


How to build a website for the IndieWeb



[jackdbd/sfscon-2024-indieweb](https://github.com/jackdbd/sfscon-2024-indieweb) 

Giacomo Debidda

Freelance full stack developer / web performance consultant

I write TypeScript / Clojure / Zig

I like 🛹 and 🚲

BTW I use ⚙️

🐙 jackdbd

👤 giacomodebidda.com






The Golden Age of the Internet

The web today is much more **centralized** than it was in the past.

I miss when the internet felt like an **endless ocean** full of passion and creativity. It's now full of corporations, tribalism and people who seek to earn money/push their political views.

—wallakfir90

Source: I Miss Forums (And the Decentralized Internet) 

The web today is much more **homogenous** than it was in the past.

Even in the days of MySpace, users enjoyed a level of **control** over their personal pages not seen today, with the ability to change the CSS and background image to their liking. Unfortunately, those times are largely behind us now —lost somewhere in the mid-2000s.

In the Facebook era, everybody shares the same sterile profile, with the only avenues for **self-expression** being the photos and comments that they post.

—Dave Heinemann

Source: This is not the Web I've Known 





Silos

A silo (AKA walled garden) is a **centralized** web site (like most social media) that stakes some **claim to content** contributed to it and **restricts access** in some way (has walls).

”

Source: [silo](#), on [indieweb.org](#) 

- require you to create an account specific to that site to use it (silo identity)
- allow you to interact only with accounts on that site (silo contacts)
- allow you to post only specific forms of content (e.g. text of 140 characters)
- claim some ownership or license to any content you create within the silo
- restrict your ability to import/export your content (e.g. posts, comments, tags)
- have an access wall that prevents indexing of (at least some of the) content you contribute
- have restrictive terms of service (TOS)
- can be shut down ([acqui-hire](#) , [site-death](#) ) , taking content and permalinks with them

Perhaps you're done with
others owning your **content**,
your **identity**, and your **self**

Reclaim your
content and
self host it




Own your domain

A personal domain name is an inexpensive, internationally **universal identifier** which gives you more control over your space than other IDs (e.g. email address or phone number.) ”

Source: IndieWebify.Me @

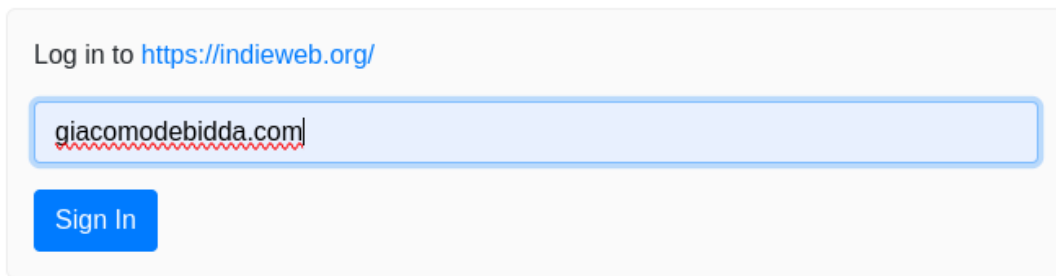
Why not just a **subdomain**?

A subdomain (like `example.wordpress.com` or `example.github.io`) **is not something you own**, rather you are at the mercy of the corporation that owns the actual domain name, who thus has ownership and control of all subdomains as well. ”

Source: personal-domain, on indieweb.org 

Web Sign-in

Web sign-in  is a **mechanism** for signing in to websites using your domain.



A mockup of a web sign-in form. It features a light gray background with rounded corners. At the top, it says "Log in to" followed by the URL "https://indieweb.org/" in blue. Below this is a light blue input field with a rounded rectangle border, containing the text "giacomodebidda.com" with a red wavy underline. At the bottom of the form is a blue button with the text "Sign In" in white.

1. A website presents a login form to the user.
2. The user enters their personal domain name into the login form.
3. The website verifies that the user has control of that domain using an **authentication protocol** like RelMeAuth or IndieAuth.

RelMeAuth

RelMeAuth uses `rel="me"` link(s) to match your domain name with existing identity(ies).

