Data Entry DApp

Project Root directory

<DIR> ..

<DIR> build

<DIR> contracts

<DIR> migrations

<DIR> node\_modules

<DIR> src

<DIR> test

<DIR> www

package-lock.json

package.json

truffle-config.js

truffle.js

webpack.config.js

Unit Tests using Truffle and Testrpc

testrpc -p 8700

truffle migrate

truffle test

Install Python

For now, install python 2.7 and pip. Include python and pip to you environment PATH

(e.g. ;C:\dev\Python27;C:\dev\Python27\Scripts)

Install Nodes.js and NPM

npm install eth-ens-namehash -g

npm install web3 -g or npm install web3 -S

npm install webpack -g

Install and Run your DApp

From project Root directory:

npm install

npm run dev

Transactions using Testrpc

Get the list of available accounts from Testrpc:

Available Accounts

==================

(0) 0x69bc681fa9b90dd59147044bb831cf0e3649440b

(1) 0xab7fbd5a3d6366649caf5fef5ed3abacd5907f04

(2) 0x4954867119ca2477f313d53319ce091bee615c76

(3) 0x105d68a8db49759f8a5fc22e4ac26b77d24437dc

(4) 0x5082c43f7b5167e63e9ce5cb71328282bbfd3586

(5) 0x2ddd2bac7cb30b3cba12ba06a976614597026fbf

(6) 0x51fbca896ff41e7d2e92da863b35b2f6f0eb4896

(7) 0x1e59657ad8d81b9e0fb2e575b0de9aa1c278af04

(8) 0xdebb60de22f6f2ceaba9addca27ba19d85d8abcc

(9) 0x788a06a29db4106a858b4a9e2d5484051787b19a

Note that there is no need to personal.unlockAccount(…) when using testrpc as your Ethereum client. See src/dapp.js

window.addEventListener("load", function() {

if (typeof web3 !== "undefined") {

window.web3 = new Web3(web3.currentProvider);

} else {

// using geth 8545

//window.web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

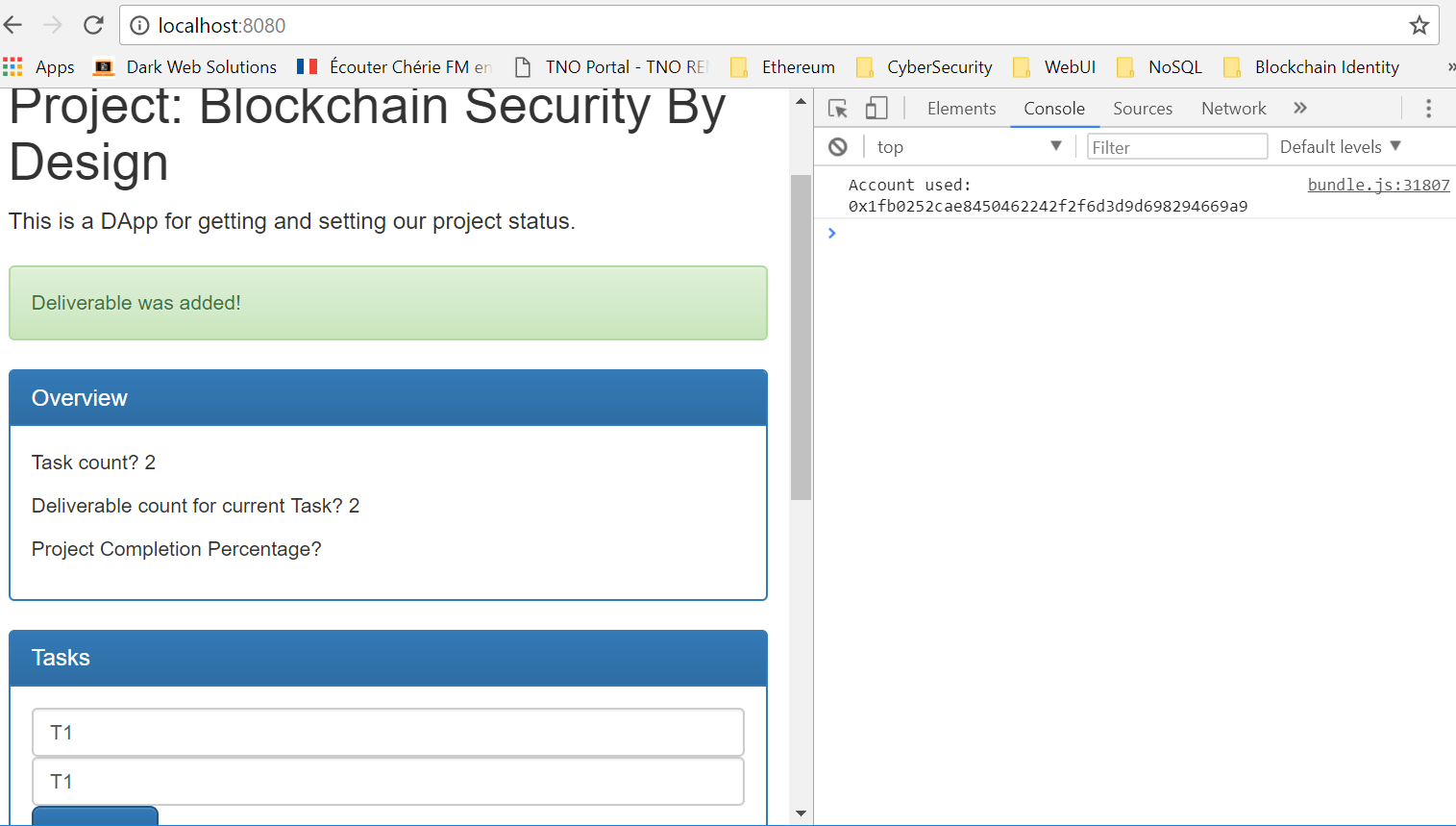
// using testrpc -p 8700

window.web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8700"));

}

From web brower:

DApp <http://localhost:8080/>



Event Monitoring DApp

Project Root directory

DIR> ..

