目录

[1 Write a Store Factory (no Proxy) 2](#_Toc3699)

[1coding 2](#_Toc10674)

[Step 1 start ganache 2](#_Toc27788)

[2 Manual test 3](#_Toc10603)

[2 make privateStamp factory upgradeable 4](#_Toc22781)

[2.1Verify old logic 4](#_Toc29800)

[2.2 Upgrade to v2 4](#_Toc19890)

[3 how to integrate ui to api 5](#_Toc356)

[Step 1: go to ui and add proxy config to package.json 5](#_Toc8464)

[Step2: link node with react front end 5](#_Toc25975)

[Step 3, start nodejs 5](#_Toc29421)

[What is the different between ui integrates api and ui +api? 6](#_Toc11194)

[4 Webpack 6](#_Toc18994)

[4 Production: Dockernize the app 7](#_Toc3805)

[5 How to set domain for a website 7](#_Toc12435)

# 1 Write a Store Factory (no Proxy)

Github.com/onebit256/proxy

## 1coding

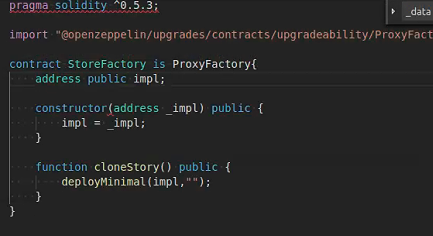
## Step 1 start ganache

Ganache-cli -m ‘’

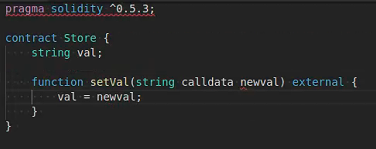
Truffle init

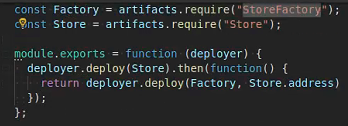
Npm init

Npm i @openzeppelin/upgrades@2.8.0



As there is no init function





If there are init data



## 2 Manual test

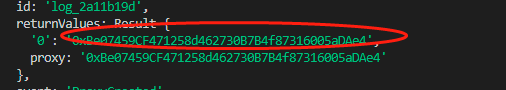
Const sf = await StoreFactory.deployed()

Sf.impl()

Networks

Sf.clonestore()

Sf.getPastEvents()



Const St1 = await Store.at(‘0xBe07459CF471258d462730B7B4f87316005aDAe4’)

St1.setVal(“ss”)

St1.val()

Sf.clonestore()

Sf.getPastEvents()

Const St2 = await Store.at(‘0xBe07459CF471258d462730B7B4f87316005aDAe4’)

St2.setVal(“ss”)

St2.val()

Sf.clonestore()

# 2 make privateStamp factory upgradeable

## 2.1Verify old logic

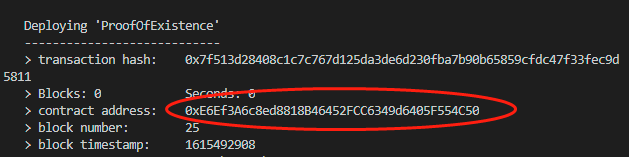
const proxyFactory =await ProxyFactory.at(ProxyFactory.address)

const proxyAdmin =await ProxyAdmin.at(ProxyAdmin.address)

const salt = "20";

const deploymentProxyAddress = await proxyFactory.getDeploymentAddress(salt, accounts[0]);

const implementationAddress = await proxyAdmin.getProxyImplementation(deploymentProxyAddress);





const poe1 =await ProofOfExistence.at(deploymentProxyAddress)

poe1.storeHash("123")

poe1.verifyHash(accounts[0],"123")



## 2.2 Upgrade to v2

await proxyAdmin.upgrade(deploymentProxyAddress, ProofOfExistenceV2.address)

