### 1、安装工具和准备工作

下载安装nodejs18

https://nodejs.org/dist/v18.15.0/node-v18.15.0-x64.msi

安装yarn和hardhat：

**npm install yarn -g**

**npm install hardhat -g**

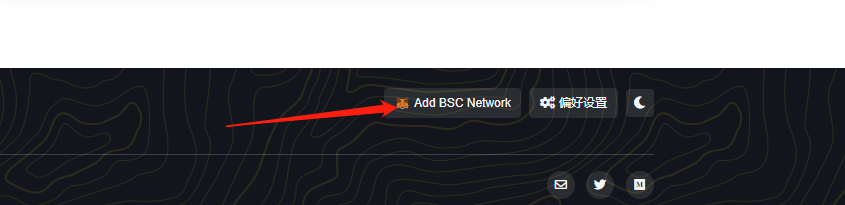
部署过程中提示缺少什么再安装什么

谷歌应用商店安装metamask

<https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn>

之后打开币安链测试网，为metamask添加网络

<https://testnet.bscscan.com/>



导入以下钱包到metamask：

**地址: "0x19aF2DA194998563E534fc8fF46AcCC6a237F3Aa"**

**私钥: "24ec8f2c2b21af6ef458c6232dc223ac9aea88ebfc7ca37c6d2e5bf9a5fdbf51"**

**助记词: "anxiety heart brain trumpet whale tip machine theory suspect awkward success buzz"**

### 2、编译、部署合约

进入contract目录

查看hardhat.config.js配置，里面有一些相关信息

执行命令：**yarn**

执行编译命令：**npx hardhat compile**，成功后会提示编译的数量

部署工厂、WETH、Multicall、MockERC20合约：**npx hardhat run scripts/deploy1.js**，执行后需要等待几十秒，得到所有合约地址，**备份它**：

-----------------合约部署结果------------------

工厂合约: 0x5bcFC70E36C384c4D1fe425b367A0D55a4fd52Ad

INIT\_CODE\_PAIR\_HASH: 0xe768ab8afc95f274319dac75e48584815e4510db521416621d643a8df9a08dc9

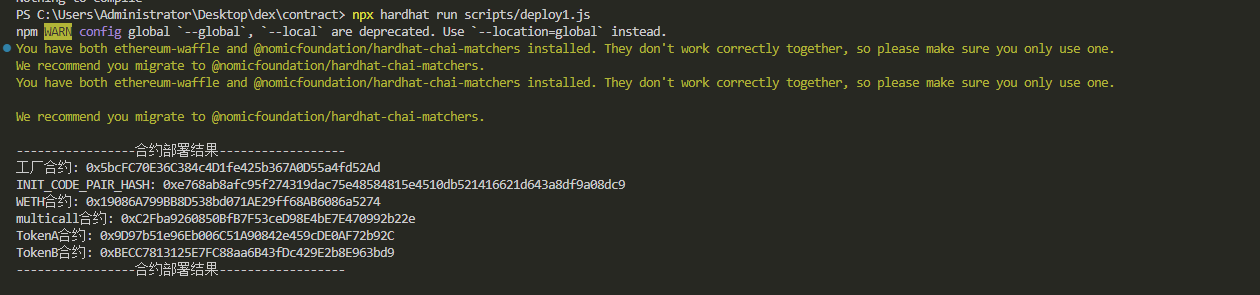
WETH合约: 0x19086A799BB8D538bd071AE29ff68AB6086a5274

