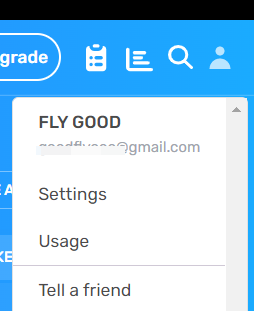
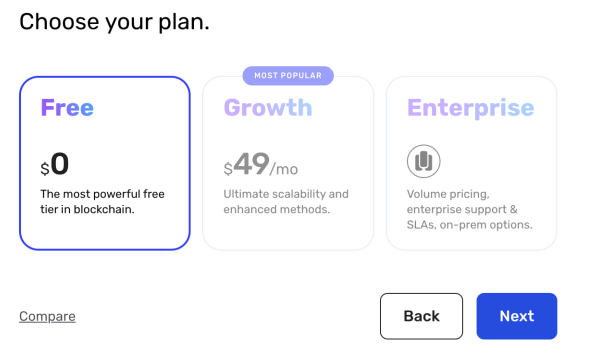
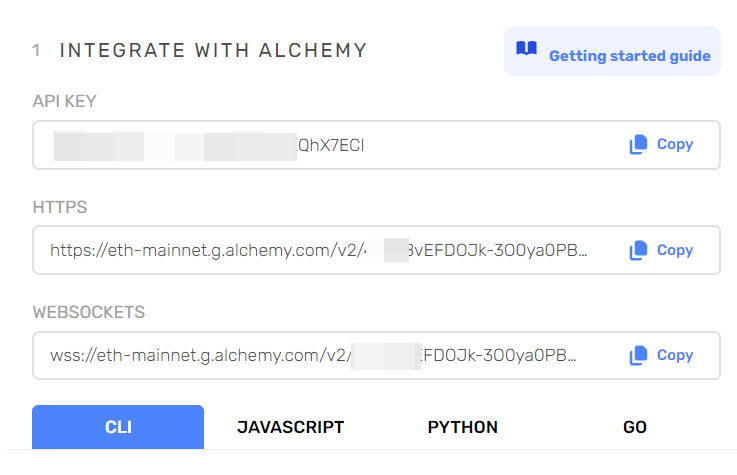
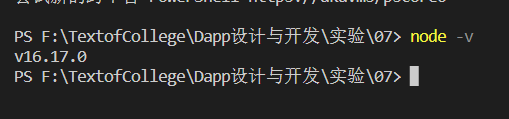
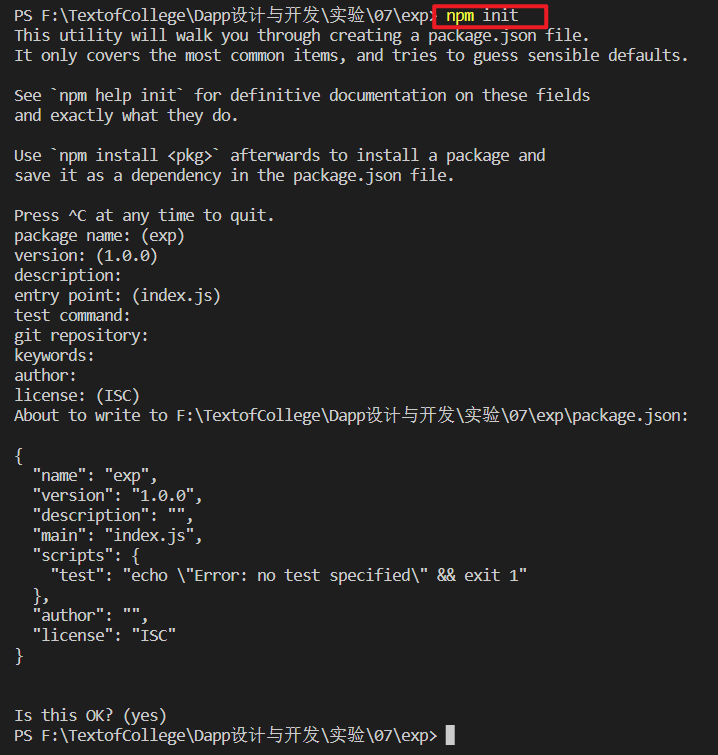
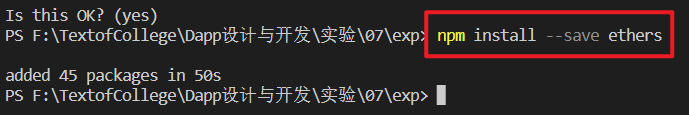
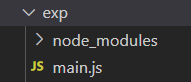
**周亚男 2020131062 202班**

