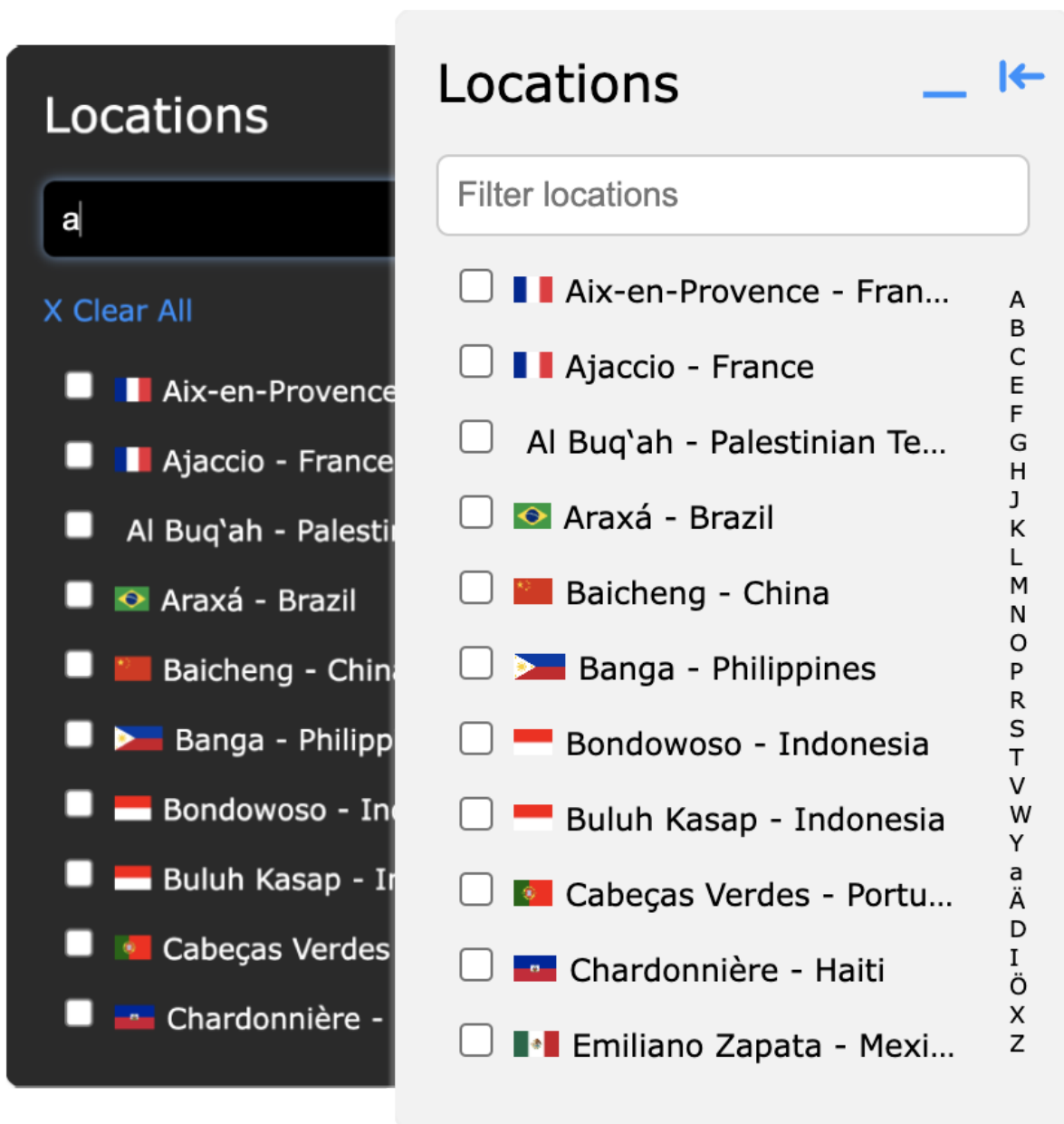


WIDGET COMPONENT

Mohamed Sharbudeen
iamsharbu@gmail.com

Overview

A data listing component with search filtering and indexing for quick navigation. Widget customisations include themes such as light, dark themes and component sizing categorized as small, medium and large. Code can be extended in future to include multiple components such as facet, calendar, weather and much more.



Approach

- Going with React JS (giving it a try) to build UI components. No third-party libraries.
- Modularized code at the correct level.
- Given scope for future extensibility.
- Mock input data is pre-processed for sanity.
- Constants file is maintained.

Assumptions

- For the current initial project, I will consider that checkbox has no functionality, selecting or deselecting state is not maintained. I.e. Having a submit button to send a request with locations selected.
- Iterative approach for searching the data.

Future Enhancements

- For larger datasets, go for a filtering and caching algorithm to search and filter efficiently.
- Lazy load data on scrolling. Fetch data in batches and maintain lightweight DOM.
- i18n - Translate the input data for user locale language. I.e. "Location" translates to "位置" for Japanese locale.
- Tooltips, accessibility options/ UI design for visually impaired.
- Mobile based responsive design.
- Automation cases to test UI functionalities.
- Additional widget components like calendar, date-picker, combo-box and much more.

Execution

- Download the attachment Project.zip and unzip it.
- Go to the folder Project and in command line run "npm install"
- Once dependencies are installed, run "npm start". Refer readme.txt for full details.

Code Details

- Under Project/src/App.js, properties of the widget is specified

```
const widgetProps = {  
  title : WIDGET_LOCATIONS,  
  type : WIDGET_TYPE_SEARCH,  
  data : preProcessData(jsonData),  
  theme : {  
    type : WIDGET_THEME_TYPE.DARK,  
    size : WIDGET_SIZE.MEDIUM,  
  }  
}
```

- type : SEARCH
- theme.type : DARK | LIGHT
- Theme.size : SMALL | MEDIUM | LARGE

Manual test cases

- Initial data listing and filtered data should always be sorted.
- On each keystroke, data is filtered. Clear search.
- Indexes navigate to the starting location with that letter.
- Maximize, Minimize, Expand and Collapse functionality should be tested.

Dependencies

Production build:

- React
- React-Dom