Functional requirements related to the users are following:

1. Component must support multiple options providers
2. Dropdown with results will be displayed on focus
3. Each option should show its state (additional info- is selected, provider, comment, etc.)
4. User must be able to filter among these options
5. Selection and multi-selection must be provided
6. Component must provide a way to create new options
7. Tree structure

Other functional requirements for developers:

1. Component must be able to work with linked data sources
2. Component should support multiple input formats – JSON, CSV

Non-functional requirements for users:

1. Operations like search and render can’t slow/freeze UI
2. Processing data, filtering and rendering should be real-time (not take longer than acceptable)

Other non-functional requirements for developers:

1. Component must be easily integrable
2. Component must be flexible – custom styling, own filter method and render method

Similar solutions

TODO detailed comparison

<https://github.com/react-component/tree-select>

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

https://github.com/dowjones/react-dropdown-tree-select

## API research

Firstly, lets quickly introduce you what API is. API stands for Application Programming Interface. It is a program layer that is responsible for interacting with users, giving them responses based on their request. We will be focusing only on Web API. The simplest way to describe it is that Web API is set of dedicated web URLs, somewhere on the internet, that return some kind of response (usually in text format) to the requestor. There are several types of APIs. From historical SOAP (Simple Object Access Protocol) and SOA to more modern REST (Representational State Transfer).

Because REST API is mostly used today, let's focus only on this one type. I will be using Spotify API as an example, because of their excellent documentation and variety of their endpoints. All APIs have basically three parts. As you can see below, first is their root address, in this case, it is [*https://api.spotify.com*](https://api.spotify.com)then following a version, but this is an optical part. Next is an endpoint */artists/{id}/albums* notice an {id} parameter in the URL, this is one way how you can send some data to the API. And last part is *query parameter*. Query parameters are at the end of the URL behind question mark and contain key=value pairs connected with an ampersand. Usually, all query parameters are optional because they have a default value specified on the API side.

https://api.spotify.com/v1/artists/1vCWHaC5f2uS3yhpwWbIA6/albums?market=ES&album\_type=single&limit=2

Each request must also have a header part, where are specified some other information. E.g. Accepted-language or encoding, but most important are Referer and Host.

Last part of each request is body part. This is the second way how you can send some data to the API endpoint. But the format of the body part must be in a format that is accepted by the server side, e.g. in JSON, XML or text format.

In conclusion, there is a lot of different Web API types. You can distinguish them based on their response format (JSON, XML, img, CSV, etc.), type of their response – some supports filtering results on their side, some return only limited amount of data, while another may return only header information (e.g. IDs) so you must make another request for each ID to get detailed information.