<DIR> src

<DIR> www

package-lock.json

package.json

truffle.js

webpack.config.js

Open truffle.js, this project references the same contracts from above project’s build directory.

It also connects to the same Ethereum client port to access the contract(s).

module.exports = {

contracts\_build\_directory: "..\tnoproject\build",

networks: {

development: {

host: "localhost",

port: 8700, // run testrpc -p 8700, default was 8545

Check src/dapp.js

var TNOProject = contract(require("../../tnoproject/build/contracts/TNOProject.json"));

Deploy Event Monitoring DApp to specific Host and Port

Check package.json,

"scripts": {

"build": "webpack",

"dev": "webpack-dev-server --host localhost --port 8081"

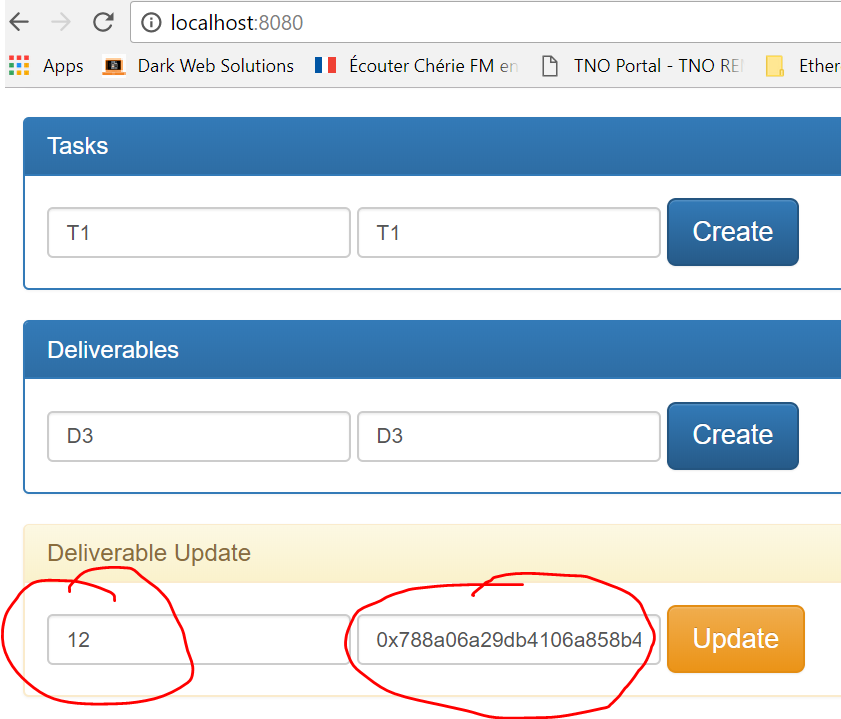
},

Note that this DApp will use port 8081

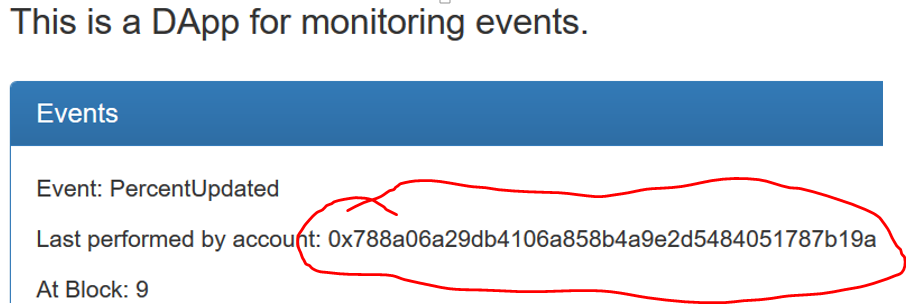
Running and Monitoring Event

Updates from DApp <http://localhost:8080/> will trigger contract Events.

Shown below, enter percentage update and the Account used for this update. (Please refer to the list of Available Accounts in Testrpc)



DApp Events <http://localhost:8081/>



Known Issues

Problem

<contract> has not been deployed to detected network

Solution

Run truffle migrate again

Problem

truffle migrate Exceeds block gas limit

Solution

https://stackoverflow.com/questions/42924634/unable-to-run-truffle2-1-2-migrate-network-live-exceeds-block-gas-limit

Add to truffle.js

gas: 3000000, // use web3.eth.getBlock("pending").gasLimit >>4712388

Problem

<contract> has not been deployed to detected network

Solution

Run truffle migrate again

References

<https://github.com/dickolsson/irondoers>

<https://dickolsson.com/building-dapps-on-ethereum-part-3-user-interface/>

<https://github.com/angular/angular-cli/issues/3765>

<https://stackoverflow.com/questions/33272967/how-to-make-the-webpack-dev-server-run-on-port-80-and-on-0-0-0-0-to-make-it-publ>