HITACHI
Inspire the Next

NewMapComponent

Kleyson Rios
November 2017

Agenda

- Introduction
- Main features
- Demo

Introduction

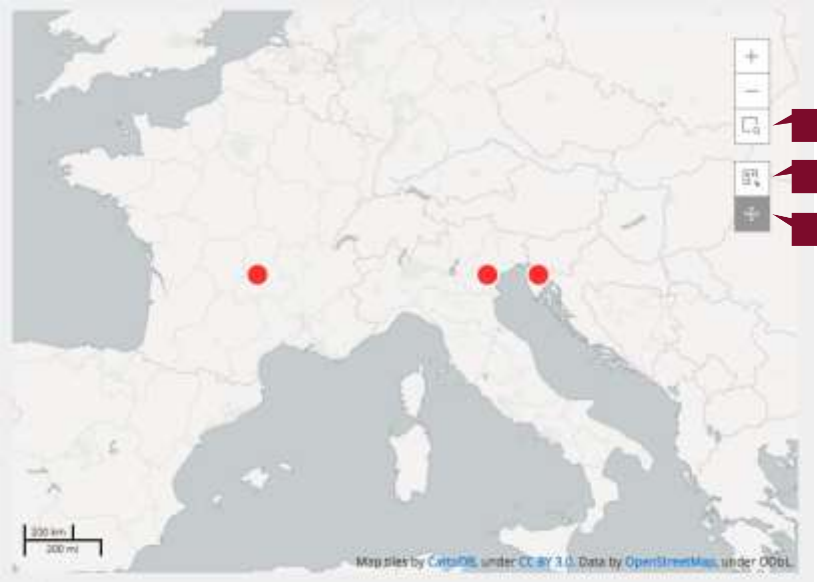
- Map engine
- Create dashboards with georeferenced data
- Markers and Shapes
- Library Openlayers2 and GoogleMaps*****

**** Require API key***

Map States

Operation Modes

- Panning mode (pan)
  - Allows navigation over the map, e.g. dragging.
- Zoom box mode (zoombox)
  - The user can draw a rectangle to define a new zoom area.
- Selection box mode* (selection)
  - Allows the drawing of a rectangle that selects a set of items. The items inside the rectangle are added to the selection list. Clicking on an item toggles its selection state, whereas clicking on an empty area of the map clears the list of selected items.


200 km
200 mi
Map tiles by CartoDB, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

Zoom box mode

Selection mode

Panning mode

***** To active the Selection mode, set the Parameter property.*

Feature States

State-dependent styles

- **disabled**
  - When the map does not behave as a selector (map is not in Selection mode)
- **noneSelected**
  - When no item is selected.
- **allSelected**
  - When all items are selected.
- **someSelected**
  - When at least one item is selected.
- **selected**
  - When the item selected.
- **unselected**
  - When the item is not selected.
- **normal**
  - When the mouse is not over an item.
- **hover**
  - When the mouse is over an item.

Styling the Map

More Details

- The combination of the map type (marker/shape), map operation mode and item state defines the individual style for each item on the map.
  - Example:

    global/markers/shapes : pan/zoombox/selection : featureStates : Attribute

- Styling based on the CSS concept, where the more specific definitions have priority over the more generic ones.
  - Hierarchy

    - global
    - markers/shapes
    - disabled/noneSelected/someSelected/allSelected
    - pan/zoombox/selection
    - unselected/selected
    - normal/hover
    - attribute
    - calculated attributes based on the result set*

Changing the style map via preExecution:

```
this.styleMap = {
  markers:  {
    cursor:  "pointer”
  }
};
```

Features

Attributes

- **Define the visual style for the item**
  - Stroke, color, opacity, etc
- **In general, they are SVG attributes.**
  - Openlayers2 attributes.

- **Static values**
  - Set via styleMap
- Dynamic values.
  - Based on business rules and/or dataset.

