type: string|object

string (hotspot permalink)

Can be found by Right-Clicking a hotspot item (from the “Hotspots” panel) and choosing “Copy link address” (or equivalent).

Notes

- Not to be confused with a share permalink (or a live URL).
- Not to be confused with a “bit.ly” link.

Example

```
'?name=Chlorophyll%20A%20and%20Globcurrent%20%40%20Agulhas%20region&date=1412426592000&timespan=1d%3B1w'
```

object

Field	Type	Required?	Description
name	string	Required	The name shown in the “Hotspots” panel
description	string	Optional	A URL of a description page
date	Date	Optional	If provided, the date of the hotspot
extent	number[4]	Optional	The extent of the viewport defined as [left, bottom, right, top] all in the projection projection.
timespan	Timespan	Optional	The timespan used
products	string[]	Optional	A list of product ids
opacity	number[]	Optional	A list of product opacities
stackLevel	number[]	Optional	A list of product stack levels

Example

```
{  
  name: 'Chlorophyll A and Globcurrent @ Agulhas region',  
  description: '',  
  date: new Date(1412426592000), // 2014-10-04T12:43:12.000Z  
  extent: [768344.98869256, -5048359.949897, 4280779.311964, -3243223.0901656],  
  timespan: '1d',  
  products: [  
    '900913_Chlorophyll_a_concentration_VIIRS',  
    '900913_GlobCurrent_L4_geostrophic_streamline',  
  ],  
  opacity: [100, 50],  
  stackLevel: [50, 90],  
}
```

DrawableShape

Type: object

Field	Type	Required?	Description
title	string	Required	The tooltip of the toolbar button that allows the user to draw that shape.
cssClassNames	string	Optional	The CSS class names added to the toolbar button that allows the user to draw that shape. Useful for customizing the icon.

Field	Type	Required?	Description
wkt	string	Required	The geometry of the drawable shape formatted as WKT in the projection projection .

For each `DrawableShape` a button is added in the user shapes toolbar. For more info see `Shapes`.

Example

```
{
  title: 'Right Arrow',
  cssClassNames: 'olControlDrawFeatureRightArrow',
  wkt: 'MULTILINESTRING((0 0, 100 0),(90 10, 100 0, 90 -10, 90 10))',
}
```

Shape

Type: object

Field	Type	Default	Description
<i>General</i>			
id	string	Optional	The ID of the shape used to generate the corresponding dataset's ID. If not given, the index of the shape in it's product's <code>shapes</code> array will be used.
type	ShapeType	Required	The type of the shape. For more info see <code>ShapeType</code>
text	string	Optional	The label displayed with the shape. Also used for filtering.
start	Date or number	Optional	If provided, the shape will be hidden before that date. A number is considered as milliseconds since 1 January 1970 00:00:00 UTC.
end	Date or number	Optional	If provided, the shape will be hidden starting from that date. A number is considered as milliseconds since 1 January 1970 00:00:00 UTC.

Field	Type	Default	Description
<i>Arrow Shape</i>			
points	number [2] [2]	Required	<i>Applies to "ARROW" shapes only</i> A couple of coordinates pairs defined as [lon, lat] both in EPSG:4326. The first pair marks the start (the tail) of the arrow. The second marks the end (the head) of the arrow.
arrowSize	number > 0	15000	The size in meters of the arrow head.
<i>Text Shape</i>			
location	number [2]	Required	<i>Applies to "TEXT" shapes only</i> The coordinates of the shape defined as [lon, lat] both in EPSG:4326.
<i>Line Shape</i>			
points	number [2] []	Required	<i>Applies to "LINE" shapes only</i> A list of at least two coordinate pairs defined as [lon, lat] both in EPSG:4326.
arrow	ShapeArrow	Optional	The description of an optional arrow head to rendered on the line. For more info see ShapeArrow
<i>Ellipse Shape</i>			
center	number [2]	Required	<i>Applies to "ELLIPSE" shapes only</i> The coordinates of the center of the ellipse defined as [lon, lat] both in EPSG:4326.
rx	number > 0	Required	The radius of the ellipse on the x-axis in meters .
ry	number > 0	rx	The radius of the ellipse on the y-axis in meters .
angle	number [-180, 180]	0	The amount of rotation applied to the ellipse in degrees (counter-clockwise).
arrow	ShapeArrow	Optional	The description of an optional arrow head to rendered on the line. For more info see ShapeArrow
<i>Mushroom Shape</i>			
			<i>Applies to "MUSHROOM" shapes only</i>

