Redux with React

Stay Strong!

TriValley coders

What Is Redux?

“Redux is a predictable state container for JavaScript apps”

i.e., not just React

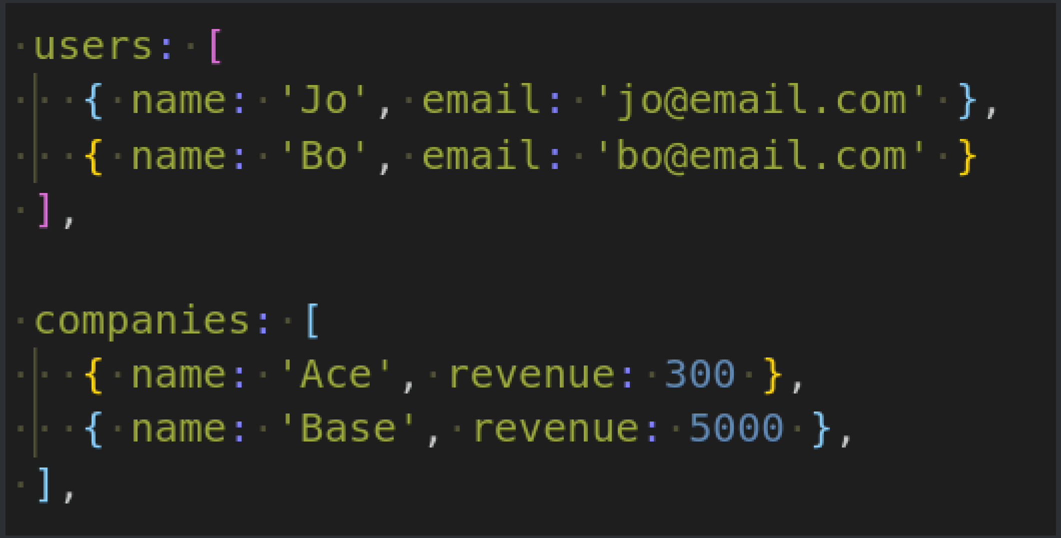
What Does Redux do?

- It helps you create, modify & maintain your applications state
- It does so in a predictable way
- It makes things easier *once you get good at it*

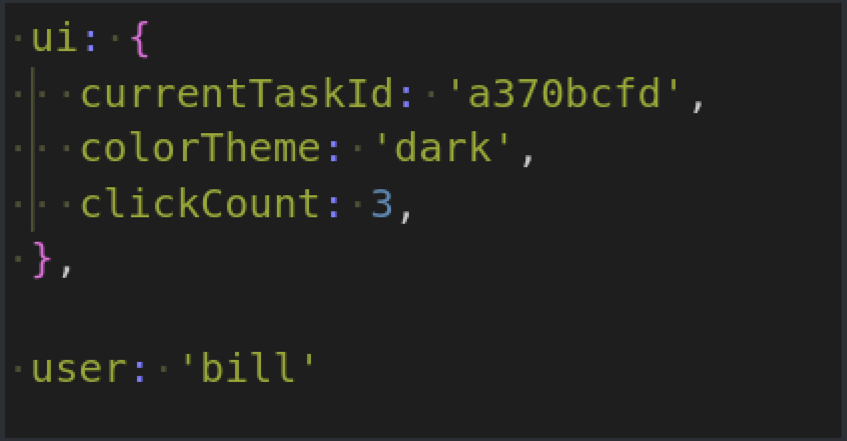
Packages to Add

- Required
  - react-redux
  - redux
- Optional
  - redux-logger
- Advance
  - redux-thunk

State



```
users: [
  { name: 'Jo', email: 'jo@email.com' },
  { name: 'Bo', email: 'bo@email.com' }
],

companies: [
  { name: 'Ace', revenue: 300 },
  { name: 'Base', revenue: 5000 },
],
```



