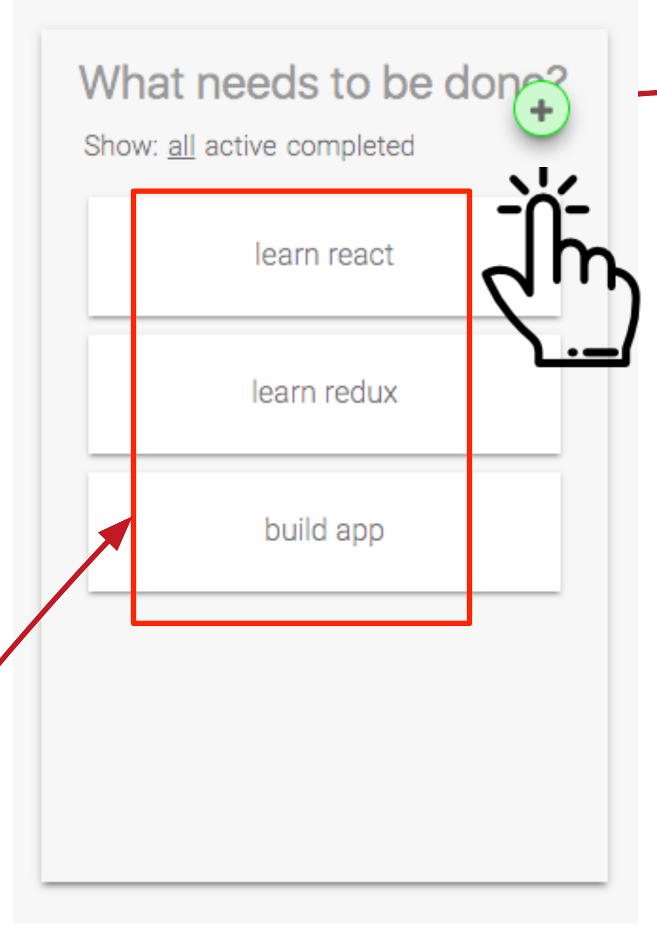
Redux Flux Cycle



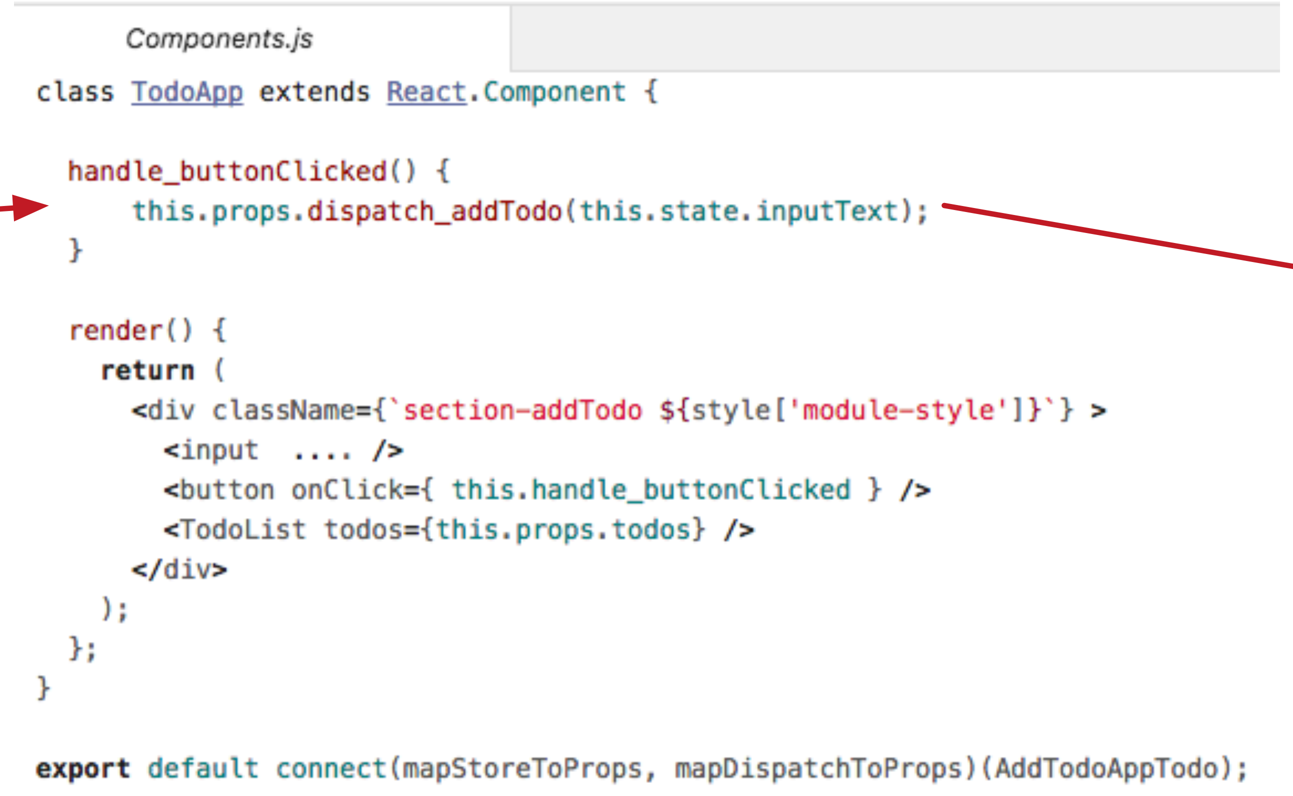
What needs to be done?