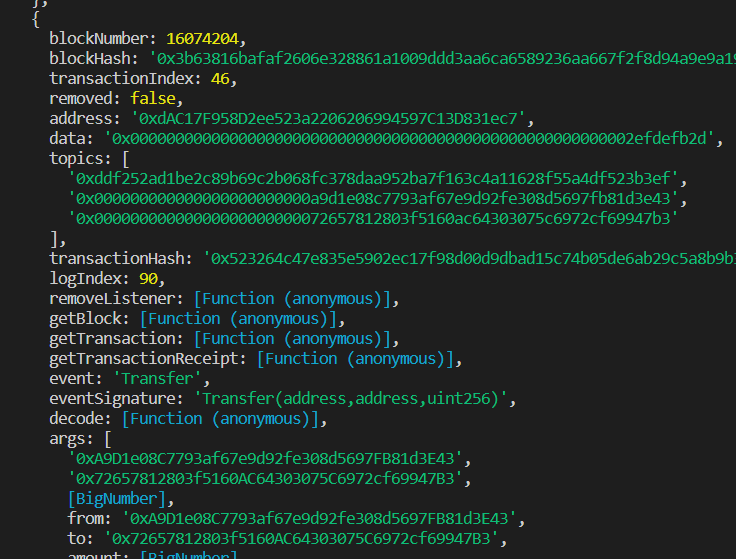
1. 注册alchemy账户



2. 选择免费计划  
  
3. 找到相应的key和Ethreum的配置，作为代码里RPC的配置  
  
4. 在本地安装node.js  
  
5. 在本地新建目录，并使用 npm init 进行初始化；  
  
6. 安装ethers.js  
  
7. 新建文件 main.js 在程序里实现以下功能：



·查询最新区块高度   
·查询gasPrice   
·根据区块高度查询区块的miner；   
·查询USDT合约的symbol   
·查询USDT合约的decimals   
·查询USDT合约的totalSupply   
·查询USDT合约最新一百个区块的Transfer事件；



加分项： 统计USDT的所有持币账户，分成address/amount 两个属性 生成一个JSON文件。并且和区块链浏览器数据做对比，账户 余额是否正确。

实验报告内容 提交源代码，并回答：   
1. 除了alchemy之外，你还知道哪些RPC Provider？Ethereum Nodes   
etherscan  
2. Provider的作用是什么？   
给用户一个接口，查询区块链数据  
3. ABI 的内容是啥？如何在程序里使用？

对合约编译之后生成的json格式的文件，里面有对abi具体变量函数的总结。

根据地址获取合约实例，然后调用合约方法

源码：

const ethers = require('ethers');

const { AbiCoder } = require('ethers/lib/utils');

const fs = require('fs');

// 查询ethers 官方文档，查询如何配置一个alchemy的provider

const apiKey = "4tIczBvEFDOJk-3O0ya0PBwbXQhX7ECl";

const provider = new ethers.providers.AlchemyProvider("homestead", apiKey);

// 生成合约实例

const address = "0xdAC17F958D2ee523a2206206994597C13D831ec7";

const abi = JSON.parse(fs.readFileSync("./ERC20.json"));

const contract = new ethers.Contract(address, abi, provider);

async function main() {

  // 获取链基本信息

  // const network = await provider.getNetwork();

  const blockNumber = await provider.getBlockNumber();

  const gasPrice = await provider.getGasPrice();

  const block = await provider.getBlock(10000);

  // console.log(network);

  console.log('block number', blockNumber);

  console.log('gas price', gasPrice.toString());

  console.log('block miner', block.miner);

  // 获取USDT合约基本信息

  const decimals = await contract.decimals();

  const totalSupply = await contract.totalSupply();

  const symbol = await contract.symbol();

  console.log('decimals', decimals);

  console.log('total supply', totalSupply.toString());

  console.log('symbol', symbol);

  // 获取从最新高度往前的100个区块里的Transfer事件

  const events = await contract.queryFilter("Transfer", blockNumber - 100, blockNumber);

  console.log('event length', events);

}

main();