SuperDARN Catalogue App Documentation

Release 1.0.0

Tim Barnes, British Antarctic Survey

July 17, 2017

CONTENTS

Contents:

CHAPTER

ONE

GETTING STARTED

This guide assumes you already have a working Apache configuration capable of talking to an instance of a SuperDARN Catalogue API.

1.1 Setting up the app

The repository contains a working `index.html` file, however you may wish to create your own (for instance, if you have specific style sheets you want to use). The repository also contains a number of Javascript and CSS files, some of which are snapshots of third party software e.g. JQuery. The provided versions have been tested and work with this app. You may use different versions than those provided, but be advised that in some cases, the use of different versions may prevent certain parts of the app from working or even prevent the app from working as a whole.

1.2 Required Javascript libraries

The following Javascript libraries need to be called:

- `/js/jquery/jquery-1.12.4.min.js`
- `/js/jquery/jquery-ui.min.js`
- `/js/flot/jquery.flot.min.js`
- `/js/flot/jquery.flot.selection.min.js`
- `/js/flot/jquery.flot.crosshair.min.js`
- `/js/flot/jquery.flot.navigate.min.js`
- `/js/sdcat/SDCat.js`
- `/js/SDCat.App.js`
- `/js/SDCat.CoveragePlot.js`
- `/js/catalogue-plot/CataloguePlot.js`

1.3 Required CSS libraries

The following CSS libraries need to be called:

- `/css/jquery-ui/jquery-ui.min.css`

- `/css/SDCat.App.css`
- `/css/catalogue-plot/CataloguePlot.css`

1.4 Other requirements

Your `index.html` page requires a container div into which the app shall be placed. This div should have the ID `sdCatContainer`, though if you decide to use a different ID, this can be specified in the app initialisation.

```
<div id='sdCatContainer'></div>
```

Once all this is in place, the app can be initialised.

CHAPTER

TWO

INITIALISING THE APP

To initialise the app with the default settings, add the following code:

```
var options = {
   sdcat: {
      url: "<url_to_your_sdcat_api_instance>"
   },
};
var app = new SDCat.App(options);
```

In this example, `<url_to_your_sdcat_api_instance>` is the URL to the SuperDARN Catalogue API instance you wish this app to query e.g **http://localhost:8082/api/v2/**. For a full list of options, see *SDCat.App*.

CHAPTER

THREE

USING THE APP

The app is divided into two sections - the sidemenu on the left and the main body on the right. The sidemenu houses all of the controls that are needed to generate the plots you required. The generated plot then appears in the main body of the page.

3.1 Sidemenu

The sidemenu houses all of the widgets used to control the app. Each control can be configured to be shown or hidden on the sidemenu. By default, all of the controls are set to be shown. To completely hide the sidemenu from the app, simply set the sidemenu **show** option to **false** at initialisation:

```
var options = {
   sdcat: {
      url: "http://localhost:8082/api/v2/"
   },
   sidemenu: {
      show: false
   }
};
var app = new SDCat.App(options);
```

The next sections deal with each of the individual controls and how they work.

3.1.1 Radars

The radars section is made up of multiple controls - the main one being the radar selection box. When the app is initialised, this selection box is populated with the radar codes of all SuperDARN radars available from your API. The user can then selected the ones they require.

Below the selection box are additional "shortcut" controls which can pre-selected groups of radars based on various parameters. The **North** and **South** buttons allow the user to select only the radars located in the Northern or Southern hemisphere respectively. The next control is a drop down box which allows users to select radars based on the institute that maintains them. The final drop down box allows users to select radars based on the country in which they are located. Lastly, there is a control which determines how to display the radar names in the selection box. By default, the app shows the radars as their unique three-letter ID code. Using this control, users can change the display to show the radar names instead.

The options to control whether or not to show these controls are:

- **showRadarList** for the radar list
- **showRadarHemisphereButtons** for the hemisphere buttons

- **showRadarInstituteSelection** for the institute selection box
- **showRadarCountrySelection** for the country selection box
- **showRadarNameDisplayButton** for the name display buttons

To change the default way the radars are listed, set the appropriate **radarNameOptions** item to **true** after initialisation:

```
var app = new SDCat.App(options);
app.items.radarNameOptions.list.code.selected = false;
app.items.radarNameOptions.list.name.selected = true;

app.display();
```

In this example, the **code** option is deselected and the **name** option is selected before the app is displayed.

