

[Subscribe](#)

IPFS for Beginners – Interact With IPFS By Javascript

We have learned how to share your files via IPFS. Also, we have learned how to host a static website on IPFS. However, if you want to make your own application to interact with IPFS it's not enough. In this article, we'll learn how to interact with IPFS by javascript programming language.

Environment Setup

Make sure you have [node.js](#) and [npm](#) on your computer. After you install node.js and npm, you should see your node and npm version by

“\$ node -v && npm -v”

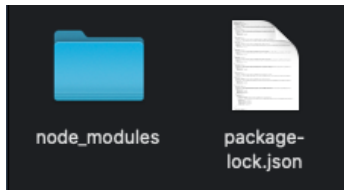
```
(base) MacBook-Air-1470:IPFS Huanry$ node -v
v12.12.0
(base) MacBook-Air-1470:IPFS Huanry$ npm -v
6.11.3
(base) MacBook-Air-1470:IPFS Huanry$
```

Now create your work directory and install “ipfs” node module inside the directory.

“\$npm install ipfs”

It took few minutes, you'll see a node_modules directory after you install the module.

```
((base) MacBook-Air-1470:IPFS Huanry$ npm install ipfs
npm WARN deprecated peer-info@0.17.5: No longer supported for js-libp2p0.28.0 or later
[.....] \ fetchMetadata: sill resolveWithNewModule crypto-browserify@3.12.0 checking installable status
```



Interact with IPFS by javascript

Way to interact with IPFS is by using "require('ipfs').create()"

```
const IPFS = require('ipfs')

async function main () {

  const node = await IPFS.create()

  // Your Codes...

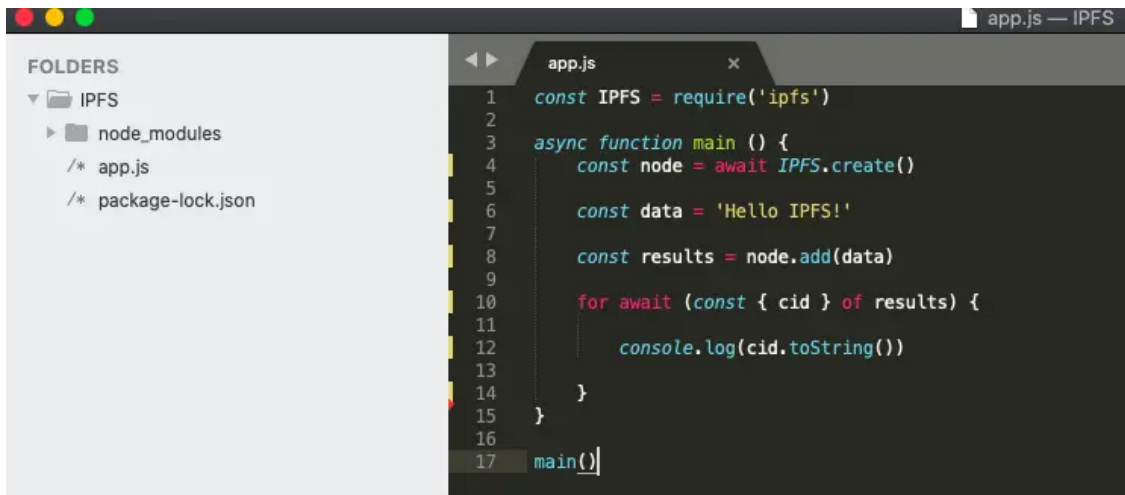
}

main()
```

Interact with IPFS by javascript

Let's make an example to push a "Hello IPFS!" content on IPFS. Create a file named "app.js" in the working directory. You can add content on IPFS through "add".

Subscribe



app.js

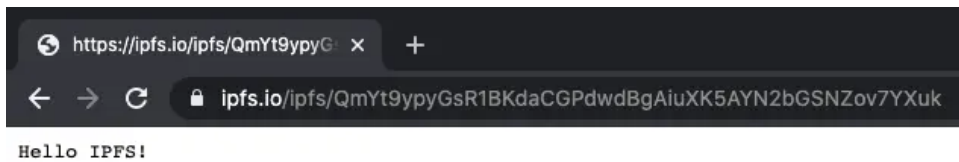
Command

"\$node app"

You can get your CID for the content.