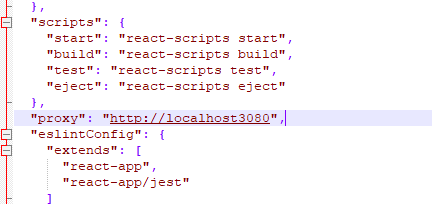


const poe2 = await ProofOfExistenceV2.at(deploymentProxyAddress)

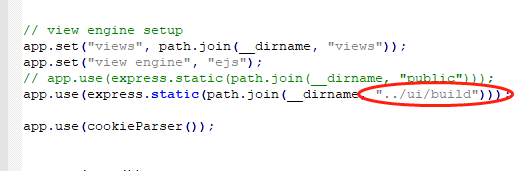


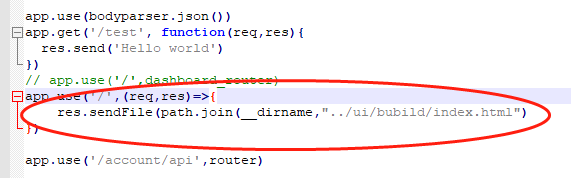
# 3 how to integrate ui to api

## Step 1: go to ui and add proxy config to package.json

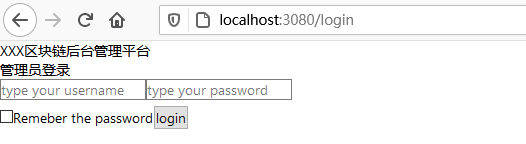


## Step2: link node with react front end



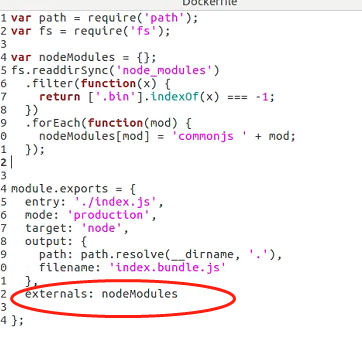


## Step 3, start nodejs

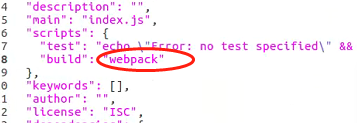


## What is the different between ui integrates api and ui +api?

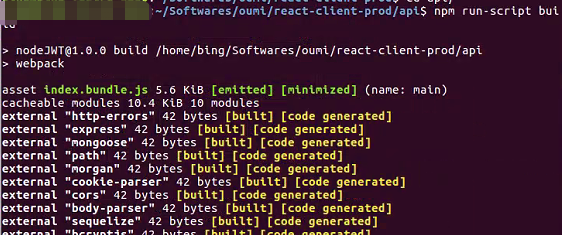
# 4 Webpack



Exclude node\_modules folder



Add build script to package.json



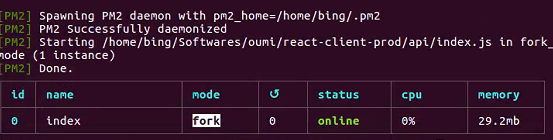
Npm run-script build

Then node index.bundle.js

# 5 pm2 host node application

npm install pm2 -g

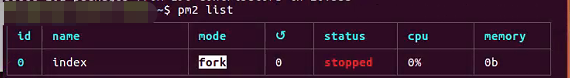
Pm2 start index.js



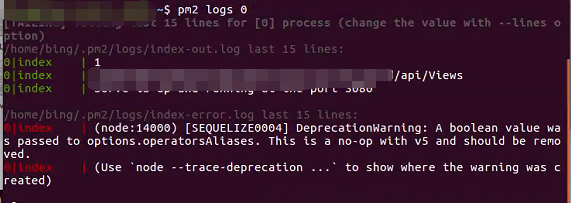
Pm2 stop 0



Pm2 list

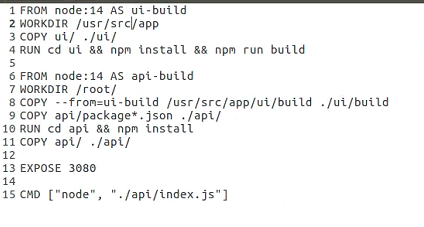


