01/17/2017

Contents

1 Create a new Component

React Documentation https://facebook.github.io/react/docs/react-component.html

Go to imports > components

* Create a new file: YourComponent.js

Listing 1: Add new Route in order to show the Component

```
1  import React from 'react';
2  import {
3    Row,
4    Col
5  } from '@sketchpixy/rubix';
6
7  export default class YourComponent extends React.Component {
8    constructor(props) {
9      super(props);
10     this.state = {
11       values : [0,1,2,3];
12     };
13   }
14
15   render() {
16
17     let labels = ["World", ...];
18
19     return (
20       <div>
21         <Row>
22           <Col sm={12}>
23             <div>
24             <h1>Hello {labels[0]} {this.state.values[1]}</h1>
25             </div>
26           </Col>
27         </Row>
28       </div>
29     );
30   }
31 }
```

2 Add a new Tab on Sidebar

Github Documentation: https://github.com/ReactTraining/react-router

Go to imports > routes.js

Listing 2: Add new Route in order to show the Component

```
1
2
3  import YourComponent from './path/to/YourComponent';
4
5  ...
6  const routes = (
7    <Route path='/Dashboard' component={App}>
8      <IndexRoute component={Home} />
9      ...
10     <Route path='/newPath' component={YourComponent} />
11     ...
12
13   </Route>
14 );
15
16
17 ...
```

Go to imports > common > sidebar.js

Listing 3: Add the new path to the Sidebar

```
1
2
3  class ApplicationSidebar extends React.Component {
4    ...
5    render() {
6      return (
7        <div>
8          <Grid>
9            <Row>
10             <Col xs={12}>
11               <div className='sidebar-nav-container'>
12                 <SidebarNav style={{marginBottom: 0}} ref={(c) => this.↩
                       _nav = c}>
13
14                   { /** Pages Section */ }
15                   <div className='sidebar-header'>CONTENT</div>
16                   ...
17                   <SidebarNavItem glyph='icon-name' name='Name of the Tab'↩
                         href='/newPath' />
18                   ...
19                 </SidebarNav>
20               </div>
21             </Col>
22           </Row>
23         </Grid>
24       </div>
25     );
26   }
27 }
28
29 ...
```

3 Read a new file in Dropzone

Dropzone Documentation: http://www.dropzonejs.com/

Go to imports > components > Dropzone.js

Listing 4: Add new case inside componentDidMount()

```

...

    case 'fileName.xxx':

          reader.onload = function(){

            let data = reader.result;
            data = d3.xxx.parse(data);

            Meteor.call('nameOf.method.toRemove.data');
            Meteor.call('nameOf.method.toAdd.data',data);

          };

...
```

Go to imports > schemas > index.js

Mongo Documentation: https://docs.mongodb.com/v3.2/core/databases-and-collections/

Listing 5: Export a new Meteor collection

```

...

    export const CollectionName = new Mongo.Collection('collection.name');

...
```

Go to imports > schemas > methods.js

Meteor Methods Documentation: https://guide.meteor.com/methods.html

Listing 6: Import the Collection and declare the Meteor methods in order to set their functions

```
1
2  ...
3      import { CollectionName} from '../schemas';
4
5  ...
6      "nameOf.method.toRemove.data": function(dataArray) {
7
8        CollectionName.remove({});
9
10      },
11
12      "nameOf.method.toAdd.data": function(dataArray) {
13
14
15        dataArray.map((data)=>{
16
17          CollectionName.insert(data);
18
19
20        });
21      },
22 ...
```

Go to imports > publications > publications.js

Listing 7: Publicate the Meteor Collection so it will be available all over the application

```
1
2  ...
3
4      import { CollectionName} from '../schemas';
5
6  ...
7
8      Meteor.publish('collection.name', function() {
9          return CollectionName.find({});
10     });
11
12 ...
```

To use the data inside a Component you most import the collection and unpack the data
Go to imports > routes > YourComponent.js

Listing 8: Publicate the Meteor Collection so data will be available all over the application

```
...
import { composeWithTracker } from 'react-komposer';
...
import { CollectionName } from '../route/to/schemas';
...
class YourComponent extends React.Component {
...
  render() {

  let data = this.props.data.map((d,key)=>{

    return(d);

        });


    return (
      <div>
        This is your data {data}
      </div>
    );
  }
}
...

function composer(props, onData) {

  const subscription = Meteor.subscribe("collection.name");
  if (subscription.ready()) {

    const data =  {
      data: CollectionName.find({}),
      ready: true,
    };
    onData(null, data);
  } else {
    onData(null, {ready: false, data:false});
  }
}
export default composeWithTracker(composer)(YourComponent);
```

4 Add a new Balancing Area

Geojson Documentation: http://geojson.org/

Go to imports > nodes

- Add the new Geojson file containing the geojson polygon node_XX.jsonther country/culture.

Go to imports > routes > Home.js

Listing 9: Import the Geojson file add a new color and use the new node

```
1
2  ...
3  import nodeXX from '../nodes/node_XX';
4  ...
5
6  const colors = { 'XX': "#feb24c",...];
7  ...
8
9  let nodes = [node0,node1,node2,node3,..,nodeXX]
10 ...
```

5 Add a new Load Zone

Go to imports > nodes > Points > coordinates.json

Listing 10: Add the point coordinates

```
1
2  {
3   "XX": [ 34.336138412413128 ,-116.562371167521917 ]  ,
4   "00": [ 25.941992862100989, -100.837135981799818 ]  ,
5   ...
6   }
```

Go to imports > nodes > Points > points.json

Listing 11: Add the Geojson point features

```
1
2  ...
3  { "type": "Feature", "properties": { "ID": "XX" }, "geometry": { "type": "↩
     Point", "coordinates": [ -116.562371167521917, 34.336138412413128 ] } ↩
     }
```