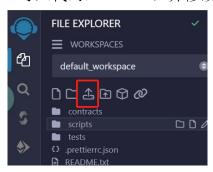
-, EVM

- 1.访问在线编译器: https://remix.ethereum.org/
- 2.导入代码: ENT.sol, 并修改配置



```
uint256 private _totalSupply;
uint256 private MAXSUP;
uint256 constant MAXIMUMSUPPLY::1000000000000 *10**18;

// 代币全称和符号
constructor(address initialOwner) ERC20("ENT coin", "ENT") Ownable(initial)
```

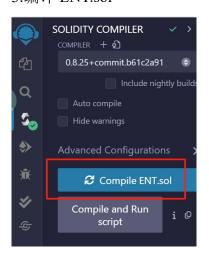
1)数字① * 10 数字② ** 数字③

数字①:发行金额

数字②:单位,默认不变

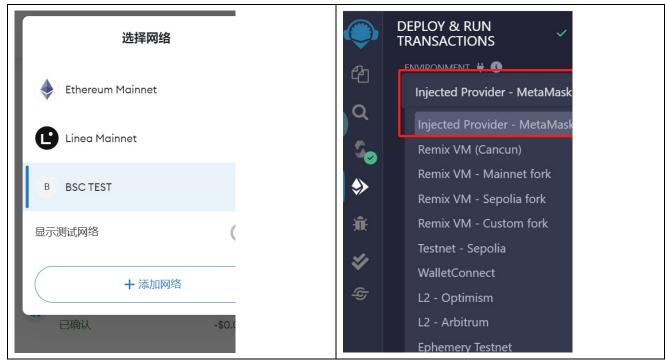
数字③: 精度, ETH 部署时修改为 6, BSC 部署时修改为 18

- 2) 核对最大发行量 MAXIMUMSUPPLY
- 3.编译 ENT.sol

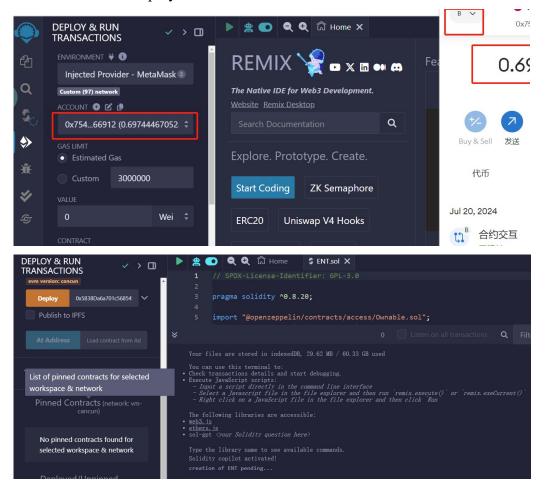


4.部署

在 matemask 中选择要部署的网络(ETH 活 BSC)后,在部署页面选择 Injected Proivder -MateMask

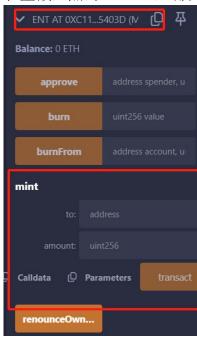


核对**部署合约的账户地址**和账户余额,确认后,将**部署合约地址的账户地**址输入进 Deploy 旁 边的框中,点击 Deploy。确认交互后,即部署成功

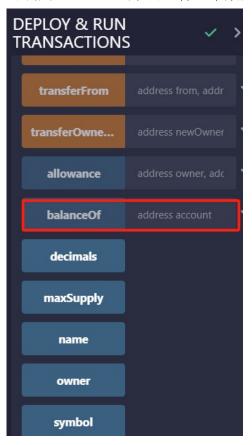


5.造币

点击左下方生成的合约部署地址,可出现所有合约可调用的方法。点击 Mint,输入目标地址和金额,点击 transact,确认交互后即可完成造币。



可调用 balanceOf 方法查看造币数量。



二、TON

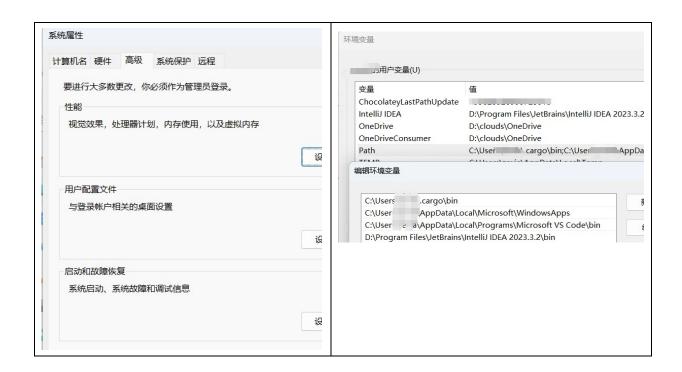
1. 环境准备

(1) node.js 下载安装

https://nodejs.org/zh-cn



安装成功后,配置环境变量node安装目录 到path



(2) 安装yarn

在CMD命令行程序中,定位ton工程目录

npm install -g yarn

D:\clouds\OneDrive\我的备份\00. 我的电脑\Desktop\发币文档\token\ton发币>npm :
changed 1 package in 24s