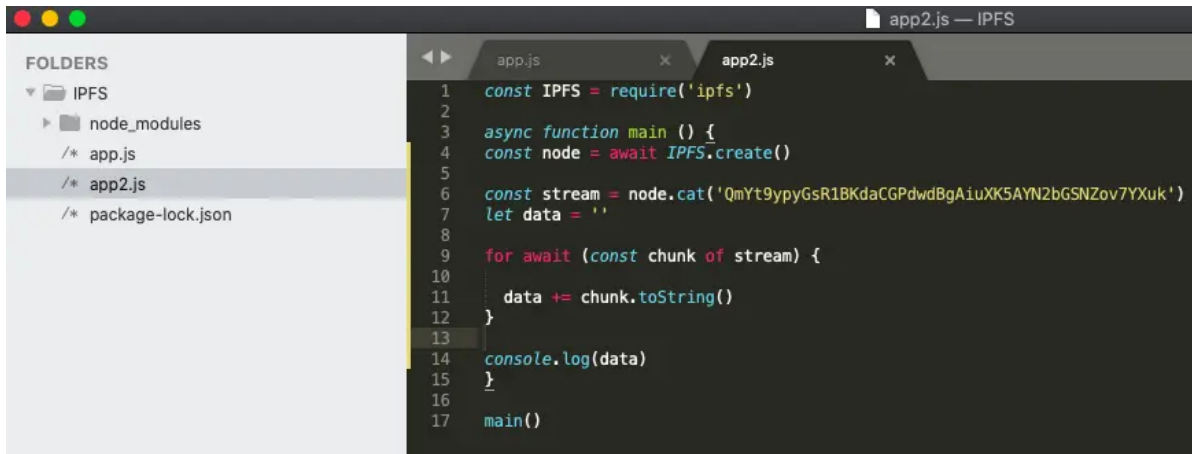
```
((base) MacBook-Air-1470:IPFS Huanry$ node app
Swarm listening on /ip4/127.0.0.1/tcp/4002/p2p/QmYp9qqncGhSjqbf8xsowGwzAWgic7smByr9RiEHNJPT84
Swarm listening on /ip4/192.168.1.106/tcp/4002/p2p/QmYp9qqncGhSjqbf8xsowGwzAWgic7smByr9RiEHNJPT84
Swarm listening on /ip4/127.0.0.1/tcp/4003/ws/p2p/QmYp9qqncGhSjqbf8xsowGwzAWgic7smByr9RiEHNJPT84
QmYt9ypyGsR1BKdaCGPdwdBgAiuXK5AYN2bGSNZov7YXuk
```

Let's check the content through IPFS gateway.

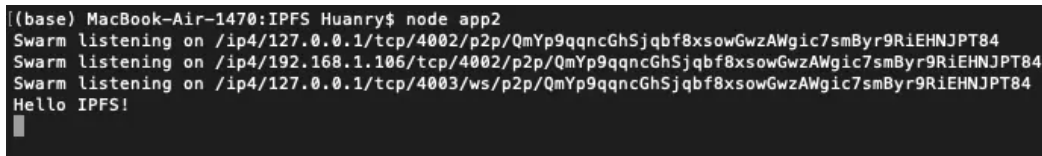


check result through IPFS gateway

Also, we can retrieve data with a cid using "cat". Make another file named "app2.js".



Run "\$node app2" you get the content from the cid now!



Subscribe

Make an application to interact with IPFS

You already know how to interact with IPFS by using javascript above. Let's make an application to interact with IPFS. IPFS have examples on Github to teach you to make your applications.

Copy files from [js-ipfs github](#) into a new working directory. (You can use "\$git clone ..." if you have git on your computer.)

Check the package.json file. In the "devDependencies", it means we'll install these node packages when we run "\$npm install".

