

Mac下安装truffle 并运行truffle-react脚手架

前言:nodejs环境的管理(版本)

安装nvm(可以用于切换nodejs和npm版本,类似python的venv)

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.8/install.sh  
| bash
```

然后查看nvm,并安装和使用视频中推荐的node版本(6.11.3)

```
nvm -v
```

```
nvm install 6.11.3
```

```
nvm use 6.11.3
```

Ps:nvm帮助我们在不同的nodejs版本里集成了不同的npm版本.所以不用关心npm.

一.安装truffle过程:

首先查看系统中是否有truffle:

```
truffle version
```

发现之前安装过,然后运行命令,删除掉它.

```
npm uninstall -g truffle (-g是全局拆卸)
```

接着重新安装指定的版本

```
npm install -g truffle@xxx
```

由于查看虚拟机中,课程用的是:

```
juicebox@juicebox:~$ truffle version
Truffle v3.4.9 (core: 3.4.8)
Solidity v0.4.15 (solc-js)
juicebox@juicebox:~$ npm -v
3.10.10
juicebox@juicebox:~$ node -v
v6.11.3
juicebox@juicebox:~$
```

于是:

```
npm install -g truffle@3.4.9
```

安装完整后,查看版本号:

```
truffle version
```

二.安装脚手架:(安装好`truffle`之后)

```
truffle unbox react
```

三.配置&运行

我们用的是官网推荐的react版本的truffle脚手架.打开其文档:

<http://truffleframework.com/boxes/react>

可以看到如下:

TRUFFLE

Installation

1. Install Truffle globally.

npm install -g truffle

2. Download the box. This also takes care of installing the necessary dependencies.

truffle unbox react

3. Run the development console.

truffle develop

4. Compile and migrate the smart contracts. Note inside the development console we don't preface commands with `truffle`.

compile
migrate

5. Run the webpack server for front-end hot reloading (outside the development console). Smart contract changes must be manually recompiled and migrated.

// Serves the front-end on http://localhost:3000
npm run start

6. Truffle can run tests written in Solidity or JavaScript against your smart contracts. Note the command varies slightly if

使用命令发现:

使用命令`truffle compile`

会提示没有找到rpc接口.嗯,大家可以用npm安装testrpc或者推荐 truffle官网出的 ganache 可以代替testrpc..地址:<http://truffleframework.com/ganache/>

```
buliang@172 ~$ npm install -g truffle
hereumJS TestRPC v4.1.3 (ganache-core: 1.1.3)

Available Accounts
=====
  > 0x8f6711147a65d1a3e47b225e9c1607fb659fdbad
  > 0x8c97e1714b4bad316d8cee5817ba8bc356507b
  > 0x8792100ef94cc08922ca48d1dfcc9a196257641
  > 0x9214f9c69f570468faad16d2af60f9a989f6c8e
  > 0xe5ab00b9589ac98ebbbe27b4feafef7f217a380
  > 0x75f145ab97322c339a67b8a0da747393d5a878ec
  > 0x284f1b491d983d70606563a62c649c9eacae446
  > 0x2cf6e88c8b1664c4986ce481d1d781652ecfad2c
  > 0xc4d26e19f6f2a3817c2696507ba9eaf3caabca47
  > 0x72ce29ebc1e76d12fb71afec7cbad904ecfb4

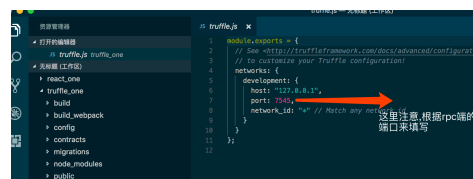
Private Keys
=====
  > d6e62cf0d1dd5af021d3796ef64eccc88b25215c402e1d07a48886bea17a27
  > 73ce5ba1eb482ab13a661e10b09ea7ccb47e55cf8ead38c7886565811565dd7e
  > 379a4326fa642f5d124dae210664aaacf6bd2fd3fa031bc34664661abb17b0
  > 0b247a22576f275ae4b5087e399023c88dffc6c8a3ebc8f92ce26d9066a9e9
  > 0b1721664ca316322db9e17a89902204c4f6dd1445d1fdaab36dbb3eb2a98a
  > 14dbbb7a3e6f492a40bbcc6f94ee59edf80eb1f6915efffa87d5180e908004
  > 5837aec2c7ccf9a5ad8c6f331a0bd94b20a18766703ef7322086ace6f34c6bb
  > 8b9223bf2dd009c0830bcad1351e06664a05a6945422cc739c2f0f9e433f48
  > ac14496a44234f29a836bb6939be502ffd3d51ea6e4f4dcf679f96b2bc38d1
  > 12d450d6cd28f4a8b1fe259656ae98ed63e3b7f3f9ee79b99a0a83420b6ca5ec

Wallet
=====
emonic:  alone suspect know angle parent canal spin heart cube often
yle
se HD Path:  m/44'/60'/0'/0/{account_index}

Listening on localhost:8645
```

然后设置和修改代码中的rpc接口配置:

truffle.js中的配置



```
truffle.js
1 module.exports = {
2   // See http://truffleframework.com/docs/advanced/configuration
3   // to customize your truffle configuration
4   networks: {
5     development: {
6       host: '127.0.0.1',
7       port: 7545,
8       network_id: '*' // Match any network id
9     }
10  }
11 }
```

utils/getWeb3.js中也有rpc的ip和端口的配置:

```
// utils.js
// Fallback to localhost if no web3 injection. We've configured this to
// use the development console's port by default.
var provider = new Web3.providers.HttpProvider('HTTP://127.0.0.1:7545')

Web3 = new Web3(provider)
```

再运行truffle compile ,随后运行truffle migrate

最后一步:

之后注意了!不要直接运行npm run start

.这时候需要运行npm i 会根据package.json的内容,在当前目录下安装必备的依赖库.

这时候,再运行npm run start就可以了,在浏览器中打开测试页面了.