Define a new dynamic attribute via preExecution:

```
this.attributeMapping.fill = function(context, seriesRoot, mapping, row) {
  var value = row[2];
  if ( _.isNumber(value) && value < 25 ) {
    return "green";
  } else {
    return "red";
  }
};
```

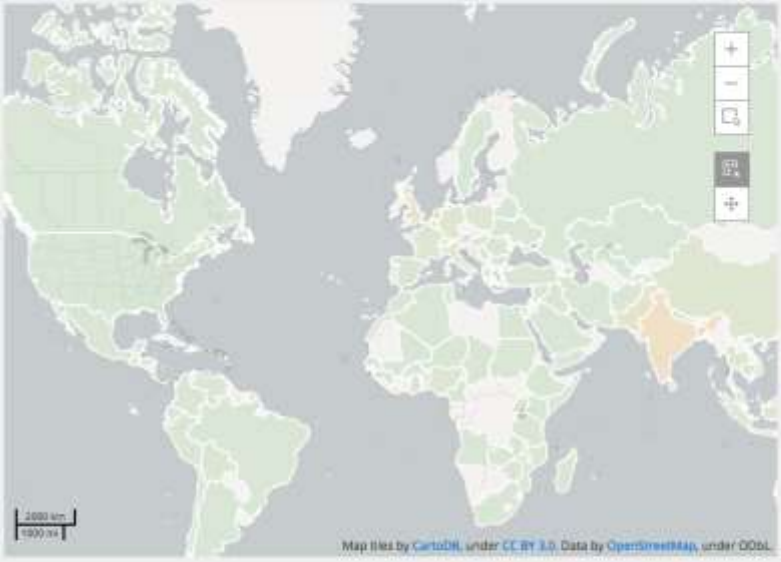
Features

Color mapping

- Based on a colour palette
- Require a column called **fill** in the resultSet
- Colour palette can be changed

Change the color palette via preExecution:

```
this.colormap = [
                              "[213,238,249]",
                              "[194,231,246]",
                              "[175,223,244]",
                              "[156,216,241]",
                              "[137,208,239]",
                              "[117,201,236]",
                              "[98,193,233]",
                              "[79,186,231]",
                              "[60,178,228]",
                              "[41,171,226]"
                           ];
```

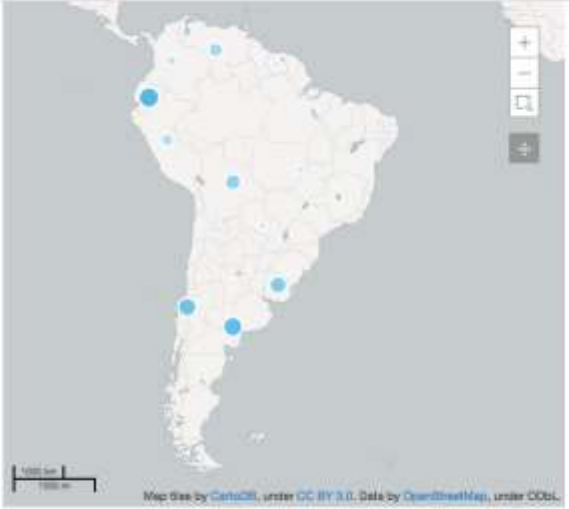

2000 km
1000 mi
Map tiles by CartoDB, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

Markers

Size

- Automatic radius size
- Require a numeric column called **r** in the resultSet
- Sizes are based on a Min and Max values

```
/*
 * Set Min and Max size (pixel) for the markers based on the r values
*/
this.scales.r = [3, 10];
```


+
−
1000 km
1000 mi
Map tiles by CartoDB, under CC BY 3.0. Data by OpenStreetMap, under ODbL.

Mandatory columns

- Needed to correctly index the data in the resultSet
  - id (unique name)
  - fill (color mapping)
  - r (dynamic radius)
  - latitude
  - longitude

ResultSet:

City – Age – Salary – Lat – Lng

Redefining mapping columns via preExecution:

```
this.visualRoles = {
        id: 0,
        fill: 1,
        r: 2,
        latitude: 3,
        longitude: 4
};
```

More ...

- Automatic zoom
- Popup’s
- Markers with CGG charts
- Events (clicks, mouseover)
- Addins
  - Latitude/Longitude lookup
  - Shape resolver
  - Marker image
- Shapefile formats (geoJSON, kml, simpleJSON)
- Handle the engine’s API*

***** Not supported**

Learn ...

- CDE Samples: Public / plugin-samples / CDE / Require Samples / CDE Reference / NewMapComponent
- Source code (https://github.com/webdetails/cde/tree/master/cde-core/resource/resources/custom/amd-components/Map)
- Community
- NMC-Samples repository* (https://github.com/kleysonr/NMC-samples)

**** Might have not supported samples.***

Demo

HITACHI
Inspire the Next

Thank you !

Twitter: @kleysonr
Github: https://github.com/kleysonr
Blog: http://kleysonrios.blogspot.com