In RelMeAuth an identity can be:

- The account you have with an OAuth 2.0 provider (e.g. your GitHub account)
- Your email
- Your OpenPGP key

website → provider

Option A: `` in the `<body>`

```
<body>
  <a href="https://github.com/jackdbd"
    rel="me">@jackdbd on Github</a>

  <a href="https://fosstodon.org/jackdbd"
    rel="me">@jackdbd on Mastodon</a>

  <a href="mailto:me@example.com"
    rel="me">me@example.com</a>

  <a href="/assets/openpgp-pubkey.txt"
    rel="pgpkey">My OpenPGP public key</a>
</body>
```

Option B: `<link rel="me">` in the `<head>`

```
<head>
  <link rel="me" href="https://github.com/jackdbd" />
  <link rel="me" href="https://fosstodon.org/@jackdbd" />
  <link rel="me" href="mailto:me@example.com" />
```

provider → website

- Add your website in your Github profile
- Verify your identity on Mastodon
- Receive a verification code via email
- Sign a challenge with your OpenPGP private key
(e.g. `echo <challenge> | gpg --clearsign -armor`)

Use indiewebify.me/validate-rel-me to validate that your domain name and profiles are linked together.

2. Set up Web Sign In

In order to be able to sign in using your domain name, connect it to your existing identities.

You probably already have many disconnected profiles on the web. Linking between them and your domain name with the `rel="me"` microformat ensures that it's easy to see that you on Google/Twitter/Github/Flickr/Facebook/email are all the same person as your domain name.

On the wiki: [How to set up Web Sign In](#).

We found the following `rel="me"` URLs on your site:

- <https://github.com/jackdbd> works perfectly
- <mailto:giacomo@giacomobidda.com> only http and https links are validated
- <https://fosstodon.org/@jackdbd> works perfectly

Example: IndieWeb.org

On my site:

```
<link rel="authorization_endpoint" href="https://indielogin.com/auth">
```

On IndieWeb.org:

Log in to <https://indieweb.org/>

giacomodebidda.com

Sign In

Sign in to indielogin.com/id

Authenticate using one of the methods below to sign in as
<https://giacomodebidda.com/>

3 supported and verified providers were found!

8 hours ago [Refresh](#)



giacomo@giacomodebidda.com

Receive a verification code via email



GPG (giacomodebidda.com/assets/pgp-key.txt)

Click to sign a challenge with your GPG key.



github.com/jackdbd

Ok! Click to authenticate using this provider.

You will be redirected to <https://indielogin.com/redirect/indieauth> after authorizing this application.

User profile

Appearance

Editing

Recent changes

Watchlist

Search

Basic information

Username: Giacomodebidda.com

Member of groups: [Autoconfirmed users](#), [Users](#)

Number of edits: 0

Registration time: 18:15, 30 October 2024

Real name:

Real name is optional. If provided, it may be used to give you attribution for your work.

Access account data: [My account data from this project](#)

Reset settings: [Restore all default preferences \(in all sections\)](#)

Internationalisation

Language:

en - English

▼

How do you prefer to be described?

- ☒ They edit wiki pages
(When mentioning you, the software will use gender neutral words whenever possible)
- ☐ She edits wiki pages
- ☐ He edits wiki pages

Setting this preference is optional. The software uses its value to address you and to mention you to others using the appropriate grammatical gender. This information will be public.

Save



Own your identity

Instead of logging in to websites as "you on Twitter" or "you on Facebook", you should be able”
to log in as just "you". We should **not be relying on Twitter or Facebook to provide our
authenticated identities**, we should be able to use our own domain names to log in to sites
everywhere.


Source: IndieAuth.com: Sign in with your domain name @

IndieAuth

Every service that spins up an OAuth-enabled API ends up being its own **isolated system**. ”

For example, if I want to build an app that can read someone's step count from FitBit, I have to first go register as a developer on FitBit's website in order to get API keys to use with their OAuth API.


—Aaron Parecki

Source: OAuth for the Open Web 

IndieAuth is:

- an OAuth 2.0 extension that defines an identity layer
- an authentication protocol alternative to OpenID Connect
- typically used to obtain an OAuth 2.0 Bearer tokens for Micropub clients
- used on sites like indiebookclub.biz

IndieAuth

- **Unique user identifier**: a URL
- Identity is **portable**
- Identity is tied to DNS
- OAuth 2.0 clients (aka OAuth apps) require no registration, since client IDs are resolvable URLs
- Defines 2 scopes  for the access token
- Your app can define additional scopes/claims
- access token, no ID token
- User's info available at the `/userinfo` endpoint
- Does not deal with session management, focuses on just returning the user's identifier

OpenID Connect

- **No** consistent **unique user identifier** across providers
- Identity is **not portable** → many identities
- Each identity is tied to an OpenID Connect provider
- OAuth 2.0 clients require registration (e.g. you need to create a GitHub OAuth App, a LinkedIn app, etc.)
- Defines 20 standard claims
- OIDC provider can define additional scopes/claims
- Your app can define additional scopes/claims
- access token and ID token
- Claims returned in an ID Token and are also available through the `/userinfo` endpoint
- Deals with session management, ID tokens can be used as a session

No OAuth 2.0 client registration

In IndieAuth the client ID is a resolvable URL

```
GET https://indiebookclub.biz/id
```

```
{  
  "client_id": "https://indiebookclub.biz/id",  
  "client_name": "indiebookclub",  
  "client_uri": "https://indiebookclub.biz/",  
  "logo_uri": "https://indiebookclub.biz/images/book.svg",  
  "redirect_uris": ["https://indiebookclub.biz/auth/callback"]  
}
```

Allow indiebookclub.biz/id access to giacomodebidda.com?

