

# Redux vs MobX part 2

Wachara

Growth Session #9 - November 16-17 2017

## Meet up with redux and mobX again



- are open-source libraries
- provide client-side state management
- support time-travel debugging via the [redux-devtools-extension](#)
- are not tied to a specific framework
- have extensive support for React/React Native frameworks.

# What the difference by implement source code



Use dispatch events by subscriber  
and a lot of middleware to  
communicate to each component



`@observable` variables and map into  
properties it can be access for change  
the value of variables directly



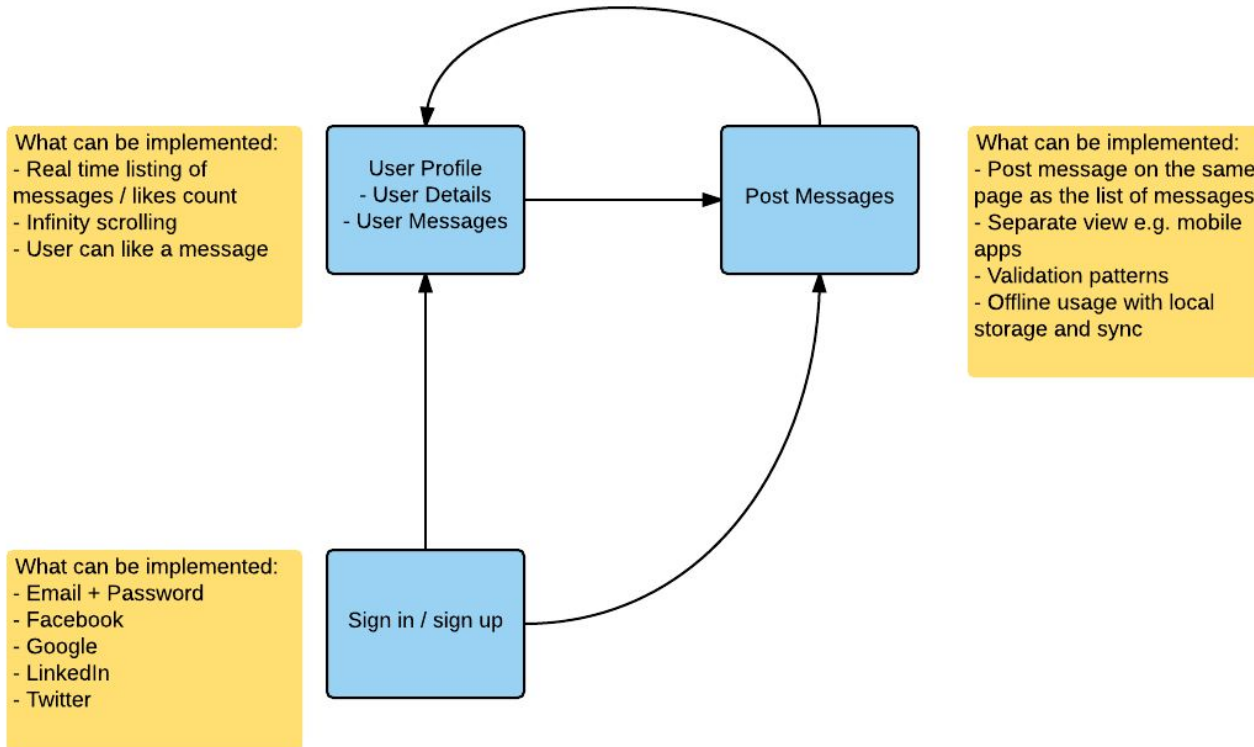
## **Into**

- Web / mobile application
- Messaging application e.g. simplified Twitter

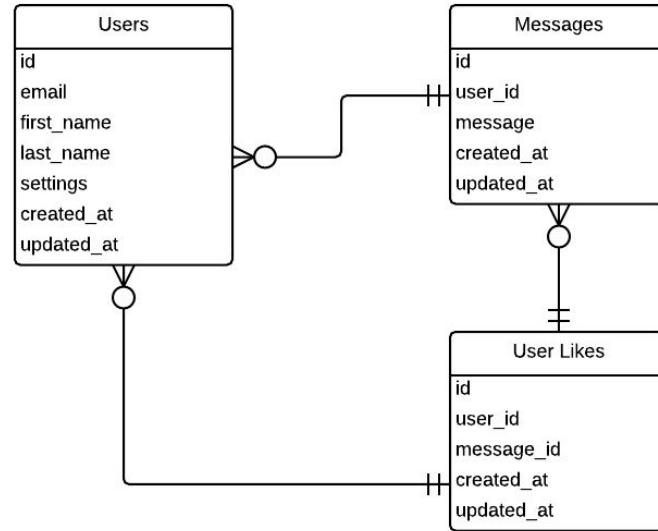
## **Benefits:**

- Real world application
- Re-usable patterns
- Diversity of implementations

# FLOW



# Data storage



Hosted at <https://nimbl3-demo-app-api.herokuapp.com/api/1/>

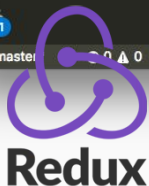
Available resources:

- **/users**
  - GET /
  - GET /:id
  - POST /
  - PATCH /:id
  - DELETE /:id
- **/messages**
  - GET /
  - POST /
  - DELETE /:id
- **/likes**
  - POST /