Field	Type	Default	Description
baseWidth	number >= 0	Required	The distance between the two vertical edges of the base of the mushroom in meters .
baseHeight	number >= 0	Required	The length of base of the mushroom in meters .
baseAngle	number [-45, 45]	0	The amount of rotation applied to the base of the mushroom in degrees (counter-clockwise).
leftLobeCenter	number [2]	Required	The coordinates of the center of the left lobe defined as [lon, lat] both in EPSG:4326.
leftLobeRx	number >= 0	Required	The radius of the left lobe ellipse on the x-axis in meters .
leftLobeRy	number >= 0	Required	The radius of the left lobe ellipse on the y-axis in meters .
rightLobeCenter	number [2]	Required	The coordinates of the center of the right lobe defined as [lon, lat] both in EPSG:4326.
rightLobeRx	number >= 0	Required	The radius of the right lobe ellipse on the x-axis in meters .
rightLobeRy	number >= 0	Required	The radius of the right lobe ellipse on the y-axis in meters .
<i>Generic Shape</i>			<i>Applies to “WKT” shapes only</i>
author	string	Optional	The author associated to the shape.
wkt	string	Required	The geometry of the shape formatted as WKT in EPSG:4326.

Note: All “TEXT” shapes require the `text` field to contain a none-empty string.

Note: All “ARROW” shapes require the `points` field to contain exactly 2 points.

For more info see Shapes.

Examples

```
{
  type: 'TEXT',
  text: 'Text goes here',
  start: 1459814400000,
  end: 1459900800000,
```

```

    location: [133.23063184654524, -21.083556353366014]
  }
  {
    type: 'ARROW',
    text: '',
    start: 1459814400000,
    end: 1459900800000,
    points: [[28.762538978264597, -33.09396148741701], [27.872646400139594, -33.745014639370424]]
  }
  {
    type: 'LINE',
    text: '',
    start: 1459814400000,
    end: 1459900800000,
    points: [[132.48356160920454, -26.78394450491424], [136.1749678592054, -26.469667679940457]],
    arrow: {
      directionFlipped: false,
      sideFlipped: false,
      ratio: 0.5,
      density: 0.5,
      offset: 0.5
    }
  }
}
{
  type: 'ELLIPSE',
  text: 'C',
  start: 1459814400000,
  end: 1459900800000,
  center: [134.02164747155183, -23.945171626522214],
  rx: 244598.49047851562,
  ry: 244598.49047851562,
  angle: 0,
  arrow: {
    directionFlipped: false,
    sideFlipped: false,
    ratio: 0.5,
    density: 0.2,
    offset: 0.125
  }
}
{
  type: 'MUSHROOM',
  text: '',
  start: 1486944000000,
  end: 1487203200000,
  baseWidth: 955.4628534317018,
  baseHeight: 1997.5447529548906,
  baseAngle: 0.6778513460412228,
  leftLobeCenter: [-4.814060593358906, 48.0702570920605],
  leftLobeRx: 761.6419180129138,
  leftLobeRy: 645.4213335569948,
  rightLobeCenter: [-4.8014029583976665, 48.07037335406251],

```



```

    rightLobeRx: 739.6453667184011,
    rightLobeRy: 664.7914310097694
  }
  {
    type: 'WKT',
    text: 'text',
    start: 1149163200000,
    end: 1149163200000,
    author: 'anonymous',
    wkt: 'POLYGON((77.51953122494571 59.26588064106204,40.95703122494571 44.339565266015946,64.6874999749
  }

```

ShapeType

Type: string **Possible values:** * 'TEXT' * 'ARROW' * 'LINE' * 'ELLIPSE' * 'MUSHROOM' * 'WKT'

Note: All “TEXT” shapes require the `text` field to contain a none-empty string.

