React – Redux Async

In this lab, you will modify the Redux solution to use asynchronous actions.

# Objectives

In this lab, you will

* Modify an existing Redux app
* Add a new property to the <Quote> component, isDelayed
* Modify the QuoteActions to add new actions
* Modify the QuoteReducer to handle the new actions
* Run the app

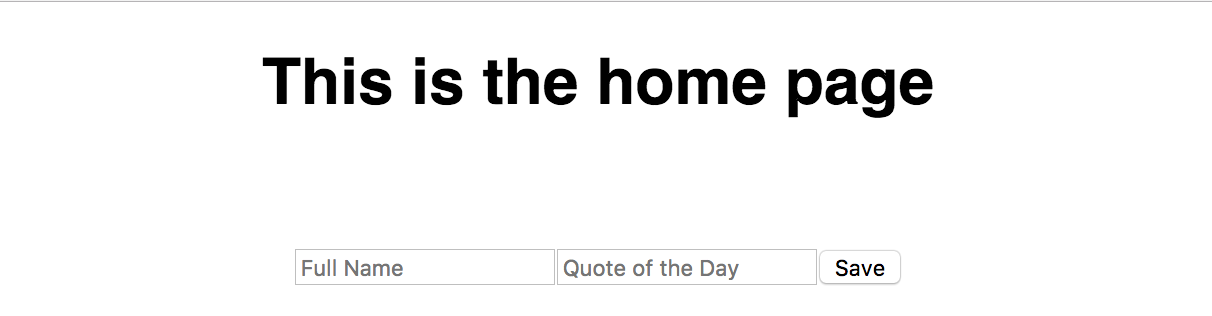
# Run the Redux app to verify correctness

1. Go to the project solution/react-app folder and start the solution with:

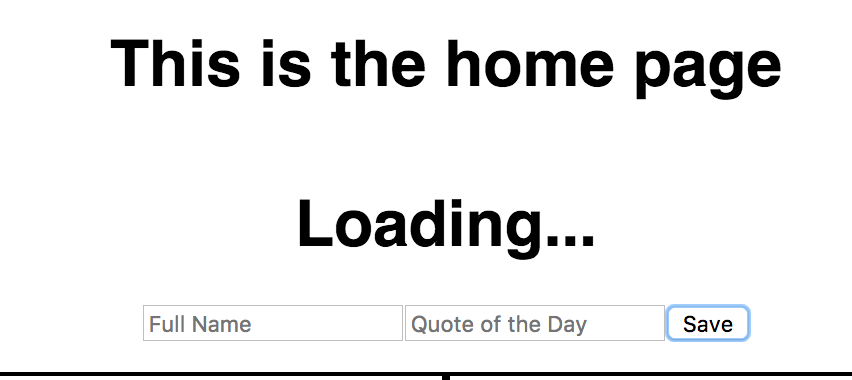
yarn install

yarn start

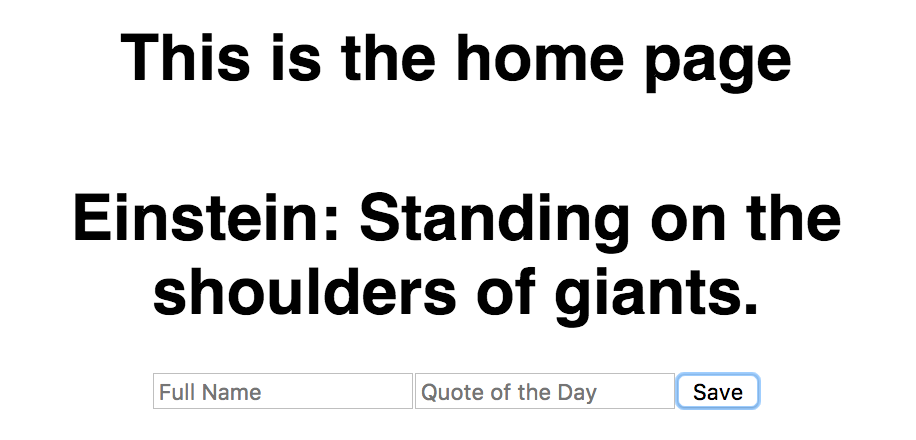
1. Open the app using <http://localhost:3000> You should see:



1. Enter a new name and quote and click save.
2. You should see the following two images: The first one lasts for two seconds, then the second one displays.