Show: all active completed

learn react

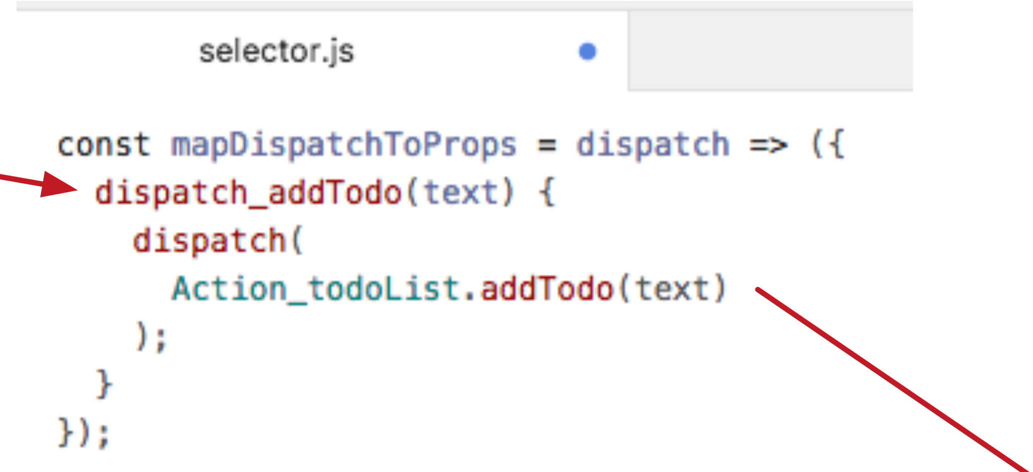
learn redux

build app



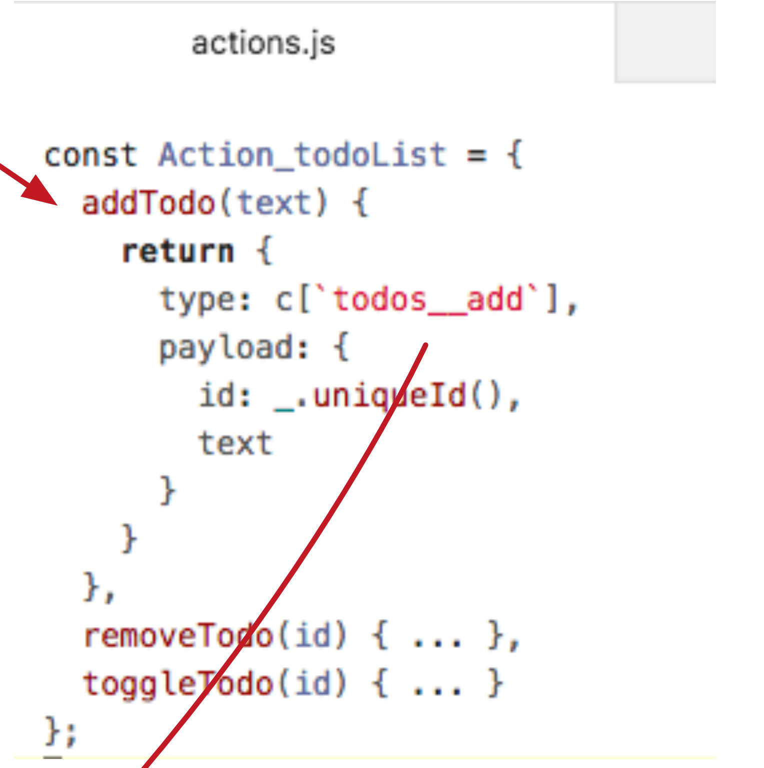
Components.js

```
class TodoApp extends React.Component {

  handle_buttonClicked() {
      this.props.dispatch_addTodo(this.state.inputText);
  }

  render() {
    return (
      <div className={`section-addTodo ${style['module-style']}`} >
        <input  .... />
