Test Task: Navigation Tree

Task Description:

In this test task you will be creating a “Navigation Tree” based on the JSON file that is attached **using React**.

If you already have experience with **TypeScript** experience you can try to create it in Typescript

The Json file attached contains information about each “Node” or “Leaf” on the “Navigation Tree”.

Using this Json you will populate the tree by choosing your library form any of the React Libraries that are listed on the page: https://react.rocks/tag/TreeView

Task Description: **Maximum 3 Days**

Sending Task Results:

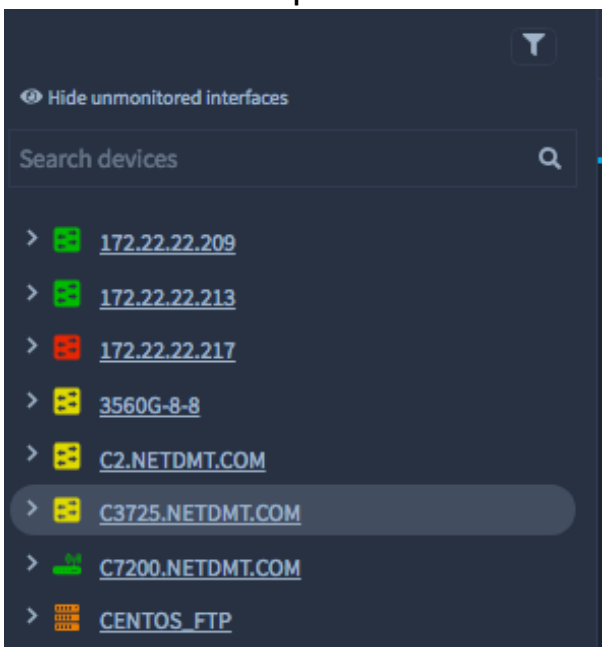
Once you have completed the task you must deploy it as a running application on either **Github Pages** or **Heroku** as your preference and send a link for both the Code and the **Running React App**.

KEY Requirements of the Task

1. POPULATING THE TREE:

The tree is populated Using the **“name”** Field of the Json Object ("name": "C2.NETDMT.COM",)

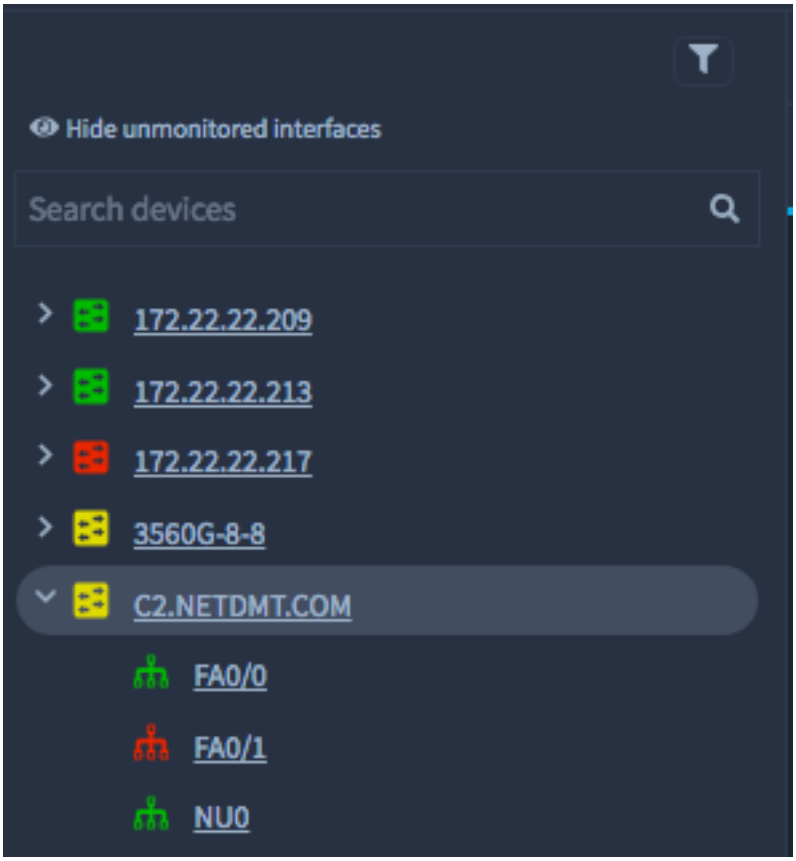
Illustrative Example:



2. EXPANDING CHILD NODES:

Upon Clicking any specific Leaf in the tree, the tree should expand and display the underlying children nodes which can be derived from the **nodes** object in the parent.