3.1.2 File Types

The file types section is comprised of a single selection box populated with the different file types available from the API. Similarly to the radars list, users can select the file types they require from the list. To hide this control from the sidemenu, set the **showFileTypeList** option to **false**.

3.1.3 Date Range

The date range section allows the user to pick a date range over which to show the coverage. This section is made up of three controls. The first set of controls allow the user to pick a specific date range. The next set of controls allow the user to jump forwards or backwards in increments of the currently selected time periods. This is accomplished by using the **<** or **>** buttons. The user can also jump to the start or the end of the available date range by using the **<<** or **>>** buttons. Note that the time periods are expressed in days, so if the currently selected time period covers the entirety of a leap year, the jump buttons would go forward or backwards 366 days as opposed to a year. The final control is a drop down box of pre-selected date ranges.

The options to control whether or not to show these controls are:

- **showDatePicker** for the date picker and the jump buttons
- **showPreselectedDateRanges** for the date range drop down box.

Date ranges can be added to the drop down box programatically. This is accomplished by adding an additional element to the app object's **preselectedDateRanges** object:

```
var app = new SDCat.App(options);
app.preSelectedDateRanges["Last Month"] = {
   start: new Date(Date.UTC(app.now.getUTCFullYear(), app.now.getUTCMonth() - 1, 1,  0,  0,  0)),
   end:   new Date(Date.UTC(app.now.getUTCFullYear(), app.now.getUTCMonth(),     0, 23, 59, 59))
};
app.display();
```

This will add a new date range called "Last Month" to the list of date ranges.

3.1.4 Buttons

The final set of controls are a collection of buttons which perform various tasks. The information for the buttons shown in this part of the sidemenu can be found in *SDCat.App.sidemenuButtons*. The buttons are as follows:

Select All

Clicking this button will select all the radars and file types.

Clear Selection

Clicking this button will clear any selected radars and file types.

Reset View

Clicking this button will reset the view of the plot after zooming in. At initialisation, this button is disabled.

Generate Plot

Clicking this button will generate the plot according to the parameters selected by the user.

3.2 Main Body

The main body houses the actual plot once it has been generated. It has been designed to share the same look and feel of the static plots available on the VT website. The difference here being that it is an interactive plot.

The plot shows the coverage of a dataset via bars, where the bars denote the time period where data is available for a particular radar and the colour of the bar represents the data type. The time is along the x-axis and the different radars are along the y-axis.

Using the mouse scroll wheel, a user can zoom in and out of the plot. The user can also zoom into a specific date range on the plot by clicking and dragging out a selection box along the plot. To return the plot to the original view, click the **Reset View** button.

Hovering over a bar reveals a tooltip which contains the start and end date of the bar beneath the cursor. The display below the plot shows the date and time of the point where the cursor is located - rounded down to the nearest two hour block so missing two-hourly SuperDARN files can be identified quickly.

CHAPTER

FOUR

JAVASCRIPT DEFINITIONS

4.1 SDCat.App

This is an object used to create an instance of the Catalogue App itself.

4.1.1 Prerequisites

The following function libraries are required in order to use this object:

- `/js/sdcat/SDCat.js`
- `/js/SDCat.CoveragePlot.js`

4.1.2 Constructor

class `SDCat.`**`App`**(*options*)

`options`
(*Object*) - List of options. These options set attributes of the object:

`sdcat`
(*Object*) - Object containing configuration parameters for the SDCat object. Here, you need to specify the URL to the SDCat API instance you wish to query.

Example:

```
var options = {
   sdcat: {
      url: "http://localhost:8082/api/v2/"
   },
};
```

`plotId`
(*String*) - The ID of the div which will contain the coverage plot itself. (Default: `sdCat`).