1. Two seconds later, the new quote appears like the following:



1. Stop the server. Go to the lab/react-app and install and start the lab server.
2. You should see the same pages but with no delay between the entry and the displayed text. Our task is to make the actions asynchronous.

# Modify the <Quote> Component

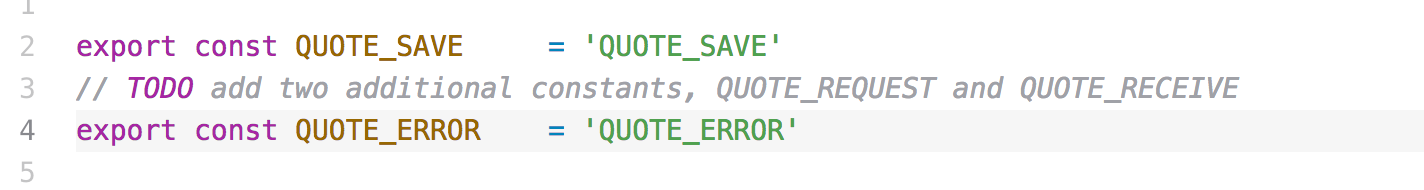
1. The <Quote> component looks like this:



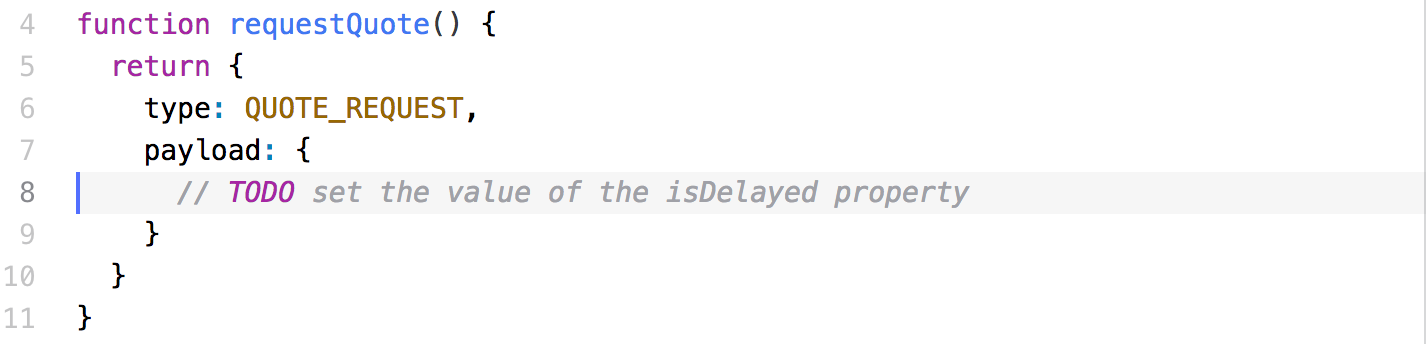
1. Notice where line 18 displays the results of the entry. The new input props.isDelayed controls whether this line renders. Add the logic to display either the “Loading…” message or the current quote.

# Modify the QuoteActions

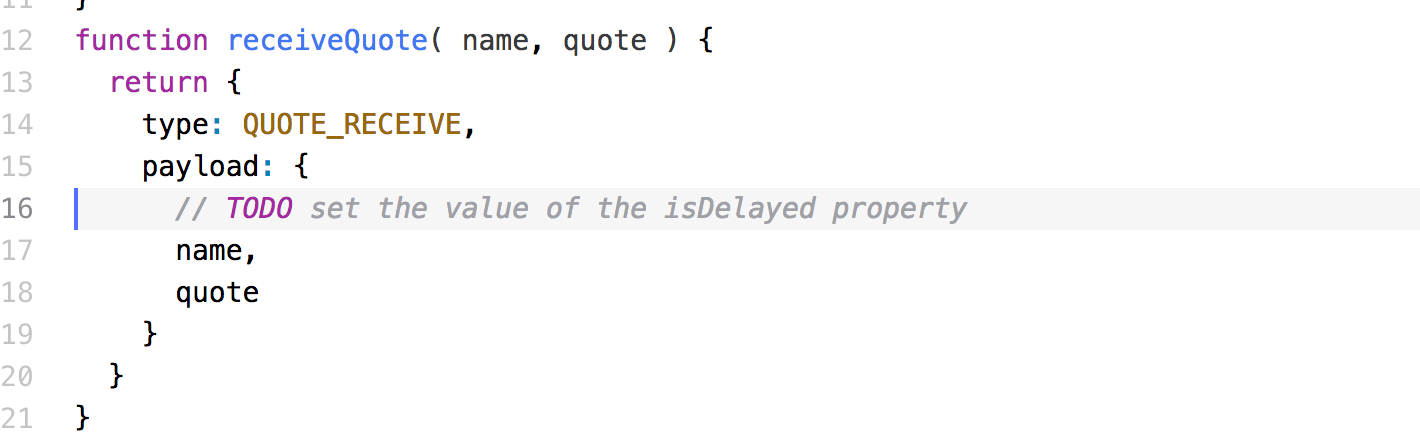
1. The action, QUOTE\_SAVE, invoked from the <Quote> component should start a sequence of actions. First, the action should dispatch the QUOTE\_REQUEST action which sets the isDelayed property to true. After the asynchronous return, it should set isDelayed to false and set the state data to the appropriate value.
2. Examine the file, src/quotes/QuoteConstants.js below:



