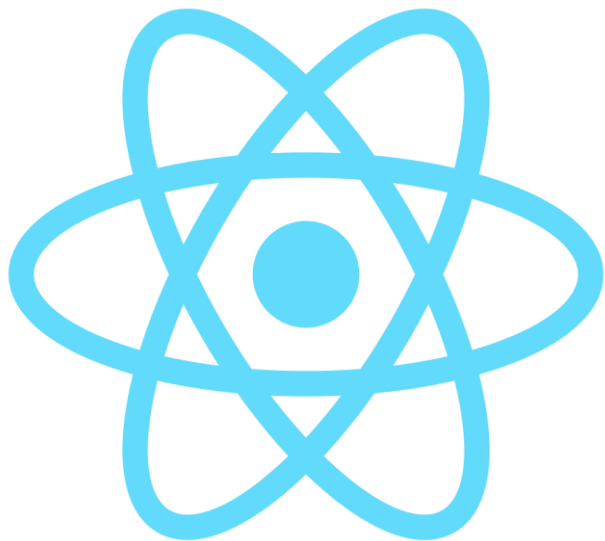
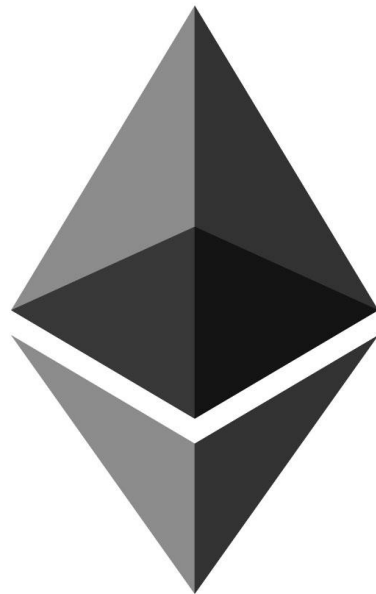
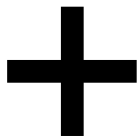


Scalable React dApp Architecture

Nick Dodson | Sep. 21 2016



REACT



ETHEREUM

Nick Dodson, founder **WeiFund/BoardRoom**

Engineer at ConsenSys

Worked on Ethereum dApps for about 2 years

Created the `meteor-dapp-boilerplate`

I've been quiet for sometime..

Working away on **WeiFund/BoardRoom**/dApp Architecture

Boilerplate... not a framework

Design opinions by default not by force

You can change any aspect or component

Boilerplate Expectations

DEVELOPMENT

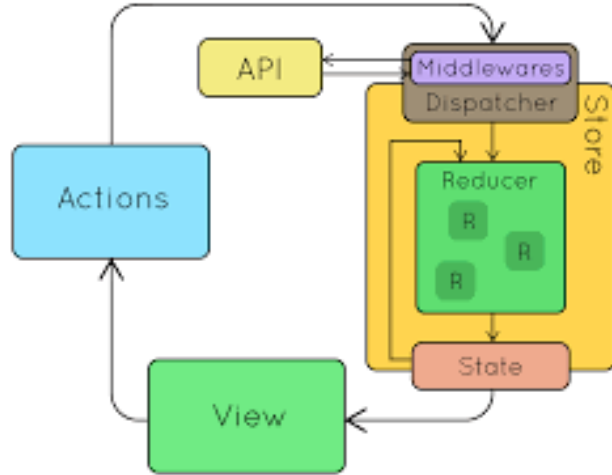
- **Contract Scaffolding** ('npm run generate contract' -> ``.sol`, `test.sol`, `test.js``)
- Enforced **Contract Linting/Testing** in two Languages [solium, dapple, chaithereum]
- **Wallet generation/Identity/fauceting during install**/pre-development [eth-lightwallet]
- **Multi-environment deployment/support** configuration [ethdeploy]
- **Testing on any chain/node** [[ethereumjs-testrpc](#), dapple/chaithereum/ethdeploy]
- Node support **without running a node** (signing done locally, password prompt) [ethereumjs-tx, Infura]
- Pre-configured for testrpc, testnet, livenet [ethdeploy]
- Pre-configured/abstracted web3 instance [es6 import]