multicall合约: 0xC2Fba9260850BfB7F53ceD98E4bE7E470992b22e

TokenA合约: 0x9D97b51e96Eb006C51A90842e459cDE0AF72b92C

TokenB合约: 0xBECC7813125E7FC88aa6B43fDc429E2b8E963bd9

-----------------合约部署结果------------------

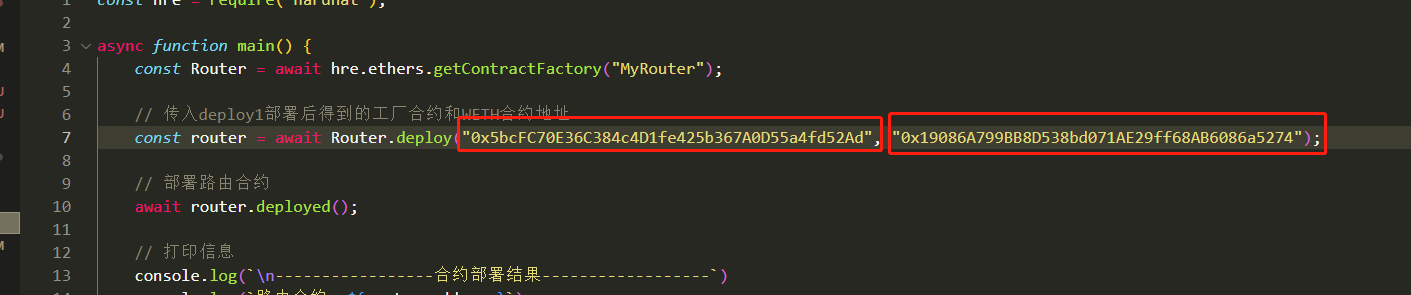


**修改contracts/MyRouter.sol路由合约文件init code hash（697行）：将单引号内的字符替换为上一步得到的INIT\_CODE\_PAIR\_HASH**：e768ab8afc95f274319dac75e48584815e4510db521416621d643a8df9a08dc9（注意**去掉0x前缀**），记得保存



