

The store supplies data and the ability to dispatch actions to our React components.

React components subscribe to the Redux store via:

- 'connect()' and are given data and thunk action creators as props.
- or 'useSelector()' to get data from our state and 'useDispatch()' to get the dispatch function.

# React & Redux

Front end , User Interface  
\*Client Side\*

Store

React

Our components dispatch actions when a user interacts with the UI.

Middleware

Resolved promises dispatch a normal action.

Thunk middleware intercepts every dispatched action.

- If the action is an object, it is sent to the reducer.
- If the action is a function, it stops the action and invokes the action with dispatch and getState as arguments.

Reducers

Reducers are the blueprint for our Redux state. Every action hits every reducer. Reducers may return a new state based off of the type property of each dispatched action.

Front End

Back End

Invoked thunk actions send an ajax request to the back end.

Back end routes send up data for our response.

Routes

Models send up data from our back end to the matching route.

Backend routes determine which models are used to extract data from the database

# Express & Sequelize

Back end, Database Management  
\*Server Side\*

Models

The models communicate with our database.

Database