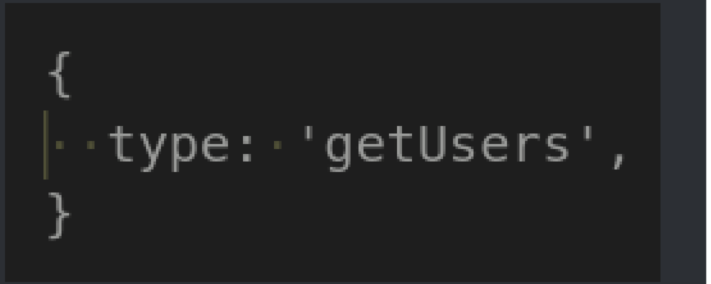
```
ui: {
  currentTaskId: 'a370bcfd',
  colorTheme: 'dark',
  clickCount: 3,
},

user: 'bill'
```

Actions & Reducers

Action

`{ } - just an object`

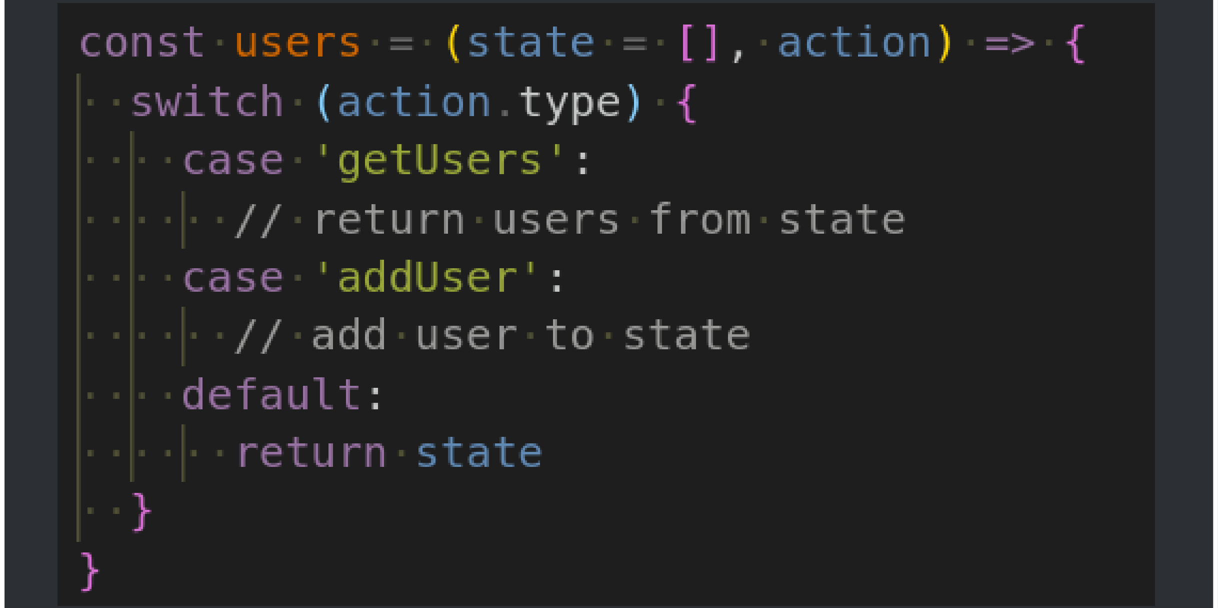
`onClick(getUsers())`



```
{
  type: 'getUsers',
}
```

→ Reducers

function ( )