Note: All “ARROW” shapes require the `points` field to contain exactly 2 points.

ShapeArrow

Type: object

Field	Type	Default	Description
<code>directionFlipped</code>	boolean	<code>false</code>	If <code>true</code> the arrow will point from the end of the line segment to it's start, instead of start to end. In the case of an ellipse <code>false</code> means counter-clockwise and <code>true</code> means clockwise.
<code>sideFlipped</code>	boolean	<code>false</code>	If <code>true</code> the colors of the arrow head are swapped.
<code>ratio</code>	number [0, 1]	0.5	The ratio of the proportion of the left triangle and the right.
<code>density</code>	number [0, 1]	<i>see below</i>	The density of arrow heads on the shape. 0 means one arrow head, and 1 means arrow head on each line segment.
<code>offset</code>	number [0, 1]	<i>see below</i>	The offset of the first arrow head. 0 means on the first line segment, and 1 means on the last.
<code>size</code>	number > 0	15000	The size in meters of the arrow head.

Default values by type

Type	density	offset
'LINE'	0.5	0.5
'ELLIPSE'	0.2	0.125

Example

```
{
  directionFlipped: false,
  sideFlipped: false,
  ratio: 0.5,
  density: 0.2,
  offset: 0.125
}
```

ShapeStyleMap

Type: object

Field	Type	Default	Description
ALL	ShapeStyle	Optional	The style applied to all shapes.
TEXT	ShapeStyle	Optional	The style applied to “TEXT” shapes. Overrides the “ALL” style.
ARROW	ShapeStyle	Optional	The style applied to “ARROW” shapes. Overrides the “ALL” style.
LINE	ShapeStyle	Optional	The style applied to “LINE” shapes. Overrides the “ALL” style.
ELLIPSE	ShapeStyle	Optional	The style applied to “ELLIPSE” shapes. Overrides the “ALL” style.
WKT	ShapeStyle	Optional	The style applied to “WKT” shapes. Overrides the “ALL” style.

Example

```
{
  ALL: {
    strokeColor: '#00EE00',
    fillOpacity: 0,
    fontColor: '#00EE00',
    fontSize : 16,
  },
  WKT: {
    strokeColor: '#AAAAEE',
    fillOpacity: 0.4,
    strokeDashstyle: 'dash',
    fontColor: '#AAAAEE',
    fontSize : 12,
  }
}
```

```

},
}

```

ShapeStyle

Type: object

Field	Type	Default	Description
pointRadius	number > 0	10	The radius in pixels of the point disk marker when rendered on the screen.
fillColor	SVGPaint	'#AAAAAA'	See MDN's SVG Paint
fillOpacity	number [0, 1]	0.4	The opacity of the fill color. 0.0 (fully transparent) to 1.0 (fully opaque).
strokeColor	SVGPaint	'#AAAAEE'	See MDN's SVG Paint
strokeOpacity	number [0, 1]	1	The opacity of the stroke color. 0.0 (fully transparent) to 1.0 (fully opaque).
strokeWidth	string or number	2	The width of the outline on the polygon. If a value of 0 is used the outline will never be drawn. MDN's SVG stroke-width
strokeDashstyle	string	'solid'	Controls the pattern of dashes and gaps used to stroke the polygon. Can be one of ('dot', 'dash', 'dashdot', 'longdash', 'longdashdot', 'solid') or a list of white space separated <length>s and <percentage>s that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, '5 3 2' is equivalent to '5 3 2 5 3 2'.
fontColor	SVGPaint	'#AAAAAA'	See MDN's SVG Paint

Field	Type	Default	Description
fontSize	number >= 0	12	The text size in pixels when rendered on the screen.

Example

```
{
  strokeColor: '#00EE00',
  fillOpacity: 0,
}
```

Layer

Type: `OpenLayers.Layer`

For more info see OpenLayers 2's official documentation.