再次编译合约：**npx hardhat compile**，提示成功编译1个

修改scripts/deploy2.js文件第7行，填入2个参数，分别是上一步得到的**工厂合约地址**和**WETH合约地址**，记得保存

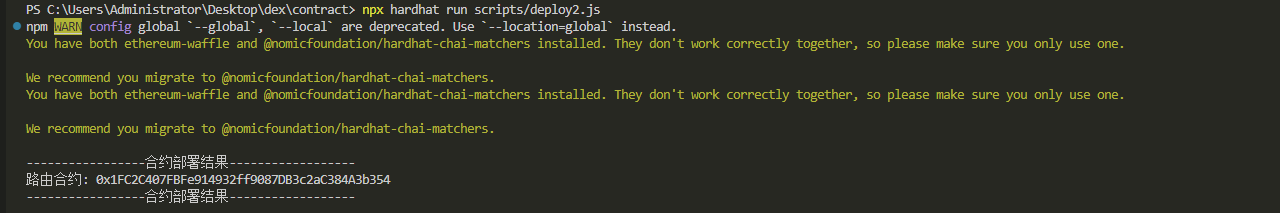


部署路由合约：**npx hardhat run scripts/deploy2.js**，需要等待几十秒，得到路由合约地址，**备份它**：

-----------------合约部署结果------------------

路由合约: 0x1FC2C407FBFe914932ff9087DB3c2aC384A3b354

-----------------合约部署结果------------------



### 3、注册npmjs账号

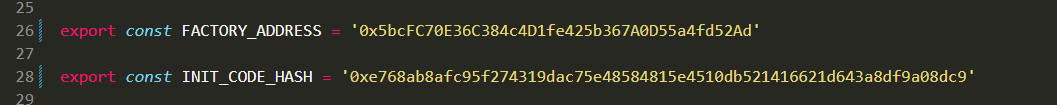
地址：https://www.npmjs.com/signup

### 4、发布dex-sdk包

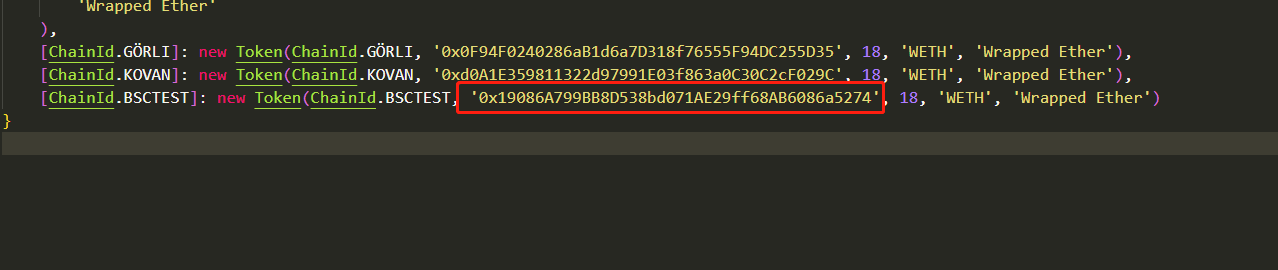
进入dex-sdk目录

执行命令yarn

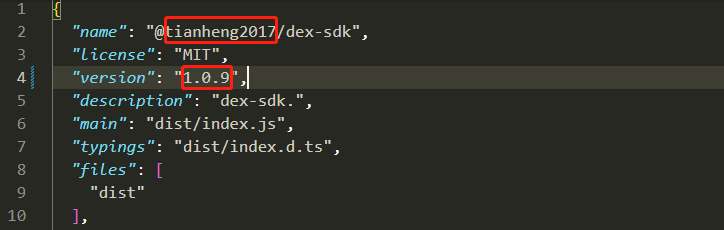
修改src/constants.ts的FACTORY\_ADDRESS和INIT\_CODE\_HASH为上一步得到的工厂合约与INIT\_CODE\_PAIR\_HASH值，保存



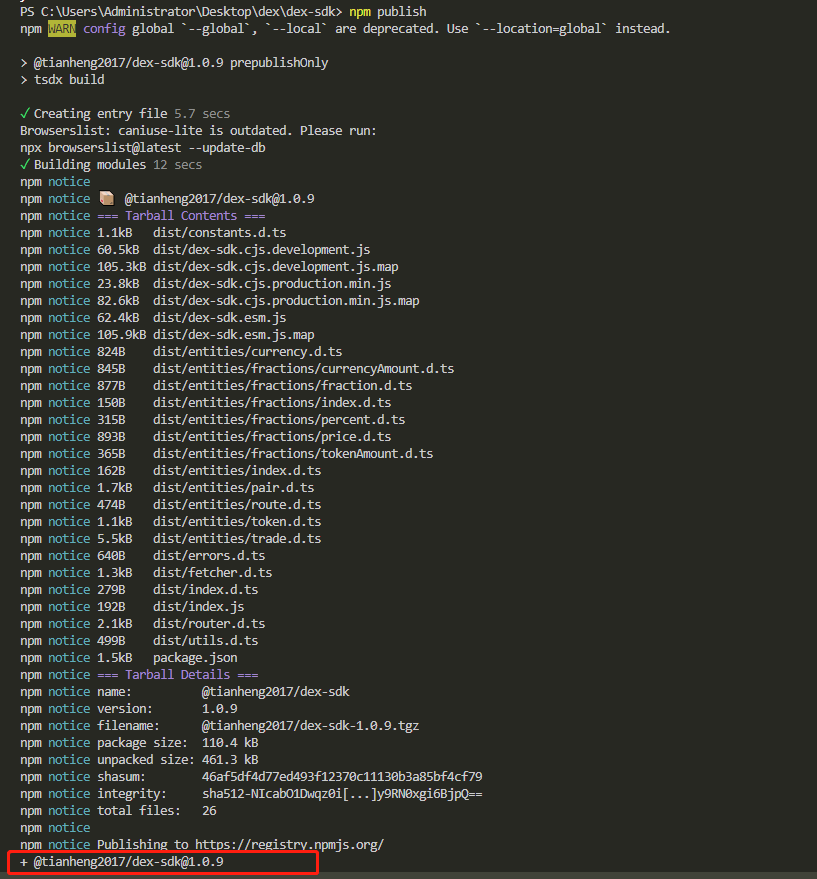
修改src/entities/token.ts的BSC的WETH地址为上一步得到的WETH合约地址



