

IPFS音乐播放器

IPFS相关

IPFS第一次亲密接触

- 什么是IPFS
- IPFS对比HTTP/FTP等协议的优势
- IPFS应用场景

-移动数据 交易 路由 网络

- 定义数据 命名
- 使用数据

具体场景：

挂载全球文件

版本管理功能

数据库

加密平台

各种类型cdn

永久访问的链接

ipfs入门

- 官网地址：<https://ipfs.io>
- 下载安装：<https://dist.ipfs.io/#go-ipfs>
- 节点初始化
 - `ipfs init`
 - 如果出现 `Error: cannot acquire lock: can't lock file` 删除其后边给出的 `repo.lock` 文件即可
- 节点配置
 - `ipfs id` 查看当前节点id等信息
 - `ipfs config show ipfs` 配置信息
- 节点服务器daemon
 - `ipfs daemon`
- 修改IPFS默认路径
- 开放API请求
 - `ipfs config --json Addresses.API "/ip4/0.0.0.0/tcp/5001"`
- 开放公共网关
 - `ipfs config --json Addresses.Gateway "/ip4/0.0.0.0/tcp/8080"`

发布数据

- 上传文件

```
ipfs add haha.txt 文本
```

```
ipfs add cat.jpg 图片
```

```
ipfs add -q cat.jpg 只输出hash结果
```

hash记录返回的文件路径信息

- 上传/查看目录

添加目录: `ipfs add -r dir`

查看目录: `ipfs ls Qmej3u92BHJVkkiC8F5uL1J4a98no76TnXmQxHGXCHGE7t`

- 发布网站

把包含index.html的文件夹add到ipfs网络即可

获取数据

命令行获取 `cat/get`

```
ipfs cat QmaoahyA1UejtgPifPvjpsviePsDnJn8vHrYAN9nfF7w9N 查看文本
```

```
ipfs cat QmaoahyA1UejtgPifPvjpsviePsDnJn8vHrYAN9nfF7w9N > new-haha.txt 保存文本
```

```
ipfs cat Qmd286K6pohQcTKYqnS1YhWrCiS4gz7Xi34sdwMe9USZ7u > new-cat.jpg 保存图片
```

```
ipfs get QmZSwQghczB41omjfJipFPQ3FovYyLvXR9oa4Y9LQS7Urp -o tomcat.jpg 获取并保存文件
```

```
ipfs get QmZSwQghczB41omjfJipFPQ3FovYyLvXR9oa4Y9LQS7Urp -Cao tomcat-go 压缩并下载文件
```

```
ipfs get QmYtrQVXatZGm1WRKZ29vUt5tFa24xnTvvhv6D71DSNVttZ -o ipfs-day01 获取文件夹
```

浏览器访问获取

1. 在<http://localhost:5001/webui>下搜索hash值, 可以raw查看原数据, download下载文件

2. 在8080端口访问:

<http://localhost:8080/ipfs/Qmd286K6pohQcTKYqnS1YhWrCiS4gz7Xi34sdwMe9USZ7u>

3. 在ipfs.io访问:

<https://ipfs.io/ipfs/Qmd286K6pohQcTKYqnS1YhWrCiS4gz7Xi34sdwMe9USZ7u>

不需要翻墙的网关:

<https://ipfs.infura.io/ipfs/Qmd286K6pohQcTKYqnS1YhWrCiS4gz7Xi34sdwMe9USZ7u>

文本数据

图片数据

音频数据

files文件/文件夹操作

```
ipfs files cp <文件/文件夹的hash> <目标文件/文件夹> 拷贝文件
```

```
ipfs files ls -l 查看目录
```

ipfs files mkdir 创建目录

ipfs files cp 拷贝

ipfs files mv 移动

ipfs files stat 状态

ipfs files read 读取

React 音乐播放器

音乐列表数据

- 音乐名称/歌手等文本信息
- 专辑图片展示

播放音频数据

- 网易音乐api地址:

<https://github.com/Binaryify/NeteaseCloudMusicApi>

- ipfs-api: <https://github.com/ipfs/js-ipfs-api>
 - 初始化环境端口号开启服务:

```
# 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
```

- cat获取数据

```
ipfs.files.cat("QmY4NqRyr9SebC3P6W3pzg22UK3QsJNDKGzDHqQZsEyPi3", function (err,
file) {
  if (err) {
    throw err
  }
  const json = file.toString('utf8');
  console.log(json)
  that.setState({
    songInfo: JSON.parse(json)
  })
})
```

- add 添加数据

```
const ipfsAPI = require('ipfs-api')
const ipfs = ipfsAPI('localhost', '5001', {protocol: 'http'})
const buffer = Buffer.from('hello ipfs-api!')
ipfs.add(buffer)
  .then( rsp => console.log(rsp[0].hash))
  .catch(e => console.error(e))
```

- 设置cors

```
ipfs config --json API.HTTPHeaders.Access-Control-Allow-Methods '["PUT", "GET",
"POST", "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"]'
```

ipns

- 绑定ipfs节点, 把一个文件/文件夹的hash发布到自己的ID下

```
ipfs name publish QmSx37PT8iV2XxzfHLMRYSxZEt87uE3jdQwCyz7otd5ktP
```

- 查看节点绑定的ipfs路径

```
ipfs name resolve [peerId]
```

- 离线客户端框架: <https://github.com/electron/electron>