Example

```
new OpenLayers.Layer.Google('Google Satellite', {
  type: google.maps.MapTypeId.SATELLITE,
  numZoomLevels: 20,
  maxExtent: new OpenLayers.Bounds(maxExtent),
}),
new OpenLayers.Layer.Google('Google Streets', {
  type: google.maps.MapTypeId.ROADMAP,
  numZoomLevels: 22,
  maxExtent: new OpenLayers.Bounds(maxExtent),
}),
```

TimeRange

Type: `[string, Timespan, boolean] | [string, Timespan]`

The elements of the tuple are as follows:

1. The label of the time range.
2. The width of the time range.
3. `true` if the time range should be selected by default, otherwise `false`.

For more info see Timeline.

Examples

```
['6-Hour', '6h']
['Daily', '1d', true]
```

Timespan

Type: `string`

Syntax

```
<Timespan> ::= <number> <unit>
<number> ::= <a positive decimal number>
<unit> ::= "s" # second
          | "M" # minute
          | "h" # hour
          | "d" # day
          | "w" # week
          | "m" # month
          | "y" # year
          | "D" # decade
          | "c" # century
```

Examples

- '6h' 6 hours timespan
- '1m' 1 month timespan

MarkerSymbols

Type: {[name: string]: MarkerSymbol}

Example

```
{
  left_arrow_tip: [-8,4, -8,-4, 0,0, -8,4],
}
```

MarkerSymbol

Type: number[]

A list of coordinate pairs (x and y) (much like the MDN's SVG polygon points) which will form a polygon anchored on (0, 0).

Example

```
[-8,4, -8,-4, 0,0, -8,4]
```

gives the following SVG polygon:

```
<polygon points="-8,4 -8,-4 0,0 -8,4"></polygon>
```



Figure 1: left arrow tip

Product

Type: object

Field	Type	Default	Description
<i>General</i>			
label	string	Required	The label of the product (used in the products side panel among other places)
id	string	Required	The ID of the product (used in services requests among other places)
type	ProductType	Required	The type of the product's datasets and how they are rendered. For more info see ProductType
description	string	Optional	The description of the product (shown as a tooltip in products side panel)
infoURL	URL	Optional	The URL of a GitHub Flavored Markdown file containing information about this product.
hidden	boolean	false	If true the product not shown in the left products list. For more info see Products
selected	boolean	false	If true the product is selected by default and hidden is ignored.
colorbar	URL	Optional	The URL of the colorbar displayed for this product in the Settings popup.
<i>Timeline</i>			
mustBeCurrent	boolean	Required	If true the product's datasets are considered valid (and thus added to the map) only when the current date/time is inside their respective definition range. Otherwise (if false) they are considered valid if their respective definition range intersects the current timespan.

Field	Type	Default	Description
<code>timelineValidity</code>	<code>number > 0</code>	Optional	If provided and <code>mustBeCurrent</code> is set to <code>true</code> , the product's datasets are considered valid if their respective definition range intersects a range of \pm the value of this field in hours around the current date/time.
<code>permanent</code>	<code>boolean</code>	<code>false</code>	If <code>true</code> the product's datasets are excluded for the availability and find nearest requests.
<i>Pagination</i>			
<code>priority</code>	<code>number</code>	<code>0</code>	For more info see <i>Pagination</i> . The priority of the product. Bigger values imply higher priority.
<code>datasetWeight</code>	<code>number >= 0</code>	<code>1</code>	The average rendering weight of the product's datasets.
<code>minPageSize</code>	<code>number >= 0</code>	<code>1</code>	The minimum number of the product's datasets that will be rendered (if available) regardless of any pagination.
<i>General Rendering</i>			
<code>noOutline</code>	<code>boolean</code>	<code>false</code>	<i>Applies to all types</i> If <code>true</code> the outlines (in white around the data) of the product's datasets will be hidden.
<code>mapMinZoom</code>	<code>integer >= 0</code>	Optional	If provided the product's datasets will be hidden for zoom level smaller (farther away) than the value of this field.
<code>mapMaxZoom</code>	<code>integer >= 0</code>	Optional	If provided the product's datasets will be hidden for zoom level bigger (nearer) than the value of this field.
<code>opacity</code>	<code>number [0, 1]</code>	<code>1</code>	The opacity of the datasets. 0.0 (fully transparent) to 1.0 (fully opaque).