修改package.json的name和version，将name中的tianheng2017替换为为您的npmjs账号，修改version为您的版本号，新账号可以从1.0.0开始，之后每次发布包需要**递增**

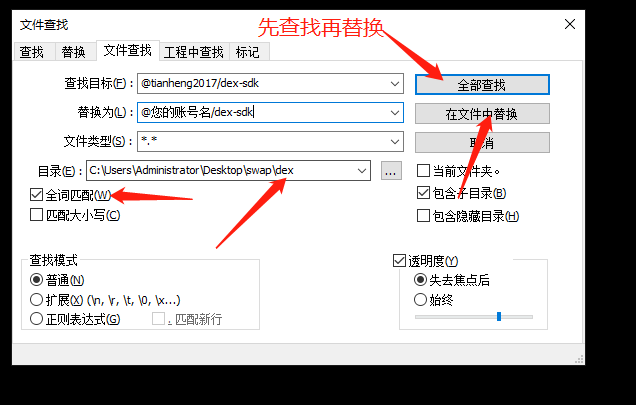


发布npm包：**npm publish**

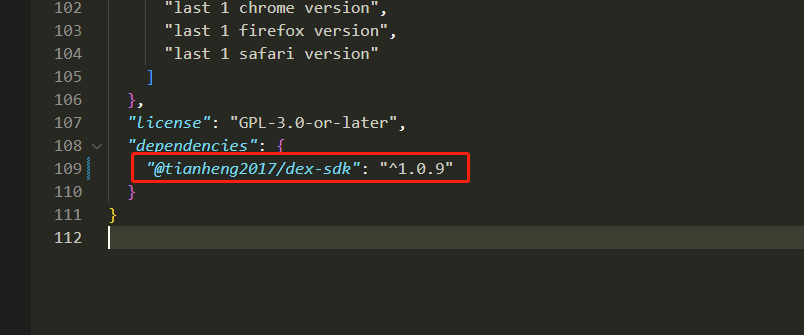


### 5、替换dex目录的所有npm包名

下载notepad++，进入notepad++，ctrl+h弹出替换界面，限定路径为dex目录，全局替换@tianheng2017/dex-sdk为你的npm信息，比如@xxxx/dex-sdk



进入dex目录，修改package.json，"@你的npx信息/dex-sdk": "^您的版本号"，如"@tianheng2017/dex-sdk": "^1.0.9"，保存

