

## State

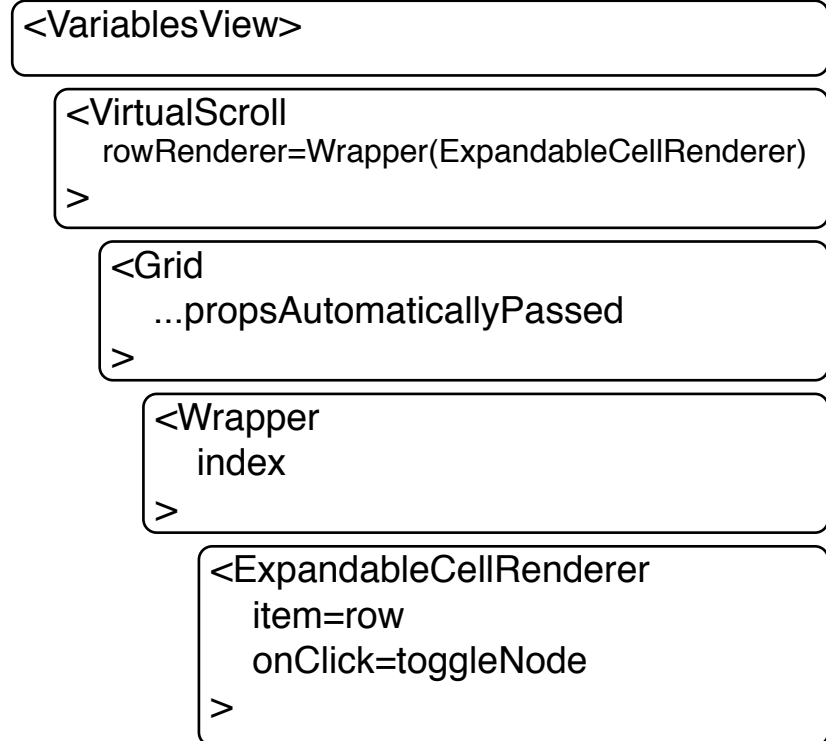
```
{
  "data": {
    "variables": {
      "obj": {
        "1a": {
          "str": "obj is not expanded"
        },
        "1b": "",
        "1c": {
          "str2": "obj is expanded"
        }
      }
    },
    "uiState": {
      "expandedNodes": [
        "obj",
        "obj/1c"
      ]
    }
  }
}
```

## Variables object is flattened into rows (maybe using reselect?)

```
[
  {
    "value": obj,
    "id": "obj",
    "parentId": null
  },
  {
    "value": 1a,
    "id": "obj/1a",
    "parentId": "obj"
  },
  {
    "value": str,
    "id": "obj/1a/str",
    "parentId": "obj/1a"
  },
  {
    "value": 1b,
    "id": "obj/1b",
    "parentId": "obj"
  },
  {
    "value": 1c,
    "id": "obj/1c",
    "parentId": "obj"
  },
  {
    "value": str2,
    "id": "obj/1c/str2",
    "parentId": "obj/1c"
  }
]
```

## Hidden rows are filtered and properties are derived

```
[
  {
    "value": obj,
    "id": "obj",
    "parentId": null,
    "expandable": true,
    "expanded": true,
    "depth": 0
  },
  {
    "value": 1a,
    "id": "obj/1a",
    "parentId": "obj",
    "expandable": true,
    "expanded": false,
    "depth": 1
  },
  {
    "value": 1b,
    "id": "obj/1b",
    "parentId": "obj",
    "expandable": false,
    "depth": 1
  },
  {
    "value": 1c,
    "id": "obj/1c",
    "parentId": "obj",
    "expandable": true,
    "expanded": true,
    "depth": 1
  },
  {
    "value": str2,
    "id": "obj/1c/str2",
    "parentId": "obj/1c",
    "expandable": false,
    "depth": 2
  }
]
```



ExpandableCellRenderer receives each object. It uses the depth to calculate the indentation.

ExpandableCellRenderer also receives an onClick function. It uses the expandable/expanded property to determine whether to show the twisty. When clicked, it uses an action with the payload {expanded, id}. It updates expandedNodes, which triggers recalculation of the state.