Field	Type	Default	Description
<code>stackLevel</code>	integer [0, 1000]	Required	Specifies the z-order of the product's datasets. When datasets overlap, z-order determines which one covers the other. A dataset with a larger <code>stackLevel</code> generally covers an element with a lower one.
<i>Tile Rendering (ZXY)</i>			
<code>tileServers</code>	URL []	Optional	<i>Applies to "ZXY" products only</i> A list of URLs to use for requesting tiles. All URLs MUST be equivalent, a list is given in order to circumvent the browser's limitation on the number of simultaneous requests to one domain. If not specified <code>dataServers</code> will be used.
<code>tileOrigin</code>	number [2]	Required	The coordinates of the tile (0, 0) defined as [lon, lat] both in the projection <code>projection</code> .
<code>tileSize</code>	number [2]	Required	The size of each tile defined as [width, height] both in pixels.
<i>Dynamic Rendering</i>			
<code>fields</code>	Fields	Required	<i>Applies to some product types and with restrictions, see Product Dynamic Rendering Notes</i> Defines the fields (or channels) found in the datasets with their min and max. For more info see Fields
<code>palette</code>	Palette	Required	The color palette used to render the product's datasets. For more info see Palette
<code>density</code>	number [0, 2]	1	The density used when rendering the product's datasets.
<i>Moorings Rendering</i>			
			<i>Applies to "MOORED" products only</i>

Field	Type	Default	Description
<code>marker</code>	MarkerStyle	Optional	Defines the pictogram displayed at the location of the dataset. Defaults to a buoy icon. For more info see MarkerStyle.
<i>Trajectory Rendering</i>			
<code>marker</code>	Marker	Optional	<i>Applies to "TRAJECTORIES" products only</i> The trajectory marker to render to indicate the position and direction at the closest (in the past) point of the trajectory to the current date/time. Defaults to a diamond shape. For more info see Marker.
<code>markerPosition</code>	number [0, 1]	Optional	If provided specifies a fixed position (independent of the current date/time) of the marker. 0 being the first point of the trajectory and 1 the last.
<code>startMarker</code>	Marker	Optional	The trajectory marker to render at the start of the trajectory. For more info see Marker.
<code>endMarker</code>	Marker	Optional	The trajectory marker to render at the end of the trajectory. For more info see Marker.
<code>withRangeMarkers</code>	boolean	<code>false</code>	If <code>true</code> brackets will be rendered around the part of the trajectory that is in the current timespan.
<i>Shapes</i>			
<code>style</code>	ShapeStyleMap	Optional	<i>Applies to "USER_SHAPES" and "SERVER_SHAPES" products only</i> The shapes rendering style description. For more info see ShapeStyleMap.
<code>shapes</code>	Shape []	Required	A list of predefined shapes. For more info see Shapes. (<i>Applies to "SERVER_SHAPES" products only.</i>)

Field	Type	Default	Description
<i>Layer</i>			
<code>layer</code>	Layer	Required	Any OpenLayers layer to be used as the one and only dataset of the product.
<code>validFrom</code>	Date	Optional	If provided, the layer will be hidden before that date.
<code>validTill</code>	Date	Optional	If provided, the layer will be hidden starting from that date.
<code>updateLayer</code>	function	Optional	If provided, gets called once on startup and whenever the current date changes and passed the layer and the current date. It allows the product to update its layer when the current date changes.
<i>ncWMS</i>			
<code>ncwmsURL</code>	URL	Required	The URL of the ncWMS.
<code>ncwmsParams</code>	Object	{}	The parameters to pass with every request to the ncWMS server.
<code>ncwmsQueryAdapter</code>	function	Optional	If provided, gets called when a dataset is created and each time it is added to the map. This function receives a dataset object and must return an object with <code>url</code> and <code>params</code> (dict containing modified query parameters) properties.