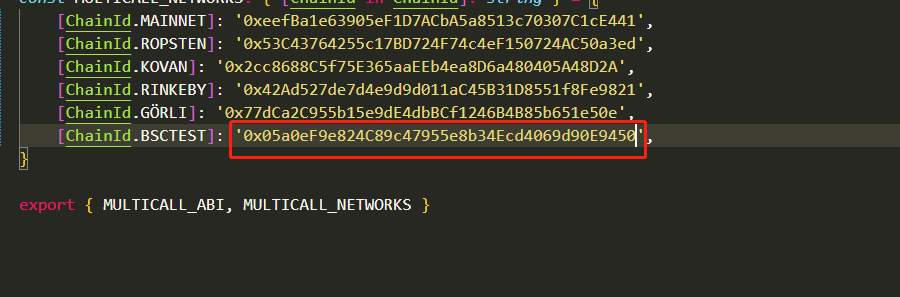


6、运行前端

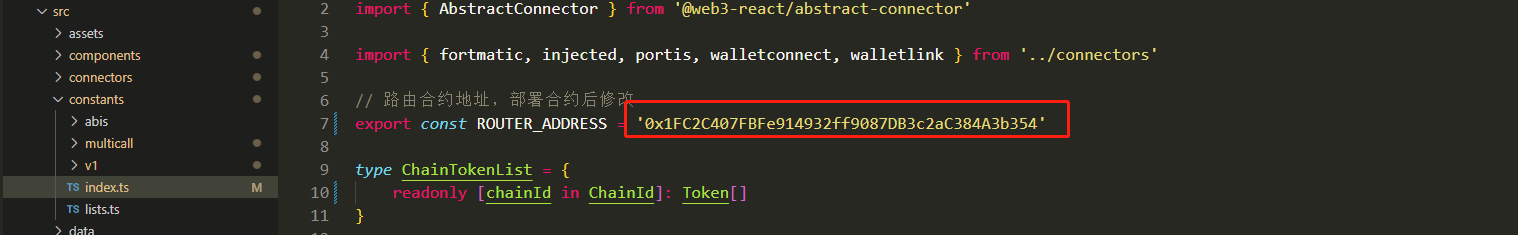
进入dex目录

执行命令yarn

修改文件multicall：dex\src\constants\multicall\index.ts，为上一步得到的multicall地址，保存

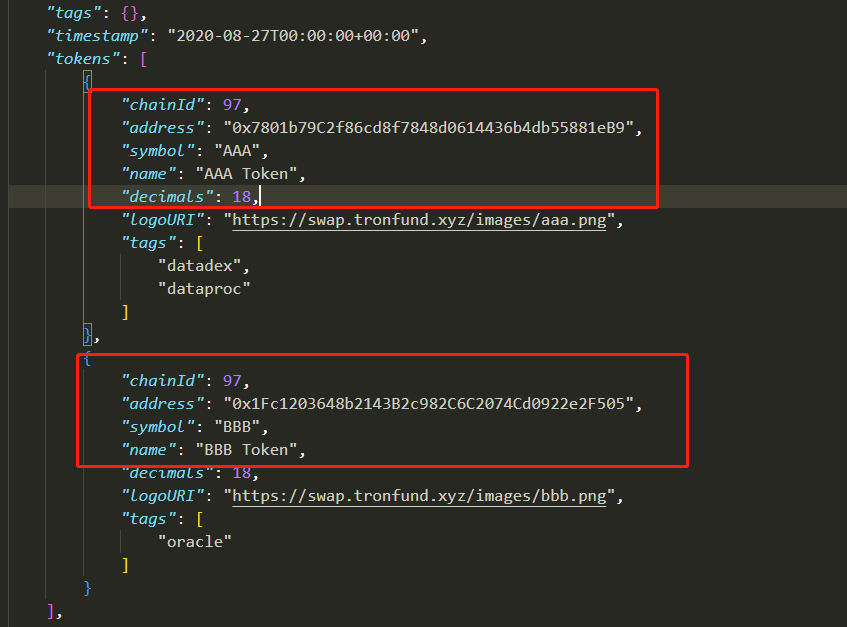


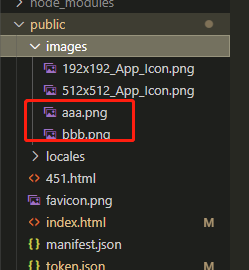
修改文件dex\src\constants\index.ts，将路由合约修改为上一步得到的路由合约地址，保存



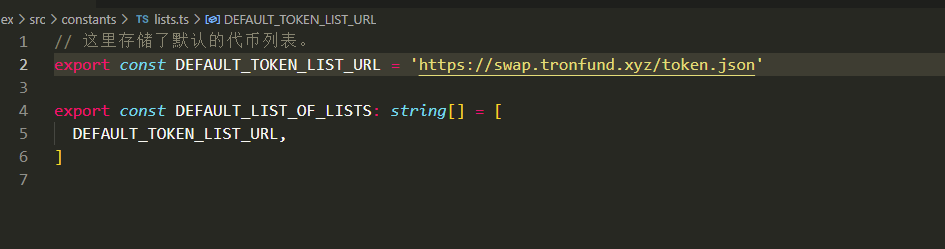
修改public/token.json，分别是上一步得到的AAA和BBB代币信息

且图标放到dex\public\images下





修改dex\src\constants\lists.ts的json地址指向token.json(我这里是线上web)，里面包含了AAA和BBB，这里储存了默认的代币列表，如图



最后执行本地运行命令：yarn start

或者打包后到传到web服务器：yarn build

等待，成功，如果不显示金额，可能是和BSC节点不通，国内需要翻墙

### 添加流动性和兑换测试

