

Intelligent tree data management component

Jakub Lečbych

Content

1. Goal
2. Introduction to the problem
3. API research
4. Technology
5. Component structure design
6. Component life cycle (data flow)
7. Pros and cons
8. Competitive solutions
9. Demo



Goal

- Create UI component for selecting/ multi-selecting options
- UI component will be represented as input field with dropdown menu

Create new option

Hide filter

- ☒ Multiselect
- ☐ Display Term State
- ☒ Show info on hover
- ☒ Render as tree
- ☒ Expanded

Select...

- 1000000 - Equipment
 - 1010000 - 0100 Aircraft Performance
 - 1010100 - 0101 Degraded performance
 - 99012007 - Degraded Performance - Fixed Wing Aer...
 - 99012008 - Degraded Performance - Glider
 - 99012009 - Degraded Performance - Helicopter
 - 99012010 - Degraded Performance - RPAS
 - 99012011 - Degraded Performance - Other

No existence of a component supporting following

- Multiple option providers
 - Downloading data from different providers simultaneously
- Linked data
- Flexible
- Maintain same functionality as current solutions
 - Multi-select, show options as tree, collapsible tree branches, etc.

Additional functionality of the new component

- Display some info such as description, status, etc.
- Creating new options
- Configuration in UI (e.g. selectin type)

Introduction to the problem

Based on several definitions

- Type – **web**, system, software, hardware, ...
- Data format – file, JSON, CSV, XML, ...
- Returned informations – complete data, data headers (IDs), ...
- Returned structure – list, tree, graph, ...

URL structure

- Protocol + 'sub domain' + Domain + 'version' + endpoint + 'query'
- Example:

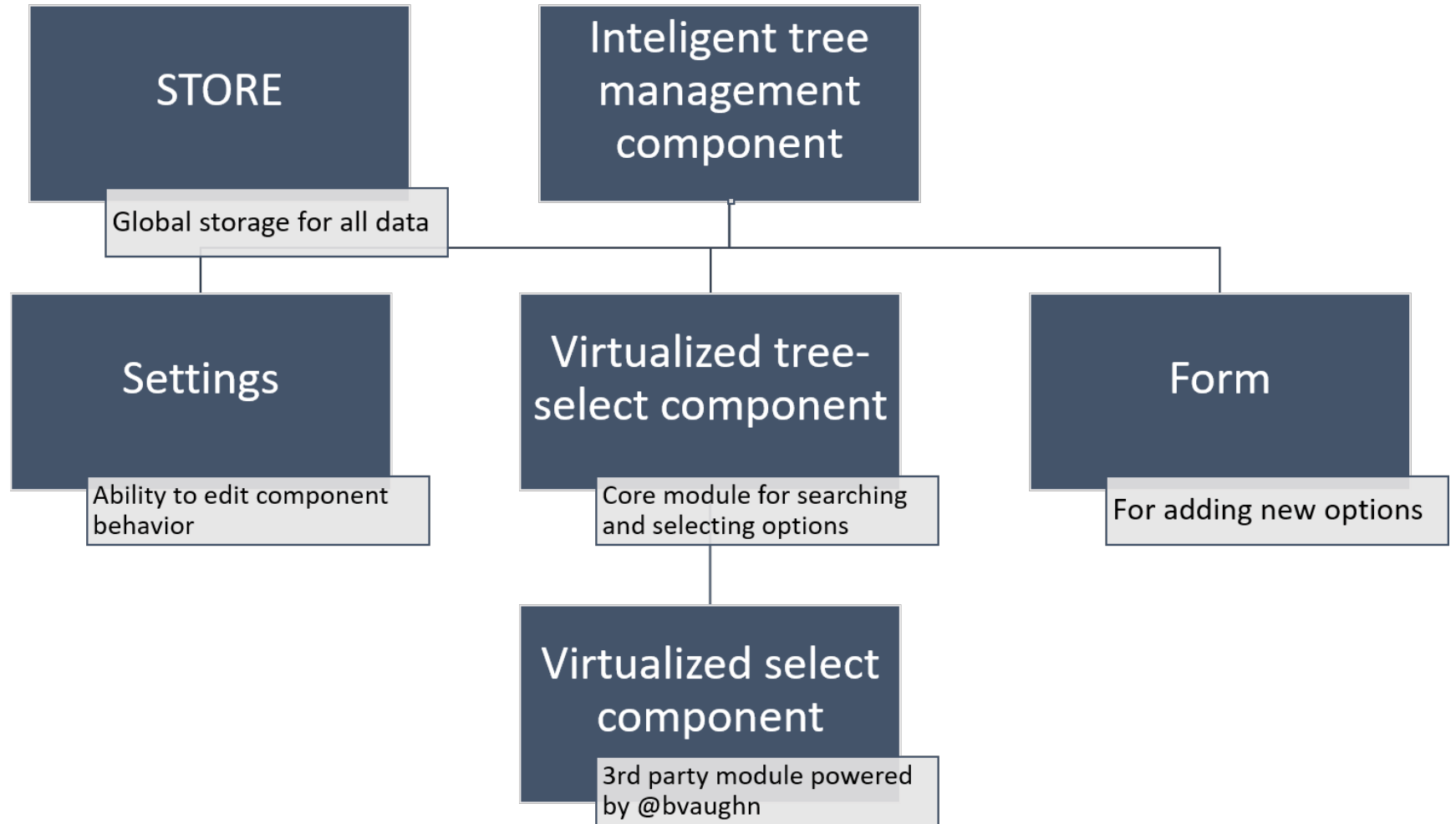
<https://www.api.spotify.com/v1/artist/123456789/albums?location=CZ,SK&limit=25&offset=5>

