Container

29 August 2017 01:41

Container is a connection between provider and component, it takes the data from provider and provides that to component.

In order to create container you will require below things

1. import {**connect**} from 'react-redux';
2. import {**bindActionCreator**} from 'redux';

- connect will be used to make application state available to container.
- bindActionCreator will be used to bind the action (action function)

Steps:
1: create containers dir in src folder
2: create user-list.js file and write below code.

```
import React, { Component } from 'react';
import { connect } from 'react-redux';
import { bindActionCreators } from 'redux';
class UserList extends Component{
render(){
return(
<ul>
<li>First User</li>
<li>Second User</li>
<li>Third User</li>
</ul>
)
}
}
export default UserList;
```

3: import UserList on components/app.js file and call UserList component just below the UserList heading.
Follow the code.

```
import React from 'react';
import UserList from '../containers/user-list'
const App = () =>(
<div>
<h1>React-Redux App</h1>
<hr/>
<h2>User List</h2>
<UserList/>
<hr/>
<h2>User Detail</h2>
</div>
);
export default App;
```

Till here we are exposing the static data that our App component is displaying. Now we need actual data that will be coming from Provider to container and container will provide it to App Component.
Now we need one functions in order to map the state to props and connect the method to component.

```
function mapStateToProps(state){
return{
users:state.users
}
}
```

mapStateToProps(state) : it takes the application state from store and passes in the container as props. Here above code takes the store data, and we are taking only users from the available state and assigning it to users property. This users will be available to User-List component.
As state has been provided as props to container using Provider and this will be passed to all the containers, now it is our responsibility to map the state to props.

Next we need to connect the mapStateToProps with Container.

```
export default connect(mapStateToProps)(UserList);
```

And now we can iterate the props to show the data coming from provider on container.

```
class UserList extends Component{
createUserListItems(){
return this.props.users.map((user)=>{
return <li>{user.first} - {user.last}</li>
})
}
render(){
return(
<ul>
{this.createUserListItems()}
</ul>
)
}
}
```

Follow complete code.

```
import React, { Component } from 'react';
import { connect } from 'react-redux';
import { bindActionCreators } from 'redux';
class UserList extends Component{
createUserListItems(){
return this.props.users.map((user)=>{
return <li key={user.id}>{user.first} - {user.last}</li>
})
}
render(){
return(
<ul>
{this.createUserListItems()}
</ul>
)
}
}
function mapStateToProps(state){
return{
users:state.users
}
}
export default connect(mapStateToProps)(UserList);
```