Pm2 logs 0



# 6 Production: Dockernize the app

Docker file



- copy --from-ui-build /usr/xxx ./ui/build

Copy directly from one image to another image

Docker compose



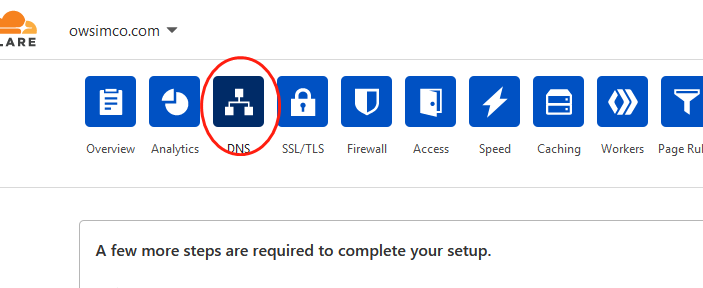
Note:

Map the database.sqlite to docker container

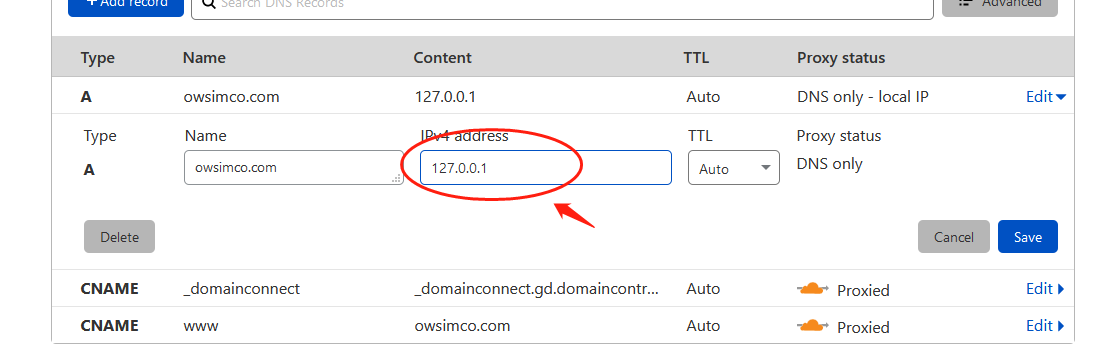
# 7 How to set domain for a website

After you buy a domain name from a dns provider, such as godaddy aws or google

1 go to your dns provider, in our case cloudflare, then click DNS

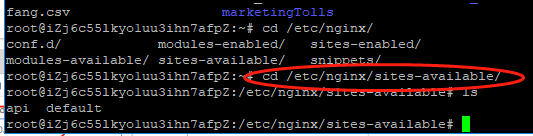


2 replace 127.0.0.1 with your read ip, and click save

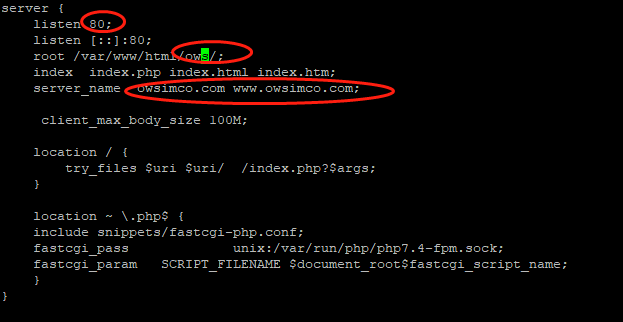


3 install nginx: sudo apt-get install nginx

4 go to /etc/nginx/sites-avaliable folder



5 create ows.conf



Put index.html to /var/www/html/ows

Set the domain name to server\_name

6 ln -s /etc/nginx/sites-avaliable/ows.conf /etc/nginx/sites-enable/

7 create a hello world index.html in ows

8 change owner to www-data



9 type owsimco.com

