
ska-javascript-components

Documentation

Release 1.0.9

SKAO , (Trevor A Swain)

Mar 22, 2024

CONTENTS:

1	Usage	3
2	Functionality	5
3	Colors	7
4	Help	9
5	Logo	11
6	Spacer	13
7	Status	15
8	Styling	17
9	Telescope	19
10	Theme	21
11	Version	23

This is a JavaScript library contained low level components and utilities for use within SKAO GUI applications

Every effort has been made to ensure that all components have a unique means of identification for testing purposes, as well as implementation of standard properties to allow for maximum accessibility for those that have access limitations

Configuration Addition

This repository is available from the Central Artefact Repository (CAR), so the following needs to be executed from the repository that needs to use this library

- *npm config set @ska-telescope:registry https://artefact.skao.int/repository/npm-internal/*
-

Update the CI/CD processor

As the library is in an SKAO location, the .gitlab-ci.yml should be amended to the following so that the Ci/CD build will complete

- *npm config set @ska-telescope:registry https://artefact.skao.int/repository/npm-internal/*
 - *yarn build* (npm can be used here as an alternative)
-

Required addition to package.json

As these are designed for use with React applications, the following needs to be run so that the library is added to the package.json

- *yarn add @ska-telescope/ska-javascript-components* (npm can be used as an alternative)
-

Addition into the JS/TS code

Implementation of the specific components vary and are detailed in later documentation, but the following format will need to be added at the top of each file where a component is to be used

- *import { ??????? } from "@ska-telescope/ska-javascript-components";*
-

FUNCTIONALITY

There follows a list of the available functionality, together with sample usage and the associated properties.

COLORS

Overview

The standard colors in use within the SKAO are provided here. It is expected that these will be used via the Theme, but have been provided for completeness

Example

```
import { Colors } from '@ska-telescope/ska-javascript-components'; ... Colors().STATUS_BG_0
```

Properties

None

Constants

None

Testing Identifier

Testing of styling will be done on the component if it is required

Overview

A standard help type for use within the SKAO are provided here.

Example

```
import { Help } from '@ska-telescope/ska-javascript-components';
```

Properties

header

“Property”, “Type”, “Required”, “default”, “”

“content”, “object”, “No”, “null”, “Intended use is that help that is relevant to the page/group is placed in here” “component”, “object”, “No”, “null”, “Intended use is that help that is relevant to a simple component is placed in here” “showHelp”, “boolean”, “No”, “false”, “display of the help is indicated by the status of this field”

Constants

None

Testing Identifier

Not applicable

LOGO

Overview

This is the SKAO logo. It is unlikely that this will be needed directly as it is currently used as part of the standard header from within the *ska-gui-components* library



Example : Mode determined by the provided function isDark()

```
import { Logo, LOGO_DEFAULT_HEIGHT } from '@ska-telescope/ska-javascript-components';  
<Logo dark={isDark()}/>
```

Table 1: Properties

Prop-erty	Type	Re-quired	default	
dark	boolean	No	false	Used to present the logo in the appropriate coloring for the mode
height	num-ber	No	LOGO_DEFAULT_HEIG	Specifies the height of the logo

Constants

- LOGO_DEFAULT_HEIGHT : Provided so can be used to standardize icons as needed (Value = 30)
-

Testing Identifier

Fixed with the value *skaoLogo*

SPACER

Overview

Component that allows for spacing to be added

Image not applicable

Example

```
import { SPACER_HORIZONTAL, SPACER_VERTICAL, Spacer } from '@ska-telescope/ska-javascript-components';  
<Spacer />
```

Table 1: Properties

Prop-erty	Type	Re-quired	default	
size	num-ber	No	90	Defines the size of the space in pixels
axis	string	No	SPACER_HORIZONTAL	SPACER_HORIZONTAL or SPACER_VERTICAL permitted

Constants

- SPACER_HORIZONTAL
 - SPACER_VERTICAL
-

Testing Identifier

Currently there is no requirement to specifically test this component

STATUS

Overview

Provides one of a number of coloured shapes that can be used to help distinguish the various status levels. These have been taken from the engineering standards, and additional logos were added.



Example

```
import { Status } from '@ska-telescope/ska-javascript-components';  
<Status testId="statusTestId" />
```

Table 1: Properties

Prop-erty	Type	Re-quired	de-fault	
level	num-ber	No	9	Valid range is 0 - 5. Anything outside that range is empty. Success = 0.
size	num-ber	No	12	This provides both height & width
testId	string	Yes		Unique identifier used for testing
text	string	No		String that will be displayed upon the shape. Single character only recommended

Constants

None

Testing Identifier

Provided by the property *testId*

STYLING*Overview*

The standard styling in use within the SKAO are provided here. It is expected that these will be used via the Theme, but have been provided for completeness

Example

```
import { Styling } from '@ska-telescope/ska-javascript-components';  
Styling().BORDER_RADIUS
```

Properties

None

Constants

None

Testing Identifier

Testing of styling will be done on the component if it is required

TELESCOPE

Overview

Standard telescope type and objects for use within any application.

Example

```
import { Telescope, TelescopeList, TELESCOPE_LOW, TELESCOPE_MID } from '@ska-telescope/ska-javascript-components';
```

Note that the Telescope object will be NULL or either of the provided constants

Properties**header**

“Property”, “Type”, “”

“code”, “string”, “Code for the telescope” “name”, “string”, “Display name for the telescope” “location”, “string”, “Name of the primary location of the telescope” “position”, “object”, “Made up of the latitude and longitude of the telescope” “image”, “string”, “URL to a stock image of the telescope”

Constants

TelescopeList, TELESCOPE_LOW, TELESCOPE_MID

Testing Identifier

Not applicable

Overview

This is the latest SKAO Theme for use within all SKAO applications. Also provided are standard constants that can be used for toggling between light and dark themes.

Code snippet *./services/theme.tsx*

```
import { createTheme } from '@mui/material'; import { Theme } from '@ska-telescope/ska-javascript-components';
const theme = mode => createTheme(Theme(mode)); export default theme;
```

Code snippet *App.tsx* import theme from *'./services/theme/theme'*;

```
function App() {
  const [apiVersion, setAPIVersion] = React.useState("LOCAL"); const { t } = useTranslation('dpc');
  const skao = t('toolTip.button.skao'); const mode = t('toolTip.button.mode'); const toolTip = { skao: skao, mode:
  mode };
  // Theme related const [themeMode, setThemeMode] = React.useState(THEME_LIGHT);
  const themeToggle = () => { setThemeMode(themeMode === THEME_LIGHT ? THEME_DARK :
  THEME_LIGHT);
};
  return (
    <ThemeProvider theme={theme(themeMode)}> <CssBaseline enableColorScheme />
      ...
    </ThemeProvider>
  );
}
export default App;
```

Testing Identifier

Testing of the theme will be done at component and page level as required

Overview

This is a simple constant that provides the latest version of the library

Code snippet `./services/theme.tsx`

```
import { createTheme } from '@mui/material';  
import { JAVASCRIPT_COMPONENTS_VERSION } from '@ska-telescope/ska-javascript-components';
```

Testing Identifier

n/a