        <button onClick={ this.handle_buttonClicked } />
        <TodoList todos={this.props.todos} />
      </div>
    );
  };
}

export default connect(mapStoreToProps, mapDispatchToProps)(AddTodoAppTodo);
```



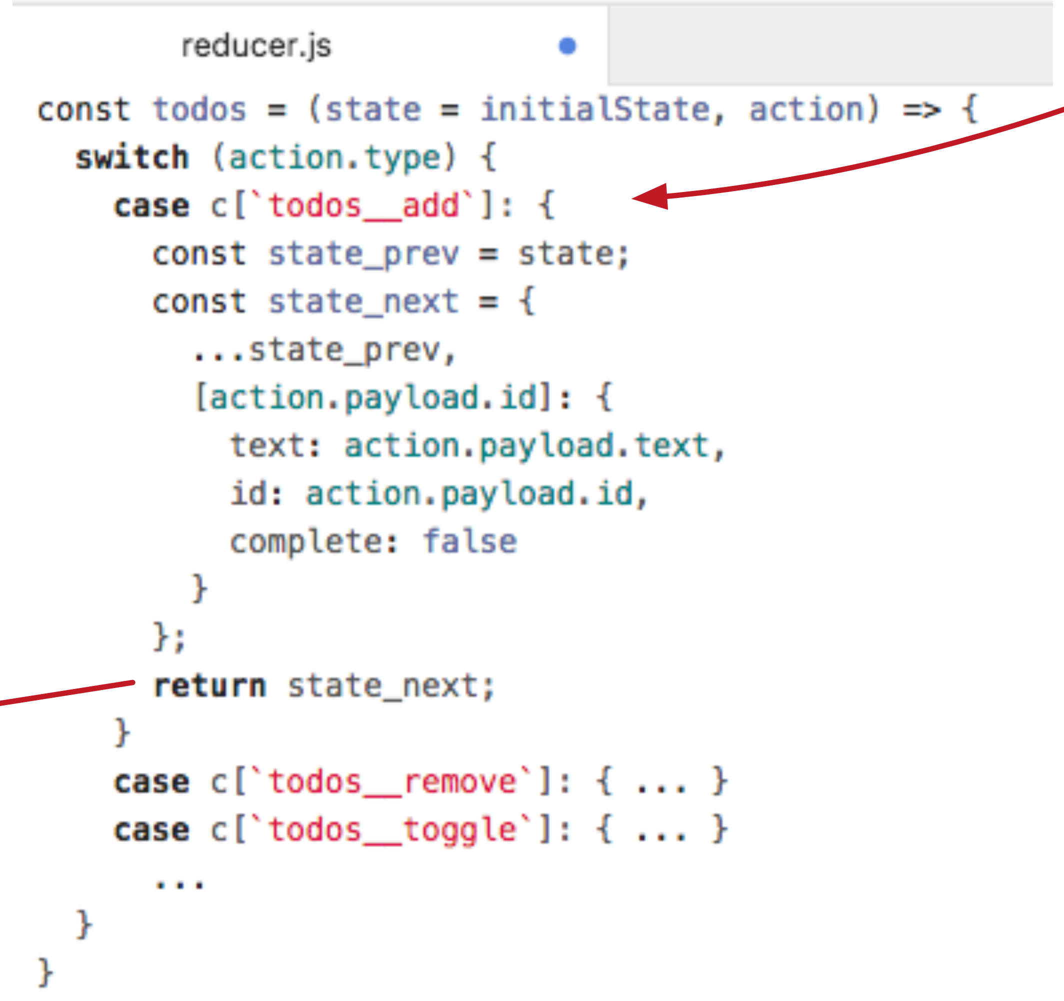
selector.js

```
const mapDispatchToProps = dispatch => ({
  dispatch_addTodo(text) {
    dispatch(
      Action_todoList.addTodo(text)
    );
  }
});
```