`containerDivId`
(*String*) - The ID of the div which will contain the app itself. A div with this ID needs to exist within the body of the page. (Default: `sdCatContainer`).

sidemenu
(*Object*) - Object containing variables which determine whether to show/hide various elements of the sidemenu controls:

show
(*Boolean*) - Show the sidemenu itself (default: `true`).

showRadarList
(*Boolean*) - Shows the list of radars (default: `true`)

showRadarHemisphereButtons
(*Boolean*) - Shows the North/South buttons to pre-select radars by hemisphere (default: `true`)

showRadarInstituteSelection
(*Boolean*) - Shows a drop down box to pre-select radars by institute (default: `true`)

showRadarCountrySelection
(*Boolean*) - Shows a drop down box to pre-select radars by the country they are located in (default: `true`)

showFileTypeList
(*Boolean*) - Shows the list of available file types (default: `true`)

showDateTypeOpts
(*Boolean*) - Shows the date type control which allows users to determine the date source of the bars (default: `true`)

showDatePicker
(*Boolean*) - Shows the date picker, allowing users to select a custom date range (default: `true`)

showPreselectedDateRanges
(*Boolean*) - Shows a selection box populated with preselected date ranges (default: `true`)

showSelectAllButton
(*Boolean*) - Shows a button which selects all the radars and file types (default: `true`)

showGenerateButton
(*Boolean*) - Shows a button which generates the plot (default: `true`)

showClearButton
(*Boolean*) - Shows a button which clears the current selection (default: `true`)

showResetViewButton
(*Boolean*) - Shows a button to reset the view of the plot after zooming in (default: `true`)

selectBoxMaxSize
(*Integer*) - Sets the maximum size of a selection box. Used for the radar list and the file types list (default: `5`)

plotBarArea
(*Number*) - Sets the proportion of the radar area taken up by the coverage bar. The value must be between 0 and 1. If this value is set to less than 0, the app will set it to 0. If the value is greater than 1, the app will set it to 1. (Default: `0.8`).

radarArea
(*Number*) - Size of the radar area. This is used to calculate the height of the div which will contain the plot. This value is not a pixel value (default: `20`)

xAxisArea
(*Number*) - Size of the x-axis area (i.e. the area below the plot). This is used to calculate the height of the div which will contain the plot. This value is not a pixel value (default: `30`)

The height of the plot div is calculated to be `(radarArea * number_of_radars) + xAxisArea`.

`preselected`
(*Object*) - Object containing list of items to preselect within the app upon loading:

`radars`
(*Array*) - Array of radar codes to preselect in the app. (default: `[]`)

`fileTypes`
(*Array*) - Array of file types to preselect in the app. (default: `[]`)

4.1.3 Attributes

`SDCat.App.`**`sdcat`**
(*SDCat*) - This is the SDCat object which will be used to query the API. It will be created automatically using the parameters specified in the SDCat.App constructor.

`SDCat.App.`**`now`**
(*Date*) - A Javascript Date object containing the current time and date when the object was created.

`SDCat.App.`**`preselectedDateRanges`**
(*Object*) - Object containing the preselected date ranges for use in the Date Range dropdown box. A number of preselected date ranges have already been specified:

- Current Day
- Current Month
- Current Year
- Last 7 Days
- Previous Day
- Previous Month
- Previous Year

To add additional date ranges, add another object to this object where the attribute name is the name of the text that will appear in the dropdown box. The new object must have the following attributes:

`start`

(*Date*) - Start date of the date range in UTC.

`end`

(*Date*) - End date of the date range in UTC.

Example:

```
app.preSelectedDateRanges["Last Month"] = {
   start: new Date(Date.UTC(app.now.getUTCFullYear(), app.now.getUTCMonth() - 1, 1,  0,  0,  0))
   end:   new Date(Date.UTC(app.now.getUTCFullYear(), app.now.getUTCMonth(),     0, 23, 59, 59))
};
```

This will add a new date range called "Last Month" to the list of date ranges.