And in the "scripts", we see

```

"start": "npm run bundle && npm run serve",

"serve": "http-server public -a 127.0.0.1 -p 8888", and

"bundle": "browserify src/index.js > public/bundle.js".
  
```

It means when we run "\$npm start" we will run "bundle" and "serve".

Running serve, we will make an HTTP server with path "public" on the port "8888". Running bundle, we will convert src/index.js into public/bundle.js for browser to load.

```

1 {
2   "name": "example-browser-browserify",
3   "description": "Bundle js-ipfs with Browserify",
4   "version": "1.1.1",
5   "main": "index.js",
6   "private": true,
7   "scripts": {
8     "bundle": "browserify src/index.js > public/bundle.js",
9     "serve": "http-server public -a 127.0.0.1 -p 8888",
10    "start": "npm run bundle && npm run serve",
11    "clean": "rm -rf public/bundle.js",
12    "test": "test-ipfs-example"
13  },
14  "keywords": [],
15  "license": "MIT",
16  "devDependencies": {
17    "browserify": "^16.2.3",
18    "concat-stream": "^2.0.0",
19    "execa": "^4.0.0",
20    "http-server": "^0.11.1",
21    "ipfs": "^0.47.0",
22    "test-ipfs-example": "^2.0.3"
23  },
24  "browser": {
25    "ipfs": "ipfs/dist/index.min.js"
26  }
27 }
28

```

Let's check file "public/index.html". Check line 6, it will render the bundle.js. (Converted from index.js). Also, we can check line 17/18/19/21/22 with id source/store/cid/content. We'll see what the index.js will do for elements with these ids.

```

1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8"/>
5     <title>js-ipfs example</title>
6     <script src="bundle.js"></script>
7     <style>
8       .content {
9         border: 1px solid black;
10        padding: 10px;
11        margin: 5px 0;
12      }
13    </style>
14  </head>
15  <body>
16    <h1>JS IPFS – Add data to IPFS from the browser</h1>
17    <textarea id="source" placeholder="Enter some text here"></textarea>
18    <button id="store">Add text to ipfs</button>
19    <div id="output" style="display: none">
20      <div>found in ipfs:</div>
21      <div class="content" id="cid">[ipfs cid]</div>
22      <div class="content" id="content">[ipfs content]</div>
23    </div>
24  </body>
25 </html>
26

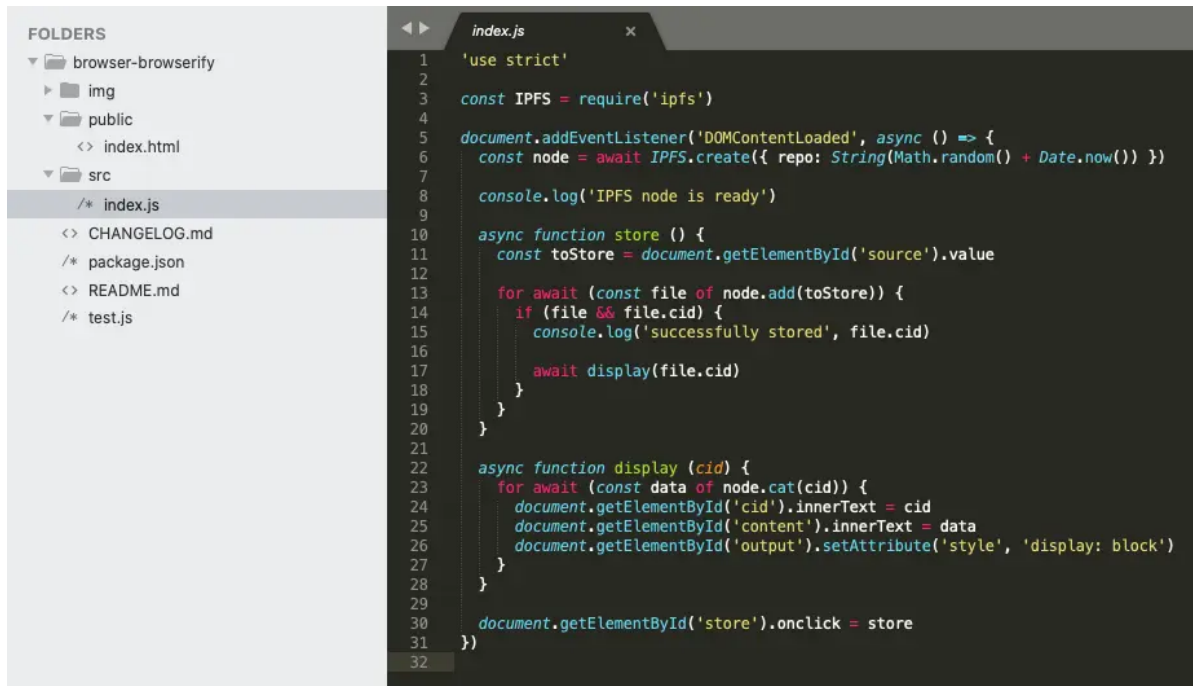
```