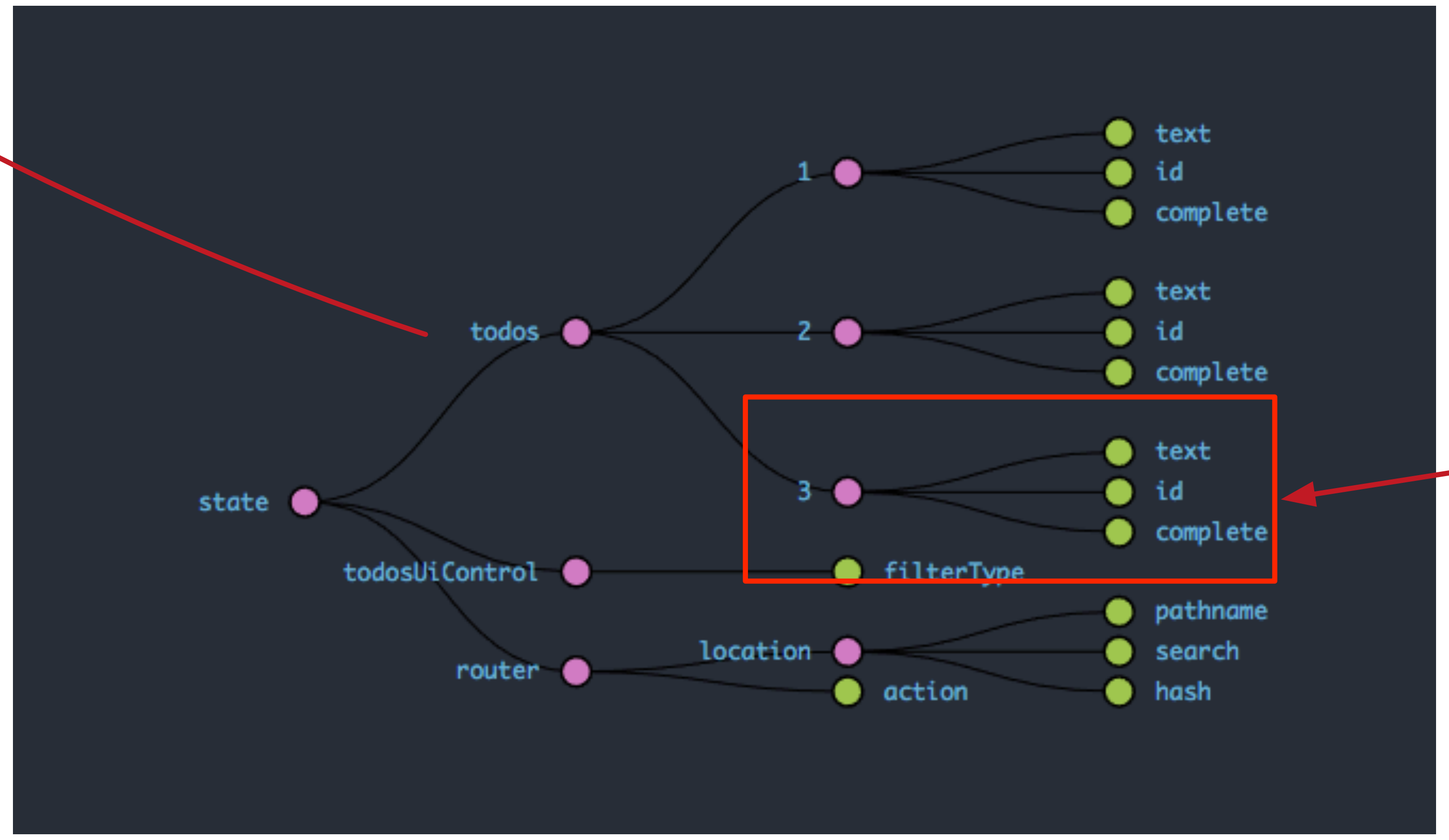
actions.js

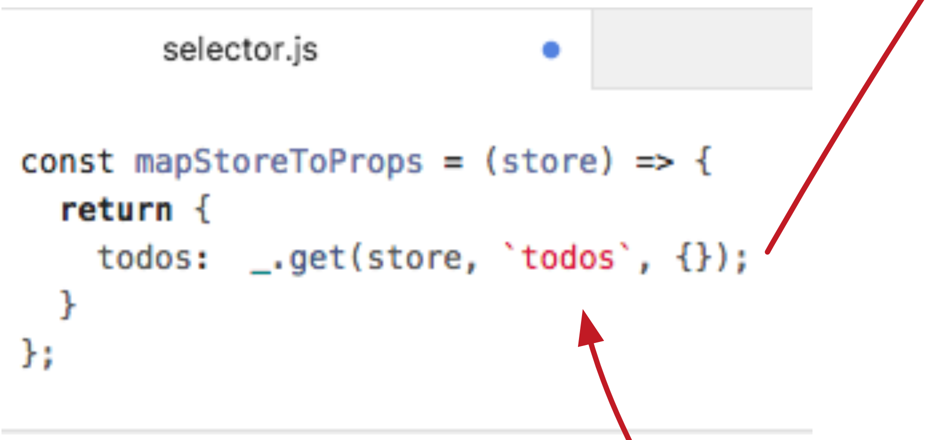
```
const Action_todoList = {
  addTodo(text) {
    return {
      type: c[`todos__add`],
      payload: {
        id: _.uniqueId(),
        text
      }
    }
  },
  removeTodo(id) { ... },
  toggleTodo(id) { ... }
};
```



reducer.js

```
const todos = (state = initialState, action) => {
  switch (action.type) {
    case c[`todos__add`]: {
      const state_prev = state;
      const state_next = {
        ...state_prev,
        [action.payload.id]: {
          text: action.payload.text,
          id: action.payload.id,
          complete: false
        }
      };
      return state_next;
    }
    case c[`todos__remove`]: { ... }
    case c[`todos__toggle`]: { ... }
    ...
  }
}
```





selector.js

```
const mapStoreToProps = (store) => {
  return {
    todos:  _.get(store, `todos`, {});
  }
};
```