`SDCat.App.`**`items`**
(*Object*) - This stores all of the various data components required for the app. The content of the various controls on the app are populated from this object. Each "component" is an object with the following attributes:

sort
(*Boolean*) - Determines whether or not to sort these items into alphabetical order when displayed on the page.

clearable
(*Boolean*) - Determines whether or not to clear the selected items when the "Clear Selection" button is pressed.

list
(*Object*) - List of items that will populate the appropriate control. Each attribute of this object will be another object containing component specific information. The name of the attribute is the text that will appear in the control on the app. See below for the component specific information.

filterFunction
(*Function*) - User specified function to return a filtered list of the items. This should return an array of required item names.

changeButtonState
(*Boolean*) - Determines whether selected elements of this item can change the state of the **Generate Plot** button. If it is not defined, it is assumed to be `false`.

coverageAttributes
(*Object*) - Contains the information required to match up the items in the components `list` with the coverage objects that will be returned. In this object `source` is the attribute in the `list` object and `coverage` is the attribute in the coverage object.

help
(*Array*) - Array of bullet points to show in the help tooltip when the cursor hovers over the control's help button.

filtersRadars
(*Boolean*) - Determines whether these items can filter the radars. You should not need to change the value of this attribute.

default
(*String* or *Array*) - List of items that are to be automatically preselected when the sidemenu is created. If this attribute is a *String*, it will be converted to an *Array*. For all items, by default this attribute is undefined.

The next sections contained detailed information about the components which can be found in the `items` object:

`SDCat.App.items.`**radars**
Contains information about the radars. The `list` object contains objects with the following attributes:

selected
(*Boolean*) - Determines if this radar has been selected by the user. (Default: `false`)

latitude
(*Number*) - The latitude of this radar in WGS84.

longitude
(*Number*) - The longitude of this radar in WGS84.

name
(*String*) - The name of the radar.

institute
(*String*) - The name of the institute that maintains the radar.

country
(*String*) - The name of the country the radar is located in.

code
(*String*) - The radar's three letter identification code.

show
(*Boolean*) - Determines whether or not to show the radar on the app.

Radars can be referenced from the `list` attribute by using their three-letter ID code. At initialisation, `show`, `clearable` and `changeButtonState` are set to `true` and the default `filterFunction` returns all the radars found in the API. `coverageAttributes` uses `code` as the `source` attribute and `radar_code` as the `coverage` attribute.

An example of manipulating this object is as follows:

```
// Set the name of the hal radar to Halley VI
app.items.radars.list.hal.name = "Halley VI";

// Preselect this radar
app.items.radars.list.hal.selected = true;
```

This renames the `hal` radar and then programatically selects it.

`SDCat.App.items.`**`fileTypes`**
Contains information about the file types. The `list` object contains objects with the following attributes:

selected
(*Boolean*) - Determines if this file type has been selected by the user. (Default: `false`)

name
(*String*) - The name of this file type.

show
(*Boolean*) - Determines whether or not to show the file type on the app.

File types can be referenced from the `list` attribute by using their name. At initialisation, `show`, `clearable` and `changeButtonState` are set to `true` and the default `filterFunction` returns all the file types found in the API. `coverageAttributes` uses `name` as the `source` attribute and `file_type` as the `coverage` attribute.

An example of manipulating this object is as follows:

```
// Create an array of the different file types and sort them
var fileTypes = Object.keys(app.items.fileTypes.list).sort()
```

This sets a variable containing an array of the different fileTypes, sorted into alphabetical order.

`SDCat.App.items.`**`dateTypes`**
Contains information about the different date types. The different date types currently available are dates based on the file name (**File**) or dates based on the data within the file (**Data**). As a result, this object has been hardcoded, though you can override the information if you require. The definition of this object is as follows:

```
dateTypes: {
   sort: false,
   clearable: false,
   list: {
      file: { selected: true,  value: "File", show: true },
      data: { selected: false, value: "Data", show: true }
   },
   changeButtonState: true,
   coverageAttributes: { source: "value", coverage: "coverage_type" },
   help: [
      "Select the date type you are interested in",
      "File dates are dates inferred from the file names",
      "Data dates are dates calculated from the data within a file",
      "Only one date type can be selected"
   ]
```

`SDCat.App.items.`**`institutes`**

Contains information about the institutes. The `list` object contains objects with the following attributes:

`selected`

(*Boolean*) - Determines if this institute has been selected by the user. (Default: `false`)

Institutes can be referenced from the `list` attribute by using their name. At initialisation, `show` and `filtersRadars` are set to `true` and `clearable` is set to `false`.