For example on clicking **C2.NETDMT.COM** the child nodes should expand and display **FA0/0, FA0/1** and **NU0** as illustrated.



Each underlying child node (**FA0/0, FA0/1** and **NU0**) is colored based on the field state in the object ("state": "up/down").

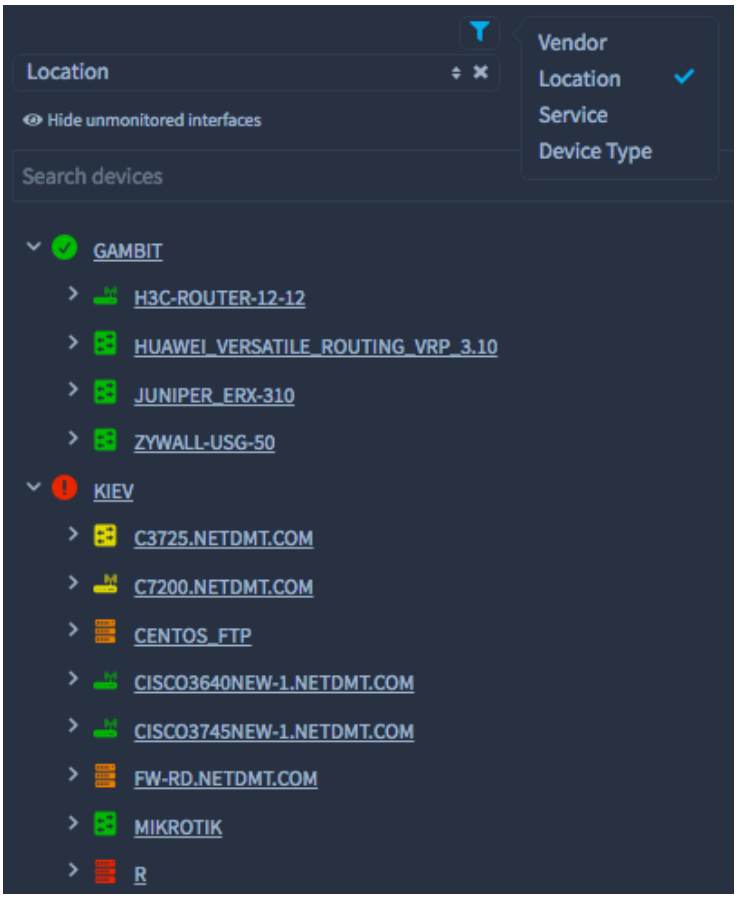
IF: ("state": "up"). It should be colored "GREEN"
IF: ("state": "down"). It should be colored "RED"

IMPORTANT NOTE:
The child nodes should be fetched **"On Click"** on a Parent Node i.e. only when C2.NETDMT.COM is clicked the Children should be fetched from the Json File (**FA0/0, FA0/1** and **NU0)**

3. FILTERING THE TREE:

There should be a **”Filter functionality”** to render the tree by using by using the following filters and embedded information in the parent node object. Once the Filter is Selected the Tree should refresh and repopulate the nodes depending on the filter selected.

VENDOR : "vendor": "ciscoSystems",
LOCATION : "location": "Unknown Location",
DEVICE TYPE : "deviceType": "switch",
SERVICE "service": "WAN",



JSON Reference:


```
{
  "id": "C2.NETDMT.COM_172.22.22.126",
  "state": "minor",
  "type": "switch",
  "name": "C2.NETDMT.COM",
  "vendor": "ciscoSystems",
  "location": "Unknown Location",
  "deviceType": "switch",
  "service": "WAN",
  "isMonitored": true,
  "nodes": [
    {
      "id": "C2.NETDMT.COM_172.22.22.126.if.4",
      "ifIndex": 4,
      "type": "interface",
      "isMonitored": true,
      "name": "Nu0",
      "state": "up"
    },
    {
      "id": "C2.NETDMT.COM_172.22.22.126.if.2",
      "ifIndex": 2,
      "type": "interface",
      "isMonitored": false,
      "name": "Fa0/1",
      "state": "down"
    },
    {
      "id": "C2.NETDMT.COM_172.22.22.126.if.1",
      "ifIndex": 1,
      "type": "interface",
      "isMonitored": true,
      "name": "Fa0/0",
      "state": "up"
    }
  ]
}
```