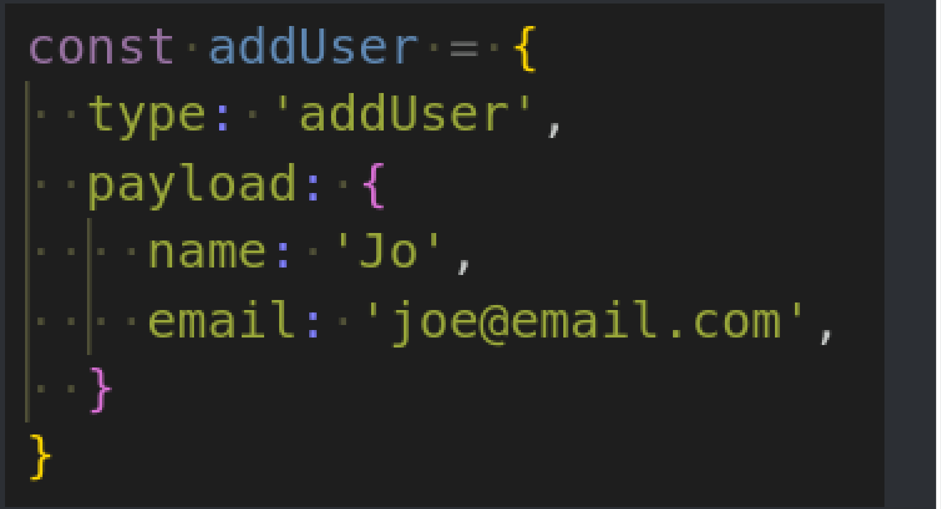
```
const users = (state = [], action) => {
  switch (action.type) {
    case 'getUsers':
      // return users from state
    case 'addUser':
      // add user to state
    default:
      return state
  }
}
```

→ UI Updates

render()

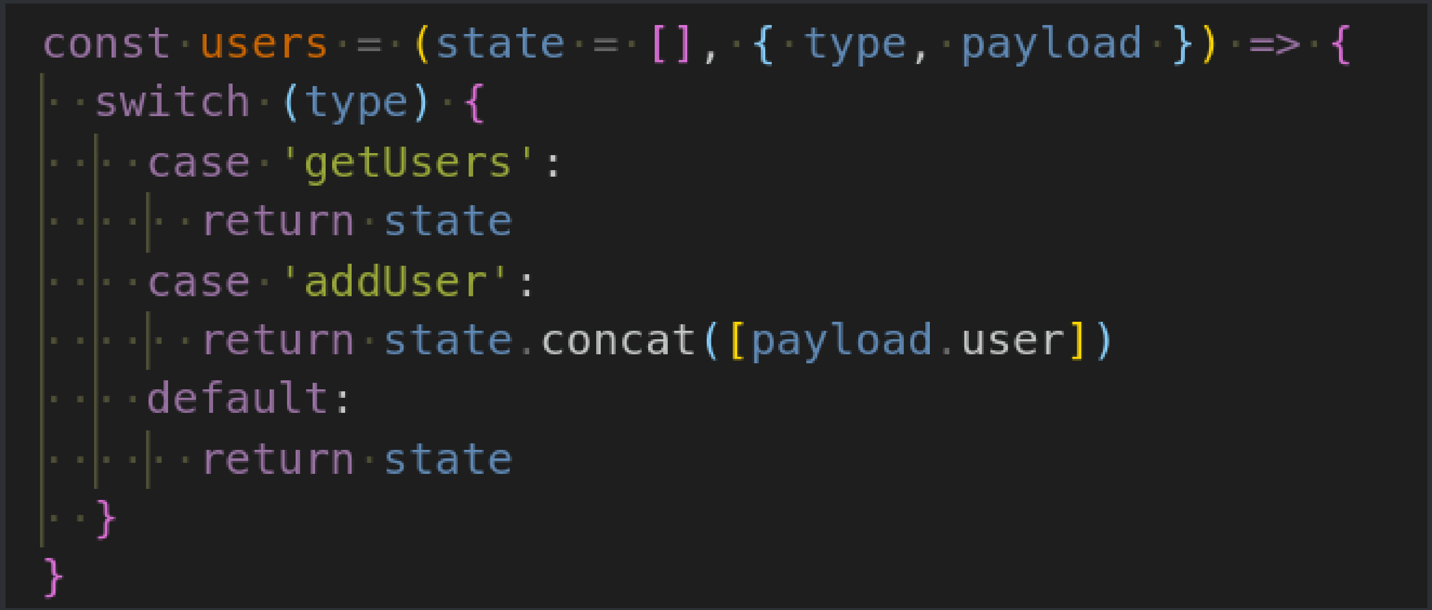
Actions & Reducers

Action



```javascript
const addUser = {
  type: 'addUser',
  payload: {
    name: 'Jo',
    email: 'joe@email.com',
  }
}
```