Subscribe

Let's take a look at the index.js file. Line 10 is a function, it define "toStore" value from element "id: source"(line 11). Line 12 adds "toStore" to IPFS, and get file value. Line17 will transfer the prop "file.cid" to function "display"

Line 22 define the function display to retrieve IPFS data with cid(line 23), put cid on element "id:cid" (line 24) and data on element "id:content" (line25).

Line make the event when we click the element "id: store" to make store function (line 10) happen.



```

FOLDERS
└─ browser-browserify
  └─ img
    └─ public
      └─ index.html
        └─ src
          └─ index.js
            └─ CHANGELOG.md
              └─ package.json
                └─ README.md
                  └─ test.js

index.js
1  'use strict'
2
3  const IPFS = require('ipfs')
4
5  document.addEventListener('DOMContentLoaded', async () => {
6    const node = await IPFS.create({ repo: String(Math.random() + Date.now()) })
7
8    console.log('IPFS node is ready')
9
10   async function store () {
11     const toStore = document.getElementById('source').value
12
13     for await (const file of node.add(toStore)) {
14       if (file && file.cid) {
15         console.log('successfully stored', file.cid)
16
17         await display(file.cid)
18       }
19     }
20   }
21
22   async function display (cid) {
23     for await (const data of node.cat(cid)) {
24       document.getElementById('cid').innerText = cid
25       document.getElementById('content').innerText = data
26       document.getElementById('output').setAttribute('style', 'display: block')
27     }
28   }
29
30   document.getElementById('store').onclick = store
31 })
32

```

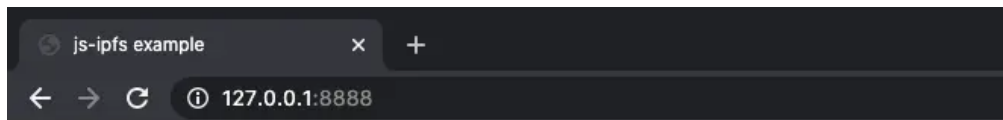
Run the application

Anyway, let's run "\$npm install && npm start", it take few minutes. (We only need to run "\$npm start" for the 2nd time since we already install node packages.) We can see the application is ready on <http://127.0.0.1:8888>, let's check it!

```

Starting up http-server, serving public
Available on:
  http://127.0.0.1:8888
Hit CTRL-C to stop the server

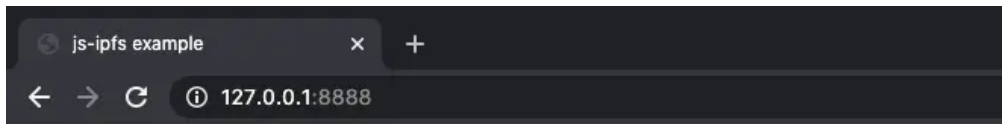
```



JS IPFS - Add data to IPFS from the browser

Let's try to add content "Hello IPFS!" on IPFS.

Subscribe



JS IPFS - Add data to IPFS from the browser

Hello IPFS!