`SDCat.App.items.`**`countries`**

Contains information about the countries. The `list` object contains objects with the following attributes:

`selected`

(*Boolean*) - Determines if this country has been selected by the user. (Default: `false`)

Countries can be referenced from the `list` attribute by using their name. At initialisation, `show` and `filtersRadars` are set to `true` and `clearable` is set to `false`.

Note: The `countries` list only contains the countries that contain radars. The countries of institutes are not included

`SDCat.App.items.`**`dateRanges`**

Contains information about the date ranges. The `list` object contains the *preSelectedDateRanges* and can be referenced by using their name. At initialisation, `show` and `clearable` are set to `false`.

`SDCat.App.items.`**`radarNameOptions`**

Contains information about the radar name options. The information in this item is used to create a control which will allow the user to display the radars by either their identification code or their name. As a result, this object has been hardcoded, though you can override the information if you require. The definition of this object is as follows:

```
radarNameOptions: {
   sort: false,
   clearable: false,
   list: {
      code: { selected: true,  value: "Radar Code", show: true },
      name: { selected: false, value: "Radar Name", show: true }
   },
   help: [
      "Use these options to switch between displaying radars as their 3 letter code or by th
   ]
};
```

`SDCat.App.`**`sidemenuButtons`**

(*Array*) - This consists of a list of single action buttons that will appear in the sidemenu. Each element of this array is in another array and it is that inner array that contains the button information. The outer arrays are there to allow the buttons to be grouped together with gaps placed between the groups.

The buttons that are created via this array are the **Select All**, **Clear Selection**, **Show/Hide Help**, **Reset View** and **Generate Plot** buttons. By default, **Select All**, **Clear Selection**, **Show/Hide Help** and **Reset View** are in one group and **Generate Plot** is in another group, though you can rearrange this or add new buttons if required. The button objects contain the following attributes:

`id`
(*String*) - The ID for the HTML element of this button.

`title`
(*String*) - The text that will appear on this button.

`action`
(*String*) - Name of the SDCat.App function that will be run when this button is clicked.

`show`
(*Boolean*) - Determines whether or not to display this button on the page. The default value for this is the value of the appropriate option in the SDCat.App constructor.

`enable`
(*Boolean*) - Determines whether or not to enable this button.

`help`
(*String*) - Text that appears when help is to be displayed and the cursor hovers over the button.

`SDCat.App.`**`dateRanges`**
(*Object*) - Contains information about the date ranges of different parts of the app:

`plot`
(*Object*) - Contains the date range that the plot will cover:

`start`
(*Date*) - Start date of the plot. (Default: 00:00:00 on January 01 of the current year)

`end`
(*Date*) - End date of the plot. (Default: 23:59:59 on December 31 of the current year)

`coverage`
(*Object*) - Contains the date range over which coverage is available

`start`
(*Date*) - Earliest date of available coverage. (Default: null)

`end`
(*Date*) - Latest date of available coverage. This usually gets set to 23:59:59 on December 31 of the last year of the coverage. (Default: null)

`help`
(*Array*) - Array of bullet points to show in the help tooltip when the cursor hovers over the control's help button.

`SDCat.App.`**`containerDiv`**
(*JQuery DIV element*) - This is the div that the app will be inserted into. The ID of this div is specified in `SDCat.options.containerDivId`.

`SDCat.App.`**`helpBox`**
(*Object*) - Contains information about the tooltip that appears when the cursor hovers over the help button. Contains the following attributes:

`div`
(*JQuery DIV element*) - The div of the help box itself. The ID of this element is `helpbox`.

`show`
(*Boolean*) - Determines whether or not to show the help box on the app. (Default: `true`)

`SDCat.App.`**`sidemenu`**

(*Object*) - Contains information about the app's sidemenu. Contains the following attributes:

`div`

(*JQuery DIV element*) - The div of the sidemenu itself. The ID of this element is `sidemenu`.