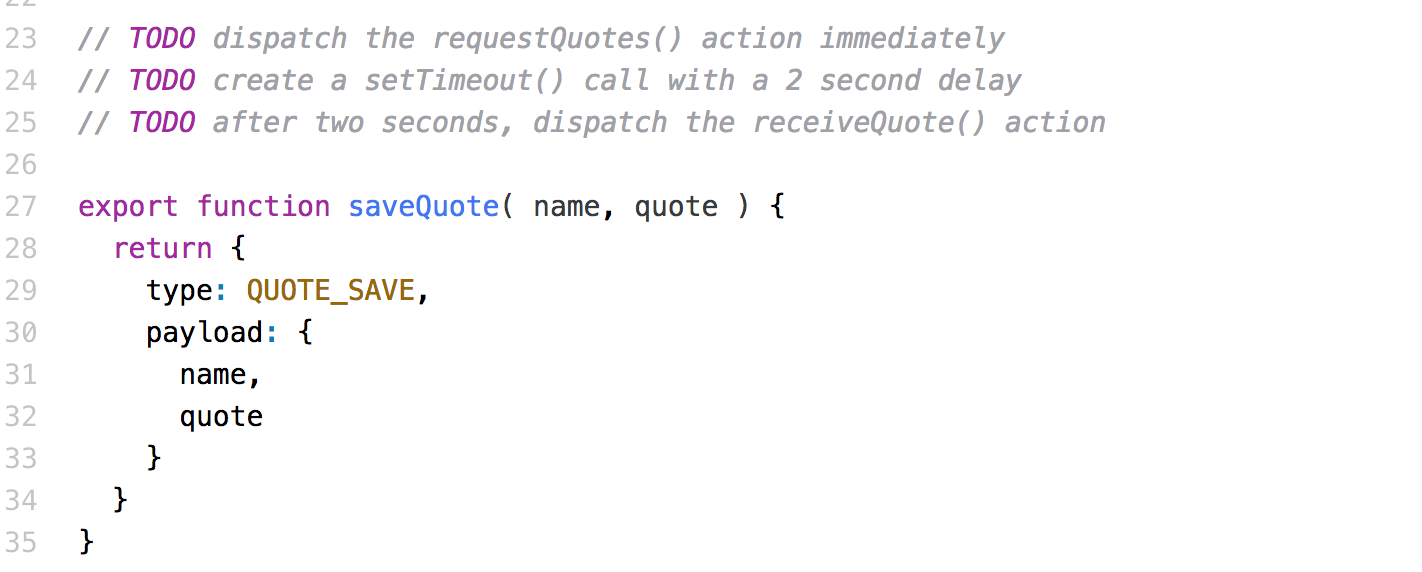
1. Add two additional constants, QUOTE\_REQUEST and QUOTE\_RECEIVE. Later, the requestQuote() function sets the isDelayed property to true. This tells the <Quote> component to display the “Loading…” message.



1. In the above, dispatch the action command. The type should be QUOTE\_REQUEST, and the payload should contain the isDelayed property which should be true.