Add text to ipfs

found in ipfs:

QmYt9ypyGsR1BKdaCGPdwdBgAiuXK5AYN2bGSNZov7YXuk

Hello IPFS!

Note that you got the same cid in the session “Interact with IPFS by javascript” since we have the same content.

Now, you have your first application to interact with IPFS.

Read More

[IPFS for Beginners – Storing Data in a Distributed Hash Table.](#)

[IPFS for Beginners – Host Your Static Website.](#)

[IPFS-JS.](#)

[Join @LearnThingsOnline on Telegram](#)

Ads

Subscribe

Leave a Reply

Your email address will not be published. Required fields are marked *

COMMENT

NAME *

EMAIL *

☐ Save my name, email, and website in this browser for the next time I comment.Post Comment

LEARNTHINGS.ONLINE TELEGRAM GROUP

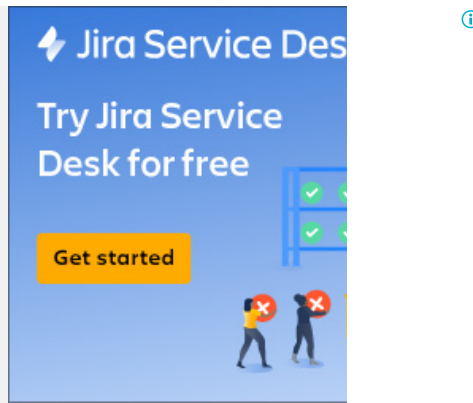
[Don't have Telegram yet? Try it now!](#)

Learn Things Online

65 members, 5 online

This group build to share some materials to learn blockchain online & news. Check LearnThings.Online

[View in Telegram](#)If you have Telegram, you can view and join
Learn Things Online right away.[Subscribe](#)

**RSS**

IPFS for Beginners – Interact With IPFS By Javascript

In this article, we'll learn how to interact with IPFS by javascript programming language. It's one way to make your own application to interact with IPFS. The post IPFS for Beginners – Interact With IPFS By Javascript appeared first on LearnThings.Online.



Download CCleaner For Free

CCleaner.com

Get the #1 tool for cleaning
your PC. CCleaner
downloaded over 2.5
times.

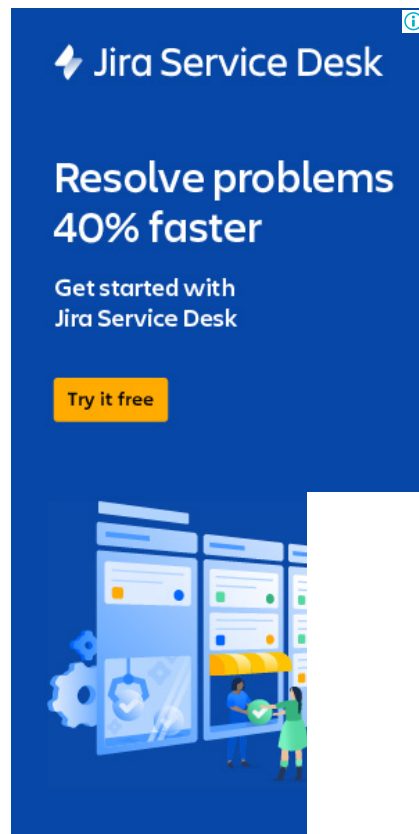
OPEN

Subscribe

Facebook Rename Its Libra Wallet Project Calibra to Novi

2020 May 26, Facebook rename its Libra wallet project Calibra to Novi. It makes its name more separate from Libra. Novi plans to launch its App in 2020. The post Facebook Rename Its Libra Wallet Project Calibra to Novi appeared first on LearnThings.Online.

Libra Appoints It's General Counsel, a Former HSBC, and Goldman Sachs



On May 19th, 2020, the Libra association appoint Robert Werner, an Ex-HSBC & Ex-Goldman Sachs the founder and CEO of GRH Consulting, as its general counsel. The post Libra Appoints It's General Counsel, a Former HSBC, and Goldman Sachs appeared first on LearnThings.Online.