Reducers

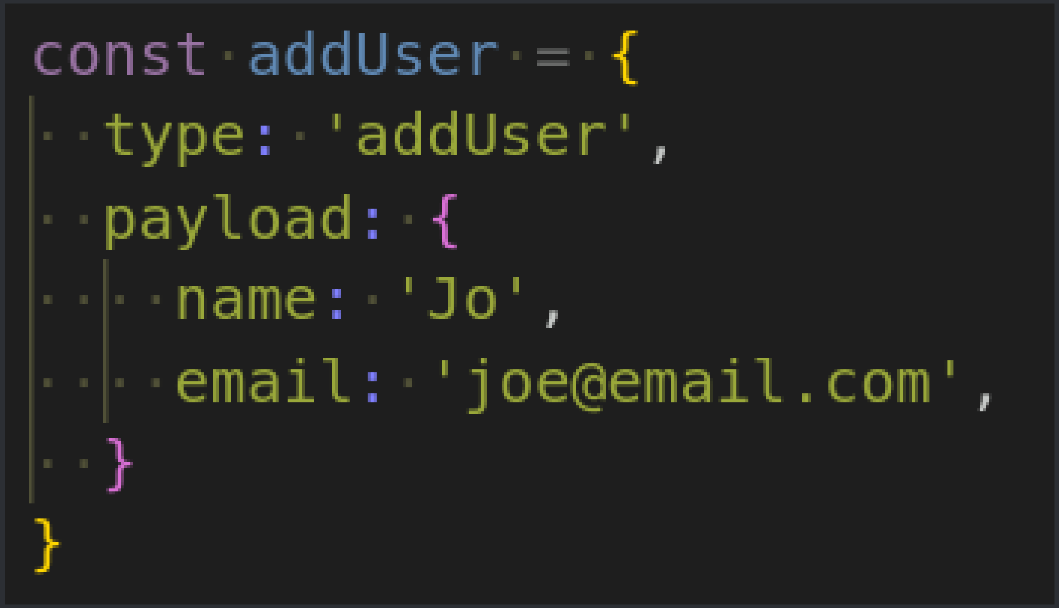


```javascript
const users = (state = [], { type, payload }) => {
  switch (type) {
    case 'getUsers':
      return state
    case 'addUser':
      return state.concat([payload.user])
    default:
      return state
  }
}
```

TriValley coders

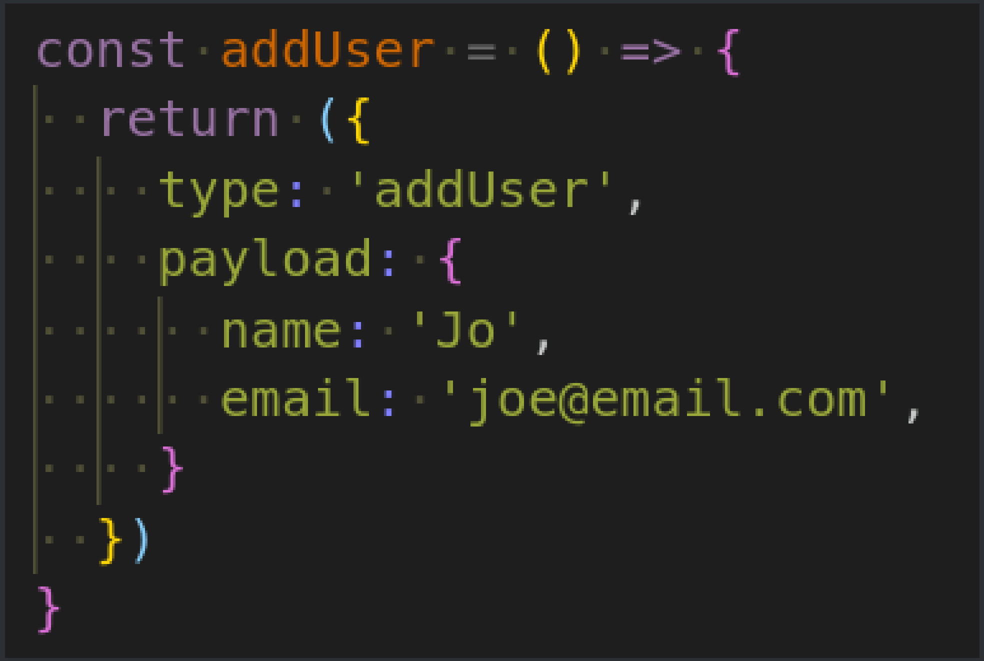
“Action Creator”