Product Dynamic Rendering Notes

Required restrictions on `fields` field by type:

type	Restrictions
ARROWS	<code>fields</code> must contain an <code>angle</code> and a <code>modulus</code> fields
BARBS	<code>fields</code> must contain an <code>angle</code> and a <code>modulus</code> fields

Required restrictions on `palette` field by type:

type	Restrictions
STREAMLINES	palette as UniformPalette
TRAJECTORIES	palette with palette.field
GEOJSON	palette with palette.field

The `density` field is restricted to the following types:

- ARROWS
- BARBS
- STREAMLINES

Example

```
{
  /* General */
  label: 'Drifters 1000m North Atlantic speed (ANDRO)',
  id: '3857_ARGO_Deep_NATL1000',
  type: 'TRAJECTORIES',
  selected: false,

  /* Timeline */
  mustBeCurrent: false,

  /* Pagination */
  datasetWeight: 0.1,

  /* General Rendering */
  mapMinZoom: 2,
  stackLevel: 90,

  /* Dynamic Rendering */
  palette: {
    field: 'speed',
    colormap: 'images/palettes/undefined',
    min: 0,
    max: 0.35,
  },

  /* Trajectory Rendering */
  marker: argoMarker, // argoMarker is a function defined elsewhere in the config file.
  markerPostion: 1,
}
```

ProductType

Type: string **Possible values:** * 'LAYER' * 'NCWS' * 'IMAGE' * 'ZXY' * 'ARROWS' * 'BARBS' * 'STREAMLINES' * 'TRAJECTORIES' * 'GEOJSON' * 'MOORED' * 'USER_SHAPES' * 'SERVER_SHAPES'

LAYER

Used for third-party/external layers like Google Earth Engine, SHOM, and Meteosat.

NCWMS

Used for ncWMS based layers like CMEMS ones.

IMAGE

Used for non-tiled raster datasets. Allows unconstrained zoom.

ZXY

Used for tiled raster datasets.

ARROWS

Used for arrow vector fields.

BARBS

Used for barbs vector fields.

STREAMLINES

Used for streamline vector fields.

TRAJECTORIES

Used for trajectories with the notion of time. Allows to have a marker.

GEOJSON

Used for trajectories without the notion of time (like synoptic datasets).

MOORED

Used for moored buoys.

USER_SHAPES

See User shapes section.

Group

Type: object

Field	Type	Default	Description
label	string	null	The label of the group (used in the products side panel if given)
products	string[]	Required	An ordered list of IDs of the products in that group

Note: All product IDs MUST be IDs of existing products in the config. **Note:** All product IDs MUST be unique (a product can be in no group or in one group and only once in that group).

Color

Type: string

See MDN's CSS Color, note that only the `rgb()`, `#RRGGBB`, `#RGB`, and `rgba()` format are supported and percentages are not supported.

- `#RRGGBB` with RR, GG and BB being 2-digit hexadecimal numbers.
- `#RGB` with R, G and B being 1-digit hexadecimal numbers.
- `rgb(red, green, blue)` with red, green and blue being integers between 0 and 255.
- `rgba(red, green, blue, alpha)` with red, green and blue being integers between 0 and 255 and alpha between 0 and 1.

Examples

- `#FF0000` is red
- `#F00` is red
- `rgb(255, 0, 0)` is red
- `rgba(255, 0, 0, 0.5)` is red with 50% opacity.

Fields

Type: {[name: string]: Field}

See [Product Dynamic Rendering Notes](#) for type specific restrictions.

Field

Type: object

Field	Type	Required?	Description
channel	string or number	Optional	The dataset's channel represented by this field. Defaults to the name of this field.
min	number	Required	The minimum value.
max	number	Required	The maximum value.

Example

```
{  
  modulus: {  
    channel: 0,  
    min: 0,  
    max: 10,  
  },  
  temp: {  
    min: 3.0,  
    max: 12.5,  
  },  
}
```

```
  },  
}
```

Palette

Type: UniformPalette|ColormapPalette

See [Product Dynamic Rendering Notes](#) for type specific restrictions.

UniformPalette (Single Color)

Field	Type	Required?	Description
field	string	Optional	The name of the field defined in <code>fields</code> of the same product.
uniform	Color	Required	The color to apply uniformly.