查看 yarn是否正确安装

yarn -v

D:\clouds\OneDrive\我的备份\00. 我的电脑\Desktop\发币文档\token\ton发1.22.22

(3) 初始化代码

yarn install

```
D:\clouds\OneDrive\我的备份\00. 我的电脑\Desktop\发币文档\token\ton发币>yarn install yarn install v1.22.22
warning package-lock.json found. Your project contains lock files generated by tools other than to mix package managers in order to avoid resolution inconsistencies caused by unsynchronized lo warning, remove package-lock.json.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning " > @dedust/sdk@0.8.6" has unmet peer dependency "@tonconnect/sdk@>=2.1.2".
warning " > @dedust/sdk@0.8.6" has unmet peer dependency "axios@>=1.5.0".
warning " > @tact-lang/emulator@3.2.2" has unmet peer dependency "ton-core@>=0.47.0".
warning " > ton-emulator@2.1.1" has unmet peer dependency "ton-core@>=0.47.0".
```

- 2. 修改发币配置信息(目录: ton/sources/deploy.ts)
 - (1) 部署网络切换
 - (2) mnemonics 输入要部署的密钥信息
 - (3) 修改代币信息: 名称name、描述description、符号symbol、图片image
 - (4) 修改最大供应量 max supply
 - (5) 修改给部署者铸币流量 supply

```
const client4 = new TonClient4({
     endpoint: "https://sandbox-v4.tonhubapi.com", // 测试网
     // endpoint: "https://mainnet-v4.tonhubapi.com",// 主网
 const mnemonics = "word word word"; // 助记词
 let keyPair = await mnemonicToPrivateKey(mnemonics.split(" ")); // 转密钥
 let secretKey = keyPair.secretKey; // 私钥
 let workchain = 0; //we are working in basechain.
 let deployer wallet = WalletContractV4.create({ workchain, publicKey: keyPai
 console.log(deployer_wallet.address);
 let deployer wallet_contract = client4.open(deployer wallet); // 打开钱包1
 //---- 调用合约固定信息, 每次设置后不要修改--
 const jettonParams = {
     name: "Entertainment Coins0713",
     description: "The governance token of the 4Arts platform",
     symbol: "ENT",
     image: "https://www.4arts.cc/home/ent-token-icon.png",
 let content = buildOnchainMetadata(jettonParams);
let max supply = toNano(60000000000); // 设置最大供应量
 // 连接编译后的合约,传入创建合约的地址和初始值
 let init = await SampleJetton.init(deployer_wallet_contract.address, content
```

3. 部署

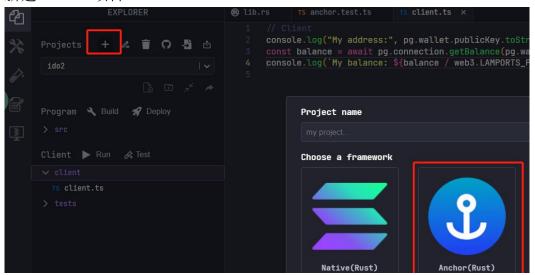
yarn deploy

```
D:\clouds\OneDrive\我的备份\00. 我的电脑\Desktop\发币文档\token\ton发币>yarn yarn run v1.22.22
$ ts-node ./sources/scripts/deploy.ts
EQAsndRoSnjB_eMnGHhuYRSU
```

可能出现的错误,timeout-5000。服务器请求超时,多次执行即可。

三、Solana

- 1. 访问在线编译器: https://beta.solpg.io/
- 2. 新建 Anchor 项目



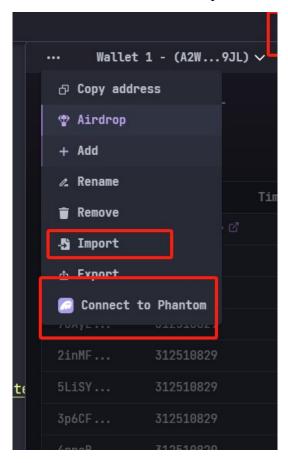
- 3. 环境配置
 - (1) 切换网络

点击左下角设置,可切换开发网络



(2) 账户设置

右上角 Wallet 中,可选择 import 或连接 Phantom 连接钱包。



4. 在 client 目录中新建 token.ts 文件,将目录中 token.ts 中的代码复制进去,右键文件运行

```
EXPLORER

♠ lib.rs

                                                            token.ts
                                                  } from "@socana/spl-token";
import { PublicKey } from "@solana/web3.js";
import { keypairIdentity, token, Metaplex } from "@metaplex-fo"
             + 4 🗑 🗘 🕹 🖒
Projects
                                                  const mintAuthority = pg.wallet.keypair;
                                                  const decimals = 9:
Program 🔌 Build 🚀 Deploy
                                                  let [tokenAccountOwnerPda] = PublicKey.findProgramAddressSync(
> src
                                                    [Buffer.from("token_account_owner_pda")],
                                                    pg.PROGRAM_ID
Client ▶ Run 🔏 Test
                                                  console.log("tokenAccountOwnerPda", tokenAccountOwnerPda.toStr
∨ client
                                                  const metaplex = new Metaplex(pg.connection).use(
  TS client.ts
                                                    keypairIdentity(pg.wallet.keypair)
  TS token.t
              A Rename
> tests
                                                  const createdSFT = await metaplex.nfts().createSft({
                          Del
              ■ Delete
                                                    uri: "https://shdw-drive.genesysgo.net/AzjHvXgqUJortnr5fXD62
name: "GDXS",
symbol: "GDXS",
                                                     sellerFeeBasisPoints: 100,
                                                     updateAuthority: mintAuthority,
```

- 5. 修改名字,符号,图标,最大供应量
- 6. Mint 铸币

四、其他问题

(1) .sol, .ts 文件可用电脑上的 txt 记事本打开。