IPFS踩坑总结：

IPFS在安装的时候，尽量不要选择为全体用户安装，否则再重新安装的过程中会出现错误。

IPFS在配置文件中有设置存储空间大小，默认是10GB，可以进行修改。

IPFS需要对跨域资源共享( CORS )进行配置。

配置命令：  
ipfs config --json API.HTTPHeaders.Access-Control-Allow-Methods "[\"PUT\", \"POST\", \"GET\", \"OPTIONS\"]"

ipfs config --json API.HTTPHeaders.Access-Control-Allow-Origin "[\"\*\"]"

ipfs config --json API.HTTPHeaders.Access-Control-Allow-Credentials "[\"true\"]"

ipfs config --json API.HTTPHeaders.Access-Control-Allow-Headers "[\"Authorization\"]"

ipfs config --json API.HTTPHeaders.Access-Control-Expose-Headers "[\"Location\"]"

<https://blog.csdn.net/a4636666/article/details/101377326>

然后目前使用的是IPFS Desktop 0.10.2版本

2020.3.26 当前进度为，开始进行IPFS和react交互功能，目前使用的是create-react-app 创建react项目，但是到目前为止，需要考虑的是如何使用IPFS-api功能，因此需要添加ipfs-api插件来完成。当前系统已经完成了配置，需要注意的是，ipfs-api安装有坑。在安装的过程中，需要安装到dev上，因此需要安装命令是：npm install –-save –-dev ipfs-api

在安装完ipfs-api插件之后，需要导入一些状态变量来进行配置。具体配置：

**Running the daemon with the right port**

To interact with the API, you need to have a local daemon running. It needs to be open on the right port. 5001 is the default, and is used in the examples below, but it can be set to whatever you need.

# Show the ipfs config API port to check it is correct

> ipfs config Addresses.API

/ip4/127.0.0.1/tcp/5001

# Set it if it does not match the above output

> ipfs config Addresses.API /ip4/127.0.0.1/tcp/5001

# Restart the daemon after changing the config

# Run the daemon

> ipfs daemon

将标红的命令复制粘贴到控制台即可

### Importing the module and usage

var ipfsAPI = require('ipfs-api')

// connect to ipfs daemon API server

var ipfs = ipfsAPI('localhost', '5001', {protocol: 'http'}) // leaving out the arguments will default to these values

// or connect with multiaddr

var ipfs = ipfsAPI('/ip4/127.0.0.1/tcp/5001')

// or using options

var ipfs = ipfsAPI({host: 'localhost', port: '5001', protocol: 'http'})

// or specifying a specific API path

var ipfs = ipfsAPI({host: '1.1.1.1', port: '80', 'api-path': '/ipfs/api/v0'})

将上面的内容复制粘贴在App.js中的开始（impot下面即可）。

以上三行的ipfs的定义是一致的，选择其一使用即可。第四个IPFS定义是特殊端口和网址的定义

<https://github.com/dn3010/js-ipfs-api>

在写IPFS时，function App 不是很好用，我换成了

class App extends Component { constructor(props) {super(props);this.state = { strHash: null, strContent: null} } saveTextBlobOnIpfs = (blob) => {return new Promise(function(resolve, reject) { const descBuffer = Buffer.from(blob, 'utf-8'); ipfs.add(descBuffer).then((response) => {console.log(response) resolve(response[0].hash); }).catch((err) => {console.error(err) reject(err); })})}

https://blog.csdn.net/weixin\_41160534/article/details/79867229