Boilerplate Expectations

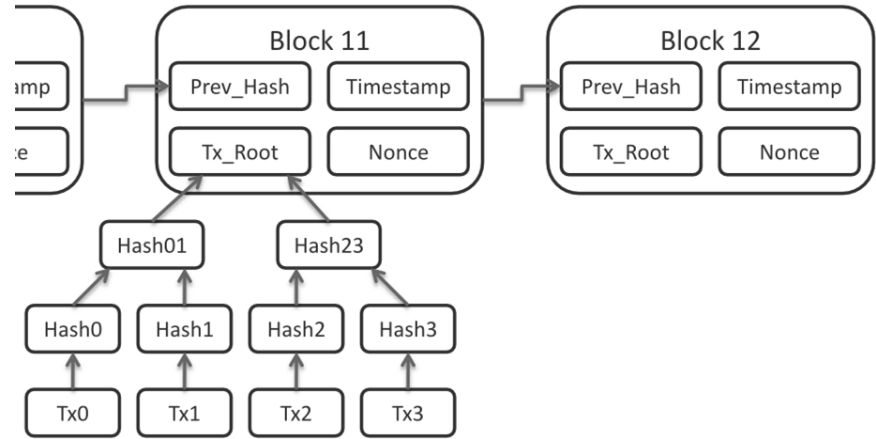
USER INTERFACE

- **Mature React Architecture** with ``github.com/mxstbr/react-boilerplate``
- **Multi-environment/chain awareness**
- **Prices listed in USD first**, Eth/BTC secondary (``kraken api``)
- **Basic multi-contract example dApp** (Registry + Factory + Example Contracts)
- **Offline-first**, dapp runs without internet (``npm start``... everything just works locally... no internet/no config)

Data Model Revelations



REDUX



BLOCKCHAIN

The data model revelations of the front-end world are very similar to those in the blockchain space.

State management.. so hot right now.

Mature React Architecture



Max Stoiber

mxstbr

🌐 Travels around the world, brews coffee, skis mountains and makes stuff on the web. Open source developer (**@KeystoneJS**, **@ElementalUI**) and part of **@react-vienna**

👤 @mxstbr

📍 Vienna, Austria

✉ contact@mxstbr.com

🔗 <http://mxstbr.com>

🕒 Joined on 8 May 2014

Organizations



👁 Watch ▾

293

★ Unstar

9,409

🍴 Fork

954

📖 README.md

Start your next react project in seconds

A highly scalable, offline-first foundation with the best DX and a focus on performance and best practices

dependencies up to date

devDependency Status

build passing

coverage 99%

backers 2

sponsors 2

❤ supported by

🗨 chat

🔗 on github

Made with ❤ by [Max Stoiber](#) and contributors. If you're using this boilerplate, we'd love to hear from you!

Features

Quick scaffolding
Create components, containers, routes, selectors and sagas - and their tests - right from the CLI!

Instant feedback
Enjoy the best DX (Developer eXperience) and code your app at the speed of thought! Your saved changes to the CSS and JS are reflected instantaneously without refreshing the page. Preserve application state even when you update something in the underlying code!

Predictable state management
Unidirectional data flow allows for change logging and time travel debugging.

Next generation JavaScript

github.com/mxstbr/react-boilerplate

Components of Note

CSS

PostCSS

CSS Modules

sanitize.css

JS

Redux

ImmutableJS

reselect

redux-saga***

I18n (react-intl)

redux-router

```
/**
 * Github repos request/response handler
 */
export function* getRepos() {
  // Select username from store
  const username = yield select(selectUsername());
  const requestURL =
    `https://api.github.com/users/${username}/repos?type=all&sort=
    updated`;

  // Call our request helper (see 'utils/request')
  const repos = yield call(request, requestURL);

  if (!repos.err) {
    yield put(reposLoaded(repos.data, username));
  } else {
    yield put(repoLoadingError(repos.err));
  }
}
```


Ethereum Integration

dependencies:

"web3": "^0.17.0-alpha",

Web3 enabled

...

devDependencies:

"ethdeploy": "0.1.3",

"ethdeploy-provider-http": "0.0.5",

"ethdeploy-provider-zero-client": "0.0.5",

"dapple": "^0.7.2",

"solium": "^0.1.16",

"eth-lightwallet": "^2.5.2",

"chaithereum": "^1.4.0",

"ethereumjs-testrpc": "^2.2.7"

....

Complex Deployment

Testing/linting

Contract Scaffolding (w/ failure default)

```
nick@nick-PEAQ-PNB-S1013-ISA1-ULTRA: ~/github/react-dapp-boilerplate
nick@nick-PEAQ-PNB-S1013-ISA1-ULTRA:~/github/react-dapp-boilerplate$
nick@nick-PEAQ-PNB-S1013-ISA1-ULTRA:~/github/react-dapp-boilerplate$ npm run generate

> react-boilerplate@3.1.0 generate /home/nick/github/react-dapp-boilerplate
> plop --plopfile internals/generators/index.js

? [PLOP] Please choose a generator.
  component - Add an unconnected component
  container  - Add a container component
  route      - Add a route
  language   - Add a language
> contract  - Add a Ethereum Solidity contract
```