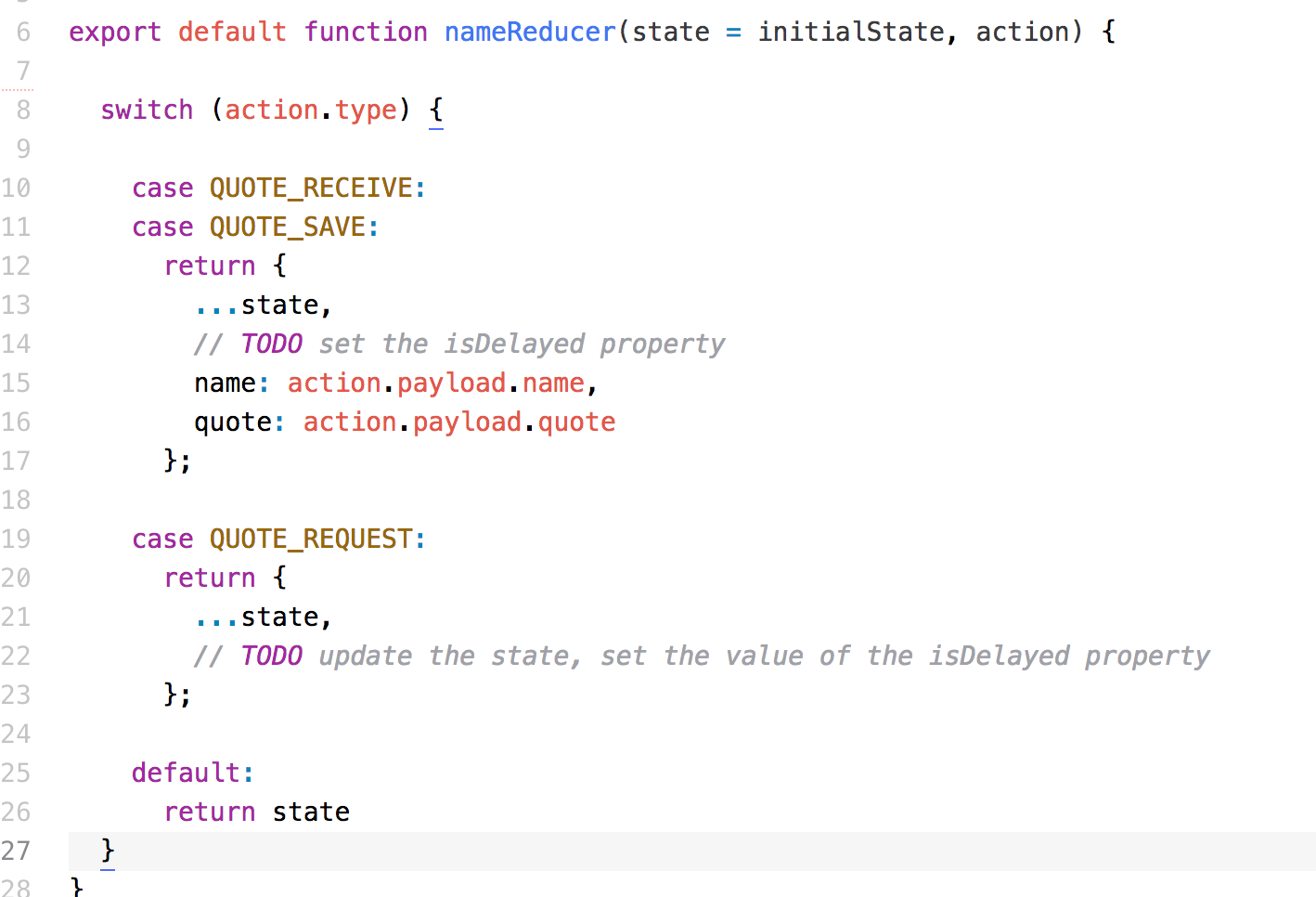
1. In the above, dispatch the action command. Make sure the payload contains both input properties as well as the ‘isDelayed’ field as ‘false’



1. The above shows the original saveQuote() function. The modified one should return an anonymous function which takes dispatch as its only argument. This anonymous function should dispatch the requestQuote() action, then wait for two seconds and dispatch the receiveQuote() action. This sequence tells the reducer to set the isDelayed property on the <Quote>. First, it sets it to true, then later, it sets it to false.

# Modify the QuoteReducer

1. Examine the lab/react-app/src/quotes/QuoteReducer.js file



1. The QUOTE\_RECEIVE and QUOTE\_SAVE actions should both set the state property, isDelayed, to the input value in the payload.
2. The QUOTE\_REQUEST action should set isDelayed to the input value in the payload.

# Modify the <QuoteContainer> Component

1. Redux create the <QuoteContainer> component and maps the input properties and the callbacks.



1. As shown above, map the ‘isDelayed’ property from the state. This passes the property to the <Quote> component.
2. Restart the server.

Congratulations. You have completed this lab.