Example

```
{  
  field: 'temp',  
  uniform: '#F00',  
}
```

ColormapPalette

Field	Type	Required?	Description
field	string	Required	The name of the field defined in <code>fields</code> of the same product.
colormap	URL	Required	The URL of a CPT or a PNG palette.

Example

```
{  
  field: 'temp',  
  colormap: 'images/palettes/rainbow01.cpt',  
}
```

Marker

Type: MarkerStyle|(color: Color) -> MarkerStyle

Example

```
function argoMarker(color) {  
  return {
```

```

    graphicName: 'left_arrow_tip',
    pointRadius: 4, // 2*4 = 8px wide and heigh on screen
    strokeWidth: 1,
    strokeColor: color,
    fillColor: color,
  };
}

```

MarkerStyle

Type: MarkerStateStyle|object

Field	Type	Default	Description
default	MarkerStateStyle	Required	The style to use for the marker by default.
selected	MarkerStateStyle	Optional	The style to use for the marker when the dataset is selected. If not specified, uses the default style.

Note: If a single MarkerStateStyle is given (instead of an object), it will be used for all the states.

MarkerStateStyle

Type: ImageMarkerStyle|PolygonMarkerStyle

ImageMarkerStyle

Type: object

Field	Type	Default	Description
rotation	number	0	The angle in degrees between the marker pointing to the right and it's position in the image CW. Use NaN to prevent rotation.
externalGraphic	URL	Required	The URL to the external image to use.
graphicWidth	number > 0	Required	The width in pixels of the marker when rendered on the screen.
graphicHeight	number > 0	Required	The height in pixels of the marker when rendered on the screen.
graphicXOffset	number	-width / 2	The pixel offset along the positive x axis (left to right) for displacing the image.

Field	Type	Default	Description
<code>graphicYOffset</code>	number	<code>-height / 2</code>	The pixel offset along the positive y axis (top to bottom) for displacing the image.

Example

```
{
  rotation: 90,
  externalGraphic: 'css/images/marker-white.png',
  graphicWidth: 14,
  graphicHeight: 17,
}
```

PolygonMarkerStyle

Type: object

Field	Type	Default	Description
<code>rotation</code>	number	0	The angle in degrees between the marker pointing to the right and its defined position CW. Use NaN to prevent rotation.
<code>graphicName</code>	string	Required	The name of the symbol to render. Can be one of ('circle', 'star', 'cross', 'x', 'square', 'triangle'), or one of the fields defined in <code>markerSymbols</code> .
<code>pointRadius</code>	number > 0	Required	Half the width in pixels of the marker when rendered on the screen.
<code>fillColor</code>	SVGPaint	Optional	See MDN's SVG Paint
<code>fillOpacity</code>	number [0, 1]	1	The opacity of the fill color. 0.0 (fully transparent) to 1.0 (fully opaque).
<code>strokeColor</code>	SVGPaint	Optional	See MDN's SVG Paint
<code>strokeOpacity</code>	number [0, 1]	1	The opacity of the stroke color. 0.0 (fully transparent) to 1.0 (fully opaque).
<code>strokeWidth</code>	string or number	1	The width of the outline on the polygon. If a value of 0 is used the outline will never be drawn. MDN's SVG <code>stroke-width</code>

Field	Type	Default	Description
<code>strokeDashstyle</code>	<code>string</code>	Optional	Controls the pattern of dashes and gaps used to stroke the polygon. Can be one of ('dot', 'dash', 'dashdot', 'longdash', 'longdashdot', 'solid') or a list of white space separated <length>s and <percentage>s that specify the lengths of alternating dashes and gaps. If an odd number of values is provided, then the list of values is repeated to yield an even number of values. Thus, '5 3 2' is equivalent to '5 3 2 5 3 2'.

Example

```
{
  graphicName: 'left_arrow_tip',
  pointRadius: 4, // 2*4 = 8px wide and height on screen
  strokeWidth: 1,
  strokeColor: 'red',
  fillColor: 'red',
}
```