`buttons`

(*Object*) - Collection of buttons to add to the sidemenu. These are added programatically. (Default: `{}`)

`SDCat.App.`**`mainBody`**

(*Object*) - Contains information about the main body of the app (i.e. where the plot appears). Contains the following attributes:

`div`

(*JQuery DIV element*) - The div of the main body itself. The ID of this element is `main`.

4.1.4 Functions

`SDCat.App.`**`display`**`()`

Populates and displays the app. In most cases, this is the only function you will ever need to run.

`SDCat.App.`**`displaySidemenu`**`()`

Adds the sidemenu to the app. This function will create the div and then add the required controls to the sidemenu, based on the options that were set when the App object was initialised.

`SDCat.App.`**`createSectionHeader`**`(`*header*, *help*`)`

Creates, populates and adds a header to a control in the app's sidemenu.

`header`

(*String*) - The text to appear in the section header.

`help`

(*String* or *Array*) - The text to appear when the cursor is over the help icon.

`SDCat.App.`**`addButtonsToSidemenu`**`()`

Loops through the *SDCat.App.sidemenuButtons* and adds them to the sidemenu.

`SDCat.App.`**`createSelectBox`**`(`*options*`)`

Creates a select box comprised of the options in a given *SDCat.App.items* list.

`options`

(*Object*) - Contains the select box specific options. The attributes of this object are as follows:

`header`

(*String*) - The text of this select box's section header. This will be passed to *SDCat.App.createSectionHeader()*.

`type`

(*String*) - The type of *items* which will populate this select box.

`SDCat.App.`**`setDefaultOptions`**`(`*box*`)`

Preselects default options in a select box, based on the contents of the item's *default* attribute.

`box`

(*String*) - The type of *items*. This will be the same as `options.type` in *SDCat.App.createSelectBox()*.

`SDCat.App.`**`refreshSelectBox`**`(`*box*`)`

Refreshes the content of a select box, based on the contents of the item's *default* attribute.

`box`

(*String*) - The type of *items*. This will be the same as `options.type` in *SDCat.App.createSelectBox()*.

SDCat.App.**createButtonSet**(*options*)

Creates a JQuery buttonset comprised of the options in a given *SDCat.App.items* list.

options

(*Object*) - Contains the buttonset specific options. The attributes of this object are as follows:

header

(*String*) - The text of this buttonset's section header. This will be passed to *SDCat.App.createSectionHeader()*.

type

(*String*) - The type of *items* which will populate this buttonset.

onChange

(*Function*) - The function to run when the buttonset's value has been changed.

SDCat.App.**createButton**(*options*)

Creates a JQuery button.

options

(*Object*) - Contains the button specific options. The attributes of this object are as follows:

title

(*String*) - The text that will appear on the button.

SDCat.App.**createDropDownBox**(*options*)

Creates a JQuery drop down box comprised of the options in a given *SDCat.App.items* list.

options

(*Object*) - Contains the drop down box specific options. The attributes of this object are as follows:

id

(*String*) - The ID of this drop down box.

type

(*String*) - The type of *items* which will populate this drop down box.

headerText

(*String*) - A description of this drop down box. This will appear on page load as the default text that appears in the drop down box.

onChange

(*Function*) - The function to run when the buttonset's value has been changed.

SDCat.App.**selectItemsByAttribute**(*itemType*, *attribute*, *attributeValue*)

Returns an array item names from a *SDCat.App.items* list where an attribute matches a value.

itemType

(*String*) - The type of *items* to search through.

attribute

(*String*) - The attribute of the *item* to search against.

attributeValue

(*String*) - The value of the attribute to search for.

Example:

```
// Find all BAS radars
var filteredItems = app.selectItemsByAttribute("radars", "institute", "British Antarctic Survey"
```

This will return an array containing the radars where the institute attribute matches **British Antarctic Survey** (i.e. **fir** and **hal**).

SDCat.App.**selectAll**()

Selects all items in the app. Note that the only items selected are those where the `clearable` flag is set to `true`.

SDCat.App.**clearSelection**()

Clears all selected items and resets the selection boxes. Note that the only items selected are those where the `clearable` flag is set to `false`.