Action



```
const addUser = {
  type: 'addUser',
  payload: {
    name: 'Jo',
    email: 'joe@email.com',
  }
}
```

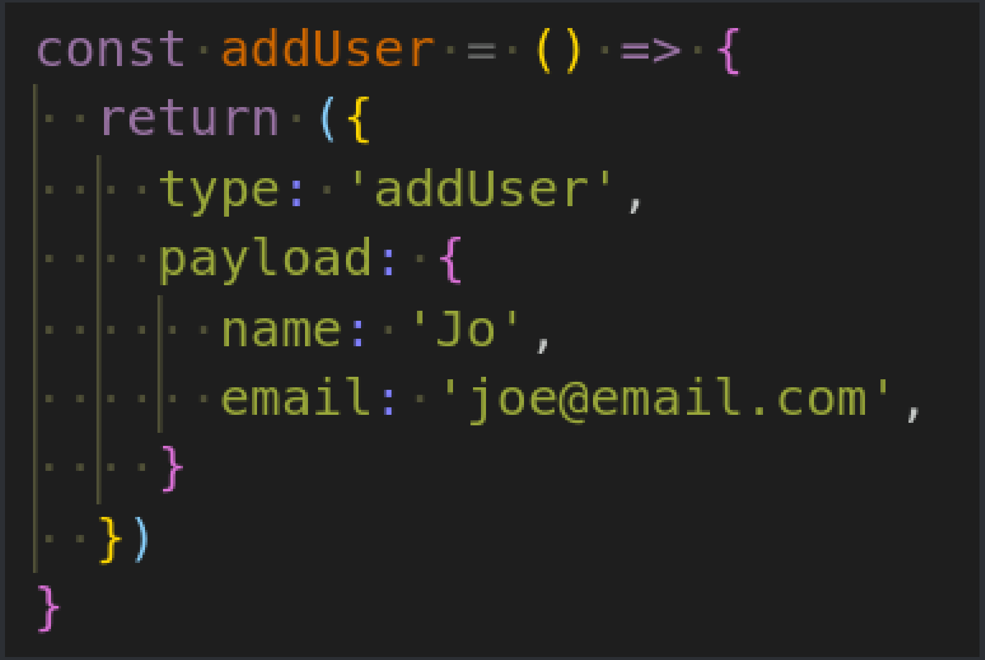
Action Creator



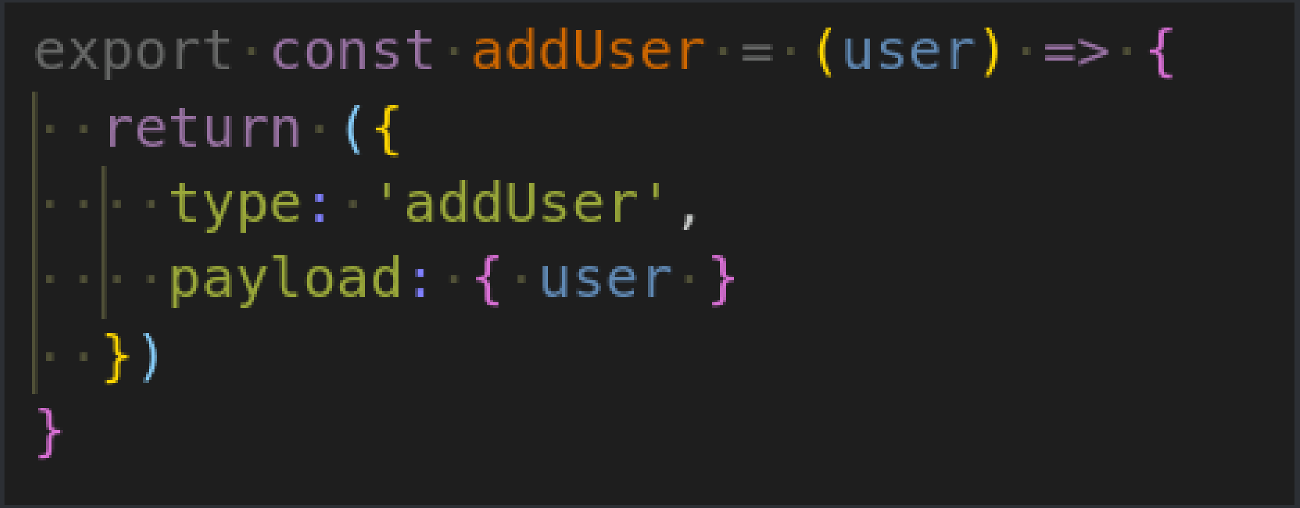
```
const addUser = () => {
  return ({
    type: 'addUser',
    payload: {
      name: 'Jo',
      email: 'joe@email.com',
    }
  })
}
```

TriValley coders

“Action Creator”



```
const addUser = () => {
  return ({
    type: 'addUser',
    payload: {
      name: 'Jo',
      email: 'joe@email.com',
    }
  })
}
```



```
export const addUser = (user) => {
  return ({
    type: 'addUser',
    payload: { user }
  })
}
```

Writing the Code

- You write an action & reducer
- You call the action
- Redux sends your actions and state to your reducers
- Your reducer code modifies state
- Redux lets React know state has change
- Your UI updates