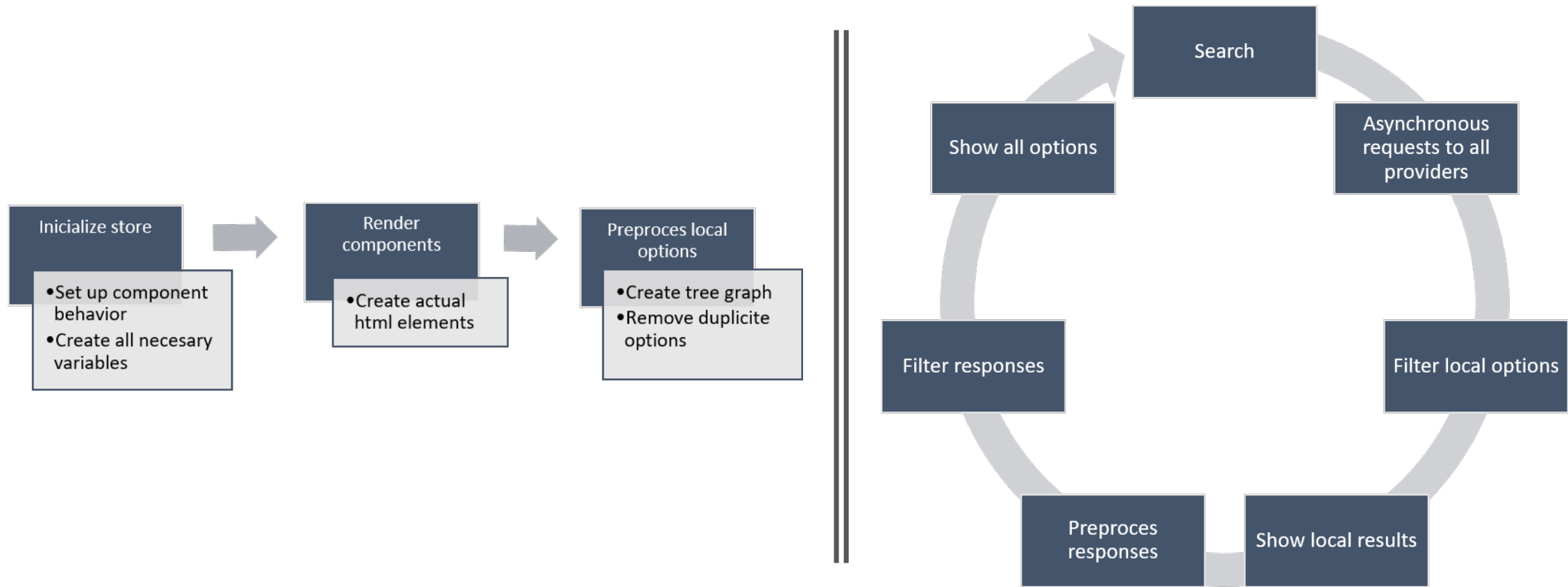
API research



Technology

Component structure





Component flow + life cycle

Pros and Cons

- Pros

- Support multiple option providers
- Work with linked data
- Huge customization potencial
- Easily integable with other react application
- Does not affect UI performance (rendering)