Wallet Generation/Management (on-install)

```
nick@nick-PEAQ-PNB-S1013-ISA1-ULTRA: ~/github/react-dapp-boilerplate
nick@nick-PEAQ-PNB-S1013-ISA1-ULTRA:~/github/react-dapp-boilerplate$ npm run generate:wallet

> react-boilerplate@3.1.0 generate:wallet /home/nick/github/react-dapp-boilerplate
> node ./internals/scripts/generate-lightwallet.js

Ethereum Wallet Generation Process:
via eth-lightwallet

You are about to begin generating an Ethereum light wallet, please follow the steps below.
You will need a wallet for deploying to the Ethereum testnet, livenet and any custom networks
.
Note, this wallet will be generated in the folder outside this repository. Please ensure th
at location is secure.

-----

? Please type your new Ethereum wallet passphrase here:  *****
? Please re-type your passphrase here:  *****
The new KeyStore constructor has been deprecated in favor of KeyStore.createVault(), and will
be removed in a future version. Read issue #109 for info. https://github.com/ConsenSys/eth-l
ightwallet/issues/109

Ethereum wallet file generated!
via eth-lightwallet
```

Complex Contract Deployment (ethdeploy)

```
module.exports = {
  output: {
    environment: 'testrpc',
  },
  entry: {
    testrpc: require('./lib/classes.json'),
  },
  module: function(deploy, contracts){
    deploy(contracts.SimpleStoreRegistry).then(function(simpleStoreRegistry){
      deploy(contracts.SimpleStoreFactory, simpleStoreRegistry.address);
      deploy(contracts.CustomSimpleStore);
      deploy(contracts.AnotherCustomSimpleStore);
    });
  },
  config: {
    defaultAccount: 0,
    defaultGas: 3000000,
    environments: {
      testrpc: {
        provider: {
          type: 'http',
          host: 'http://localhost',
          port: 8545,
        },
      },
      objects: {
        CustomSimpleStore: {
          class: 'SimpleStore',
          from: 2, // a custom account
```

INPUT PROPERTIES

```
module.exports = {
  output: {
  },
  entry: {
  },
  module: {
  },
  config: {
  }
}
```

Enforced Contract Testing in Two Languages

```
import "dapple/test.sol";
import "SimpleStore.sol";

contract SimpleStoreTest is Test {
    SimpleStore target;

    function refreshTargetInstance() {
        target = new SimpleStore();
    }

    function setUp() {
        refreshTargetInstance();
    }

    function testSetMethod() {
        uint256 testValue = 45;
        target.set(testValue);
        assertEq(target.get(), testValue);
    }
}
```

DAPPLE (Solidity)

```
const contracts = require('../build/classes.json');
const chaithereum = require('chaithereum');

before(() => chaithereum.promise);

describe('SimpleStore', () => {
    let simpleStore;

    it('successfully instantiates with blank params', () => {
        return
        chaithereum.web3.eth.contract(JSON.parse(contracts.SimpleStore.interface)).new.q({ data: contracts.SimpleStore.bytecode
    }).should.eventually.be.contract.then((_simpleStore) => {
        simpleStore = _simpleStore;
    }).should.eventually.be.fulfilled;
    });
});
```

CHAITHEREUM (Javascript)

Enforced Contract Testing in Two Languages

```
nick@nick-PEAQ-PNB-S1013-ISA1-ULTRA: ~/github/react-dapp-boilerplate

DAPPLE TESTING STARTED:

Testing...
/bin/sh: 1: solc: not found
No local solc found. Failing over to JS compiler...

SimpleStoreFactoryTest
  test factory
    Passed!

SimpleStoreRegistryTest
  test registry
    Passed!

SimpleStoreTest
  test set method
    Passed!

Summary
  Passed all 3 tests!

> react-boilerplate@3.1.0 test /home/nick/github/react-dapp-boilerplate
> npm run test:testrpc && cross-env NODE_ENV=test karma start internals/testing/karma.conf.js --single-run

> react-boilerplate@3.1.0 test:testrpc /home/nick/github/react-dapp-boilerplate
> mocha ./app/contracts/tests

SimpleStore
  ✓ successfully instantiates with blank params (61ms)

SimpleStoreFactory
  ✓ successfully instantiates with blank params (38ms)
```

The Defensive dApp

ATTACK VECTORS

- XSS (TPD)
- Spoofing (TPD)
- Data Flooding (TPD)
- Memory Leaks
- Improper Wallet Creation/Handling
- Github Access (2-Factor Please)

And many more to come =D

DEFENCE

- "xss": "^0.2.13"
- Communication Channels/Registries
- Size Identification (IPFS -> BlockSize)
- Web Workers
- Web Worker or Don't Do It (Usually)
- Enforce (2-FA) Project Wide.. good

password management.. =D

Follow the UNIX philosophy

Composable atomic parts making up a complex whole

Coming soon...

github.com/silenticicero/react-dapp-boilerplate

B/c: git commit... some kind of wall... ;-)

CONTACT

Founder, WeiFund

@WeiFund (crowdfunding)

weifund.io

Founder, BoardRoom

@GoBoardRoom (governance)

boardroom.to

Nick Dodson

@IAmNickDodson

nick.dodson@consensys.net