SDCat.App.**addHelp**(*element*, *helpText*)

Adds a help tooltip to an element in the page.

element

(*JQuery element*) - When the cursor hovers over this element, the help tooltip will be displayed.

helpText

(*String* or *Array*) - Collection of bullet points that will appear in the tooltip which make up the help text. If a *String* is supplied, it will be converted to an *Array*. This is usually the `help` attribute of the app components.

SDCat.App.**generatePlot**()

Generates the coverage plot by creating a **SDCat.CoveragePlot** object and displaying it on the page.

SDCat.App.**resetView**()

Resets the view of the plot to the original timespan after zooming in.

4.2 SDCat.CoveragePlot

This is an object used to create an instance of a plot which shows the coverage of available SuperDARN data. This is embedded within the app, but using the constructor, you can add a standalone coverage plot independent of the app. This object is, in effect, a particular implementation of the generic **CataloguePlot** library, designed to look like the static plots available on the VT website.

4.2.1 Prerequisites

The following function libraries are required in order to use this object:

- `/js/sdcat/SDCat.js`
- `/js/catalogue-plot/CataloguePlot.js`

4.2.2 Constructor

class SDCat.**CoveragePlot**(*options*)

options

(*Object*) - List of options. These options set attributes of the object:

sdcat

(*Object*) - Object containing configuration parameters for the SDCat object. Here, you need to specify the URL to the SDCat API instance you wish to query.

Example:

```
var options = {
   sdcat: {
      url: "http://localhost:8082/api/v2/"
   },
};
```

plotId
(*String*) - ID of the div that will contain the plot. If this is not specified, the plot will be appended to the main body of the page. (Default: `null`).

radars
(*Array*) - Array of 3-letter radar codes which belong to the radars that are to be shown on the plot. (Default: `empty`).

fileTypes
(*Array*) - Array of file types that are to be shown on the plot. (Default: `empty`).

coverageType
(*String*) - The coverage type to use on the plot. This is the source of the date information i.e. **File** for dates based on the date information in the file name or **Date** for dates based on the date information in the data file itself.

startDatetime
(*String*) - ISO formatted date/time string representing the start point of the plot (e.g. **2017-01-01T00:00:00**).

endDatetime
(*String*) - ISO formatted date/time string representing the end point of the plot (e.g. **2017-12-31T23:59:59**).

4.2.3 Attributes

`SDCat.CoveragePlot.`**sdcat**
(*SDCat*) - This is the SDCat object which will be used to query the API. It will be created automatically using the parameters specified in the SDCat.CoveragePlot constructor.

`SDCat.CoveragePlot.`**errorMessages**
(*Object*) - An object containing the error emssages to be displayed when certain errors are thrown. The following messages are defined:

NotInRange
(*String*) - Message to be displayed when no data could be found within the desired time period. (Default: **No data could be found for the over the time range you selected. Please refine your selection and generate the plot again**).

BadRange
(*String*) - Message to be displayed when the end date is before the start date. (Default: **The selected end date is before the selected start date. Please refine your selection and generate the plot again**).

`SDCat.CoveragePlot.`**catPlot**
(*CataloguePlot*) - The underlying **CataloguePlot** object that forms the SuperDARN coverage plot.

4.2.4 Functions

`SDCat.CoveragePlot.`**display**`()`
Populates and displays the plot. In most cases, this is the only function you will ever need to run.

`SDCat.CoveragePlot.`**`convertToCataloguePlotData`**`(`*temporal*`)`

Converts the coverage objects returned from the SDCat API into the format required by the CataloguePlot.

`temporal`

(*Array*) - Array of coverage objects to convert to CataloguePlot format.

`SDCat.CoveragePlot.`**`createLegend`**`(`*plot*`)`

Returns the HTML code to display the legend that will appear above the plot. This is used as the **contentFunction** for the **CoveragePlot** object's legend.

`plot`

(*CoveragePlot*) - The underlying **CoveragePlot** object.

`SDCat.CoveragePlot.`**`jsTimestampTo2HourBlock`**`(`*timestamp*`)`

Converts a Javascript timestamp to the nearest SuperDARN 2-hour block.

`timestamp`

(*Number*) - Number of milliseconds since the UNIX epoch.

INDEX