

# 3. Свяжем stores с view

Пишем свой FluxConnect

```
3 export const fluxConnect = (stores = [], mapStateToProps = () => ({})) => WrappedComponent =>
4   class FluxConnect extends React.Component {
5     state = {}
6     unsubscribeFunctions = []
7
8     componentWillMount () {
9       this.subscribeOnStores()
10      this.stateToPropsMapper()
11    }
12
13    componentWillMount () {
14      this.onUnsubscribeOnStores()
15    }
16
17    onUnsubscribeOnStores = () => {
18      this.unsubscribeFunctions.forEach(unsubscribe => unsubscribe())
19    }
20
21    subscribeOnStores = () => {
22      this.unsubscribeFunctions = stores.map(store => {
23        return store.subscribe(() => {
24          this.stateToPropsMapper()
25        })
26      })
27    }
28
29    stateToPropsMapper = () => {
30      const storeStates = stores.map(({ getState }) => getState())
31      this.setState(mapStateToProps(...storeStates))
32    }
33
34    render () {
35      return <WrappedComponent {...this.props} {...this.state} />
36    }
37  }
```

### 3. Свяжем stores с view

Используем fluxConnect

```
const getTodos = todosState => todosState.todos

export default fluxConnect(
  [todosStore.store],
  todosState => ({
    todos: getTodos(todosState),
  }),
)(Todos)
```