- Cons

- Processing data may be slow
 - Merging duplicates and creating tree graph can took some time for large datasets (1000+ items)
- By default support only JSON data format (? - CSV, XML)

Competitive solutions

RC-tree

- Pros
 - Draggable options
- Cons
 - Unable to customize render method
 - Render all options at once
 - Removing selected only via checkbox
 - Support only specific option format
 - `{ label: 'node1', children: [{ label: 'node1-0', children: []}, ...]}`

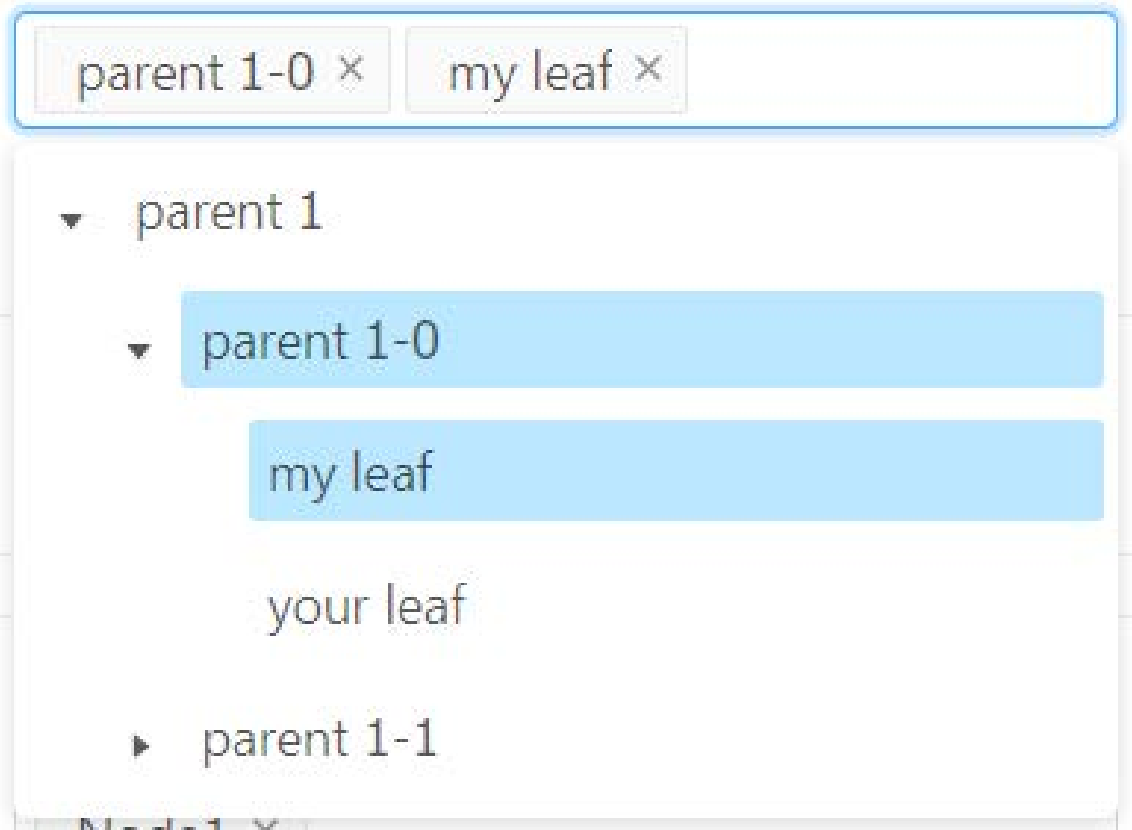
<http://react-component.github.io/tree/>



Ant.design tree-select

- Pros
 - Modern design
- Cons
 - Each option take too much space
 - Unable to customize render method
 - Render all options at once

<https://ant.design/components/tree-select/>





Demo

Thanks you for your attention

Questions?