Yes, there is some magic glueing it all together

Redux API

Top Level

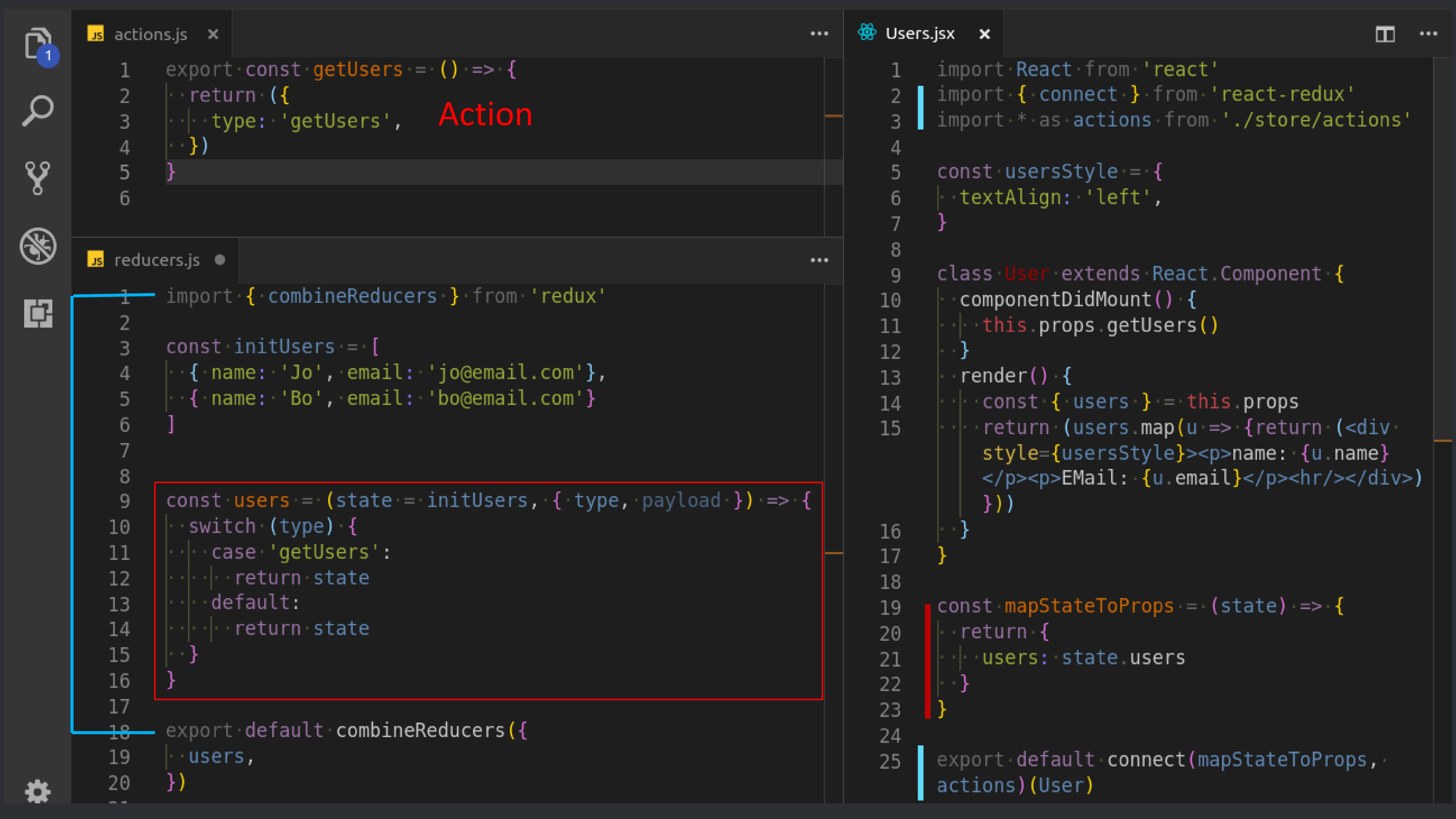
Function	Used in/by
createStore(reducer, [preloadedState], [enhancers]}	Configuration
combineReducers(reducers)	Configuration
applyMiddleware()	Configuration
Compose()	Configuration
bindActionCreators(actionCreators, dispatch)	Used via react-redux

store()

Function	Used in/by
getState()	Used via react-redux
dispatch(action)	Used via react-redux
suscribe(listener)	Used via react-redux
replaceReducer()	I haven’t used it yet

react-redux

Function	Used in/by
<Provider store>	Higher Order Component
connect([mapStateToProps], [mapDispatchToProps], [mergeProps], [options])	- Subscribe to store/state changes - Map props to parts of state
	* & More



```js
export const getUsers = () => {
  return ({
    type: 'getUsers',
  })
}

```

```js
import { combineReducers } from 'redux'

const initUsers = [
  { name: 'Jo', email: 'jo@email.com'},
  { name: 'Bo', email: 'bo@email.com'}
]


const users = (state = initUsers, { type, payload }) => {
  switch (type) {
    case 'getUsers':
      return state
    default:
      return state
  }
}

export default combineReducers({
  users,
})
```

```jsx
import React from 'react'
import { connect } from 'react-redux'
import * as actions from './store/actions'

const usersStyle = {
  textAlign: 'left',
}

class User extends React.Component {
  componentDidMount() {
    this.props.getUsers()
  }
  render() {
    const { users } = this.props
    return (users.map(u => {return (<div style={usersStyle}><p>name: {u.name}</p><p>EMail: {u.email}</p><hr/></div>)
    }))
  }
}

const mapStateToProps = (state) => {
  return {
    users: state.users
  }
}

export default connect(mapStateToProps, actions)(User)
```

Questions

BRIIA™

Thank you for hosting our event today!

TriValley Coders