# First Step of each

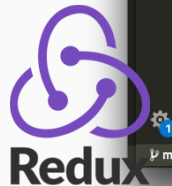
```
index.js — react-redux-app
JS index.js x
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import App from './App';
4 import registerServiceWorker from './registerServiceWorker';
5 import { Provider } from 'react-redux';
6 import { applyMiddleware, createStore } from 'redux';
7 import logger from 'redux-logger';
8 import thunk from 'redux-thunk';
9 import promise from 'redux-promise-middleware';
10 import TweetReducer from './reducers/TweetReducer';
11
12 const middleware = applyMiddleware(promise(), thunk, logger);
13 const store = createStore(TweetReducer, middleware);
14
15 ReactDOM.render(
16   <Provider store={store}>
17     <App />
18   </Provider>
19   , document.getElementById('root'));
20 registerServiceWorker();
21
```

```
index.js — react-mobx-app
JS index.js x
1 import './css/main.css'
2 import React from 'react'
3 import ReactDOM from 'react-dom'
4 import TweetStore from './stores/TweetStore'
5 import App from './App'
6
7 const app = document.getElementById("app")
8
9 ReactDOM.render(<App store={TweetStore} />, app)
10
```





# The store of each

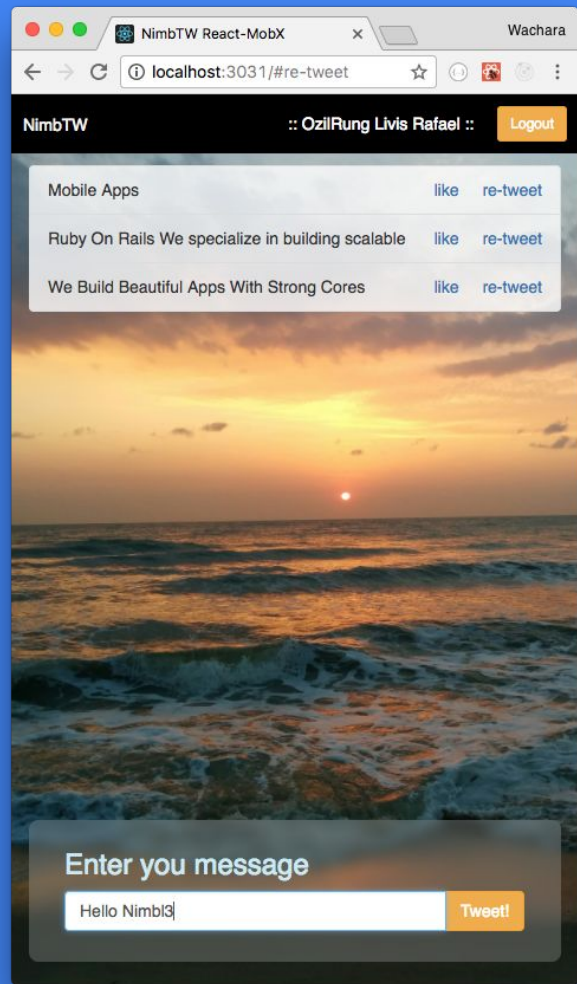


```
TweetReducer.js — react-redux-app
JS TweetReducer.js x
1 import { Constants } from '../constants/Constants';
2 import axios from 'axios';
3
4 const user = (action) => {
5   return {
6     id: Math.floor(Math.random()*1234567890),
7     user: action.user
8   }
9 }
10
11 const message = (action) => {
12   return {
13     id: Math.floor(Math.random()*1234567890),
14     message: action.message
15   }
16 }
17
18 const tweets = (state = [], action) => {
19   console.log(action);
20   let users = null;
21   let messages = null;
22   switch (action.type) {
23     case Constants.GET_USER:
24       users = [...state, user(action)];
25       return users;
26     case Constants.ADD_MESSAGE:
27       messages = [...state, message(action)];
28       return messages;
29     default:
30       return state;
31   }
32 }
33
34 export default tweets;
35
```

```
TweetStore.js — react-mobx-app
JS TweetStore.js x
1 import { computed, observable } from "mobx"
2
3 class User {
4   @observable user
5   @observable id
6
7   constructor(value) {
8     console.log('connect to ther server');
9     this.user = value
10    this.id = Math.floor(Math.random()*1234567890)
11  }
12 }
13
14 class Message {
15   @observable message
16   @observable id
17
18   constructor(value) {
19     console.log('connect to ther server');
20     this.message = value
21     this.id = Math.floor(Math.random()*1234567890)
22   }
23 }
24
25 export class TweetStore {
26   @observable tweets = []
27   @observable users = []
28
29   GetUsers(value) {
30     this.users.push(new User(value))
31
32   }
33
34   addMessage(value) {
35     this.tweets.push(new Message(value))
36   }
37 }
38
39 export default new TweetStore
```



# DEMO APP



# Thanks!

Contact Nimbl3

[hello@nimbl3.com](mailto:hello@nimbl3.com)

399 Sukhumvit Road, Interchange 21  
Klongtoey nua, Wattana  
Bangkok 10110

20th Floor, Central Tower  
28 Queen's Road  
Central, Hong Kong

[nimbl3.com](http://nimbl3.com)

