

# Building a TabBar Component

We're going to need to track which tab is active, change that on click, and re-render the tab list. We also will eventually need to swap which content panel is visible as well. I added a `<TabBar>` component that takes an array of tab names and labels and renders the tabs. Now, we could have the `<TabBar>` store the value of which tab is currently selected, but it's a good practice in React to write "container components" which store the state, and keep other components that are "presentational" and just display things based on the props they're given. So, we'll create a `<TabBarController>` component to track which tab is currently selected:

**Commit 325a103: Create a TabBar component to render a list of tabs**

## features/tabs/Tab.jsx

```
import React from "react";
import {Menu} from "semantic-ui-react";

const Tab = ({name, label, onClick, active}) => (
  <Menu.Item
    name={name}
    content={label}
    active={active}
    onClick={() => onClick(name)}
  />
);

export default Tab;
```

## features/tabs/TabBar.jsx

```
import React from "react";
import {Menu} from "semantic-ui-react";
```

```

import Tab from "../Tab";

const TabBar = (props) => {
  const {tabs, currentTab, onTabClick, ...otherProps} = props;

  const tabItems = tabs.map(tabInfo => {
    const {name, label} = tabInfo;

    return (
      <Tab
        key={name}
        name={name}
        label={label}
        active={currentTab === name}
        onClick={onTabClick}
      />
    );
  });

  return (
    <div>
      <Menu tabular attached="top" {...otherProps}>
        {tabItems}
      </Menu>
    </div>
  )
}

export default TabBar;

```

## [features/tabs/TabBarController.jsx](#)

```

import React, {Component} from "react";

import TabBar from "../TabBar";

export default class TabBarController extends Component {
  constructor(props) {
    super(props);

    const {tabs = [{name : null}]} = props;

    const firstTab = tabs[0];

    this.state = {

```

```

    this.state = {
      currentTab : firstTab.name
    } ;
  }

  onTabClick = (name) => {
    this.setState({currentTab : name});
  }

  render() {
    const {tabs, ...otherProps} = this.props;
    const {currentTab} = this.state;

    return (
      <TabBar
        {...otherProps}
        currentTab={currentTab}
        onTabClick={this.onTabClick}
        tabs={tabs}
      />
    )
  }
}

```

With that in place, we can then use it to render our main set of tabs:

### Commit 98a0e4d: Render the list of tabs using TabBarController

## App.js

```

class App extends Component {
  render() {
    const tabs = [
      {name : "unitInfo", label : "Unit Info"},
      {name : "pilots", label : "Pilots"},
      {name : "mechs", label : "Mechs"},
      {name : "unitOrganization", label : "Unit Organization"}
    ];

    return (
      <div className="App">
        <div className="App-header">
          <Header inverted as="h1">Project Mini-Mek</Header>
        </div>
        <TabBarController
          tabs={tabs}
          currentTab="unitInfo"
        />
      </div>
    )
  }
}

```

```
        <Container>
          <TabBarController tabs={tabs} size="massive" />
        </Container>
      </div>
    );
  }
}
```

No major visual changes from this, but we can now click on each tab and see the “active” tab get highlighted.