


 web3-storage / w3name




<> Code Issues 10 Pull requests 1 Actions Projects Security I



IPNS client and service for generating, updating, and tracking immutable names with signed updates

 View license



☆ 39 stars  10 forks  4 watching  Activity

 Public repository

 main ▾

...

 Branches  Tags

 dchoi27 ... on Jun 7 

[View code](#)

 README.md

w3name

Content addressing for a dynamic web.

About

w3name is a service and client library that implements [IPNS](#), which is a protocol that uses public key cryptography to allow for updatable naming in an atomically verifiable way.

Naming is famously one of the few hard problems in computer science, the exact number of which vary depending on the computer scientist you're speaking with and their predisposition to off-by-one errors.

Naming can be especially difficult in distributed and decentralized systems, where you can't assume that everyone has the same view of the network, and there may not be any centralized "naming authorities" to resolve conflicts.

IPNS works by using public key cryptography to allow "self-issued" names which don't require any coordination or central authorities. The caveat is that the definition of "name" is somewhat constrained compared to general-purpose key/value storage systems.

A "name" in the IPNS context is an identifier for a public key, or in some cases, an encoding of the public key itself.

To associate a value with the name, the holder of the corresponding private key creates a record containing the value and signs it with their private key. They then publish that record to an IPNS name service, such as the IPFS DHT or the w3name service.

Anyone can query the service using the name and retrieve the latest value, including everything needed to verify that the value was signed by the correct key and has not been altered since publication.

JS Client

The `w3name` JavaScript client library provides a simple interface to the service API, as well as everything you need to create signing keys and records.

The snippet below shows how to create a new w3name key with `Name.create()`, add a revision with `Name.v0({name}, {CID})` and publish the revision with `Name.publish({revision}, {key})`

To use the library in your project, use npm or yarn to install the `w3name` module.

node.js

```
import * as Name from 'w3name'

const name = await Name.create()

const value = '/ipfs/bafkreiem4twkqzsq2aj4shbycd4yvoj2cx72vezicletlhi7dijjciqpui'
const revision = await Name.v0(name, value)

await Name.publish(revision, name.key)
```



See <https://github.com/web3-storage/w3name/blob/main/packages/client/README.md> for a guide to using the js client for the first time.

Full API reference doc for the JS client are available at <https://web3-storage.github.io/w3name>

You can also find a [how-to guide for working with the JS client](#) included in the [Web3.Storage](#) documentation.

There is a rate limit of 30 requests per 10 seconds per IP.

cURL

You can easily get the current value for any name by sending an HTTP request to `https://name.web3.storage/name/:key`, where `:key` is the name string. This will return a JSON object with the following shape:

```
{
  "value": "the current value, as a string",
  "record": "the full IPNS record for the current value, encoded to a binary form and"
}
```



Try pasting this into a terminal with `curl` installed:

```
curl https://name.web3.storage/name/k51qzi5uqu5d1cuzv5xhg1zqn48gobcvn2mx13uoig7zfr58rz
```





You should see output similar to this:

```
{
  "value": "/ipfs/bafkreigbpn5osrexvl56ogyp5kivof2tmknkssk5odebqqeu23a22bcntu",
  "record": "long base64 string, omitted for brevity..."
}
```



There is a rate limit of 30 requests per 10 seconds per IP.

See <https://github.com/web3-storage/w3name/tree/main/packages/api> for documentation on other available API endpoints  

Building w3name

Want to help us improve w3name? Great! This project uses node v16 and npm v7. It's a monorepo that use [npm workspaces](#) to handle resolving dependencies between the local `packages/*` folders.

Install the deps with `npm`

```
# install deps
npm install
```



Run all the things with `npm start`.

```
# start the api
```



`npm start`

Monorepo

This project is a monorepo that uses [npm workspaces](#).

All `npm` commands should be run from the root of the repo. To run a command for a specific package add the `--workspace` or `-w` flag

```
# Start just the api
npm run dev -w packages/api
```



To add a new workspace (aka package) to the monorepo

```
# adds the path to the `website` package to the `workspaces` property in package.json
npm init -w packages/website
```



To run an npm script in one or more workspaces

```
# run test command in package `a` and `b`
npm run test --workspace=packages/a --workspace=packages/b
```



Testing


Each workspace has its own suite of testing tools, which you can learn more about in the relevant `packages/*` directory.

Learn more

To learn more about the w3name service, or find detailed documentation for the JS client library and API, please head over to [Web3.Storage documentation](#).

**

Releases 11

 **w3name: v1.0.8** Latest
on Feb 6

[+ 10 releases](#)

Packages

No packages published

Contributors 9



Deployments 5

✔ github-pages last year

+ 4 deployments

Languages

● TypeScript 88.7% ● JavaScript 9.3% ● Shell 2.0%