The app <https://indiebookclub.biz/id> would like to access your site, <https://giacomodebidda.com/>

The app is requesting the following [scopes](#):

- create
- draft
- profile

Authenticate using one of the methods below to approve this request.

4 supported and verified providers were found!

1 hour ago [Refresh](#)



giacomo@giacomodebidda.com

Receive a verification code via email



[IndieAuth](#)

Ok! Click to authenticate using your IndieAuth server.



[GPG \(giacomodebidda.com/assets/pgp-key.txt\)](https://giacomodebidda.com/assets/pgp-key.txt)

Click to sign a challenge with your GPG key.




github.com/jackdbd

Ok! Click to authenticate using this provider.

You will be redirected to <https://indiebookclub.biz/auth/callback> after authorizing this application.

How does IndieAuth work?

An **IndieAuth server** is a set of endpoints: authorization endpoint, token endpoint, revocation endpoint, introspection endpoint, userinfo endpoint.

A **website/app** that implements IndieAuth advertises these endpoints at the URL `rel=indieauth-metadata`, using IndieAuth Server Metadata  (an extension of OAuth 2.0 Authorization Server Metadata).

```
Link: <https://giacomodebidda.com/.well-known/oauth-authorization-server>; rel="indieauth-metadata"
```

```
<link rel="indieauth-metadata" href="https://giacomodebidda.com/.well-known/oauth-authorization-server">
```

IndieAuth clients fetch the URL at `rel=indieauth-metadata` and discover these endpoints.

```
{
  "authorization_endpoint": "https://indieauth.com/auth",
  "introspection_endpoint": "https://micropub.fly.dev/introspect",
  "issuer": "https://giacomodebidda.com/",
  "revocation_endpoint": "https://micropub.fly.dev/revocation",
  "scopes_supported": ["email", "profile", "create", "draft", "update", "delete", "media"],
  "token_endpoint": "https://micropub.fly.dev/token",
```

```
"userinfo_endpoint": "https://micropub.fly.dev/userinfo"
```

```
}
```



Own your content

Publish On your own Site

You own the canonical URL \mathbb{W} of your content...

```
<head>
  <link rel="canonical"
        href="https://www.giacomodebidda.com/articles/performance-audit-italian-news-website/">
</head>
```

Syndicate Elsewhere

...and publish a copy of your content on other platforms (e.g. social media).

Microformats2

<https://microformats.org/wiki/microformats2>

Content hosted under your domain should be presented as semantically marked up HTML+microformats.

https://indieweb.org/syndication_formats

Once you have well-marked-up data under your domain, you should make it available as a feed, and you'll probably want to syndicate it out to 3rd party services.

TODO: show snippets from your blog

Add microformats to your content

 Add the h-entry microformat markup to your posts. Many CMSs and themes already support them!

You can use the h-entry validator to verify that your recent post has validate your h-entry.

Why? This will allow other people's software to easily read and understand your content. This is useful for a variety of things like recognizing comments, likes, reposts, and displaying reply-contexts for your posts.

https://indieweb.org/Getting_Started#IndieWeb_self-starters

h-card

Update your index.html home page to include your basic information in an h-card. This h-card can be as simple as your name.

Why? When you publish content, you can link back to your home page using rel-author and your authorship information can be retrieved from the h-card.

Advantages: While you are not yet publishing content on your own site, at this point you have:

Staked your claim on the indieweb Set up an identity that you own and control Why add my basic contact information in an h-card?

This helps with providing authorship information for anything you publish and link back to your home page with rel-author.

Validate your `h-card` here: <https://indiewebify.me/validate-h-card/>



Micropub

A protocol 🚩🔴🟡 for creating, editing, deleting, undeleting a post 🚩🔴🟡 on a website hosted on your own domain using third-party clients. And for syndicating that post elsewhere.

A **website/app** advertises its micropub endpoint at the URL `rel=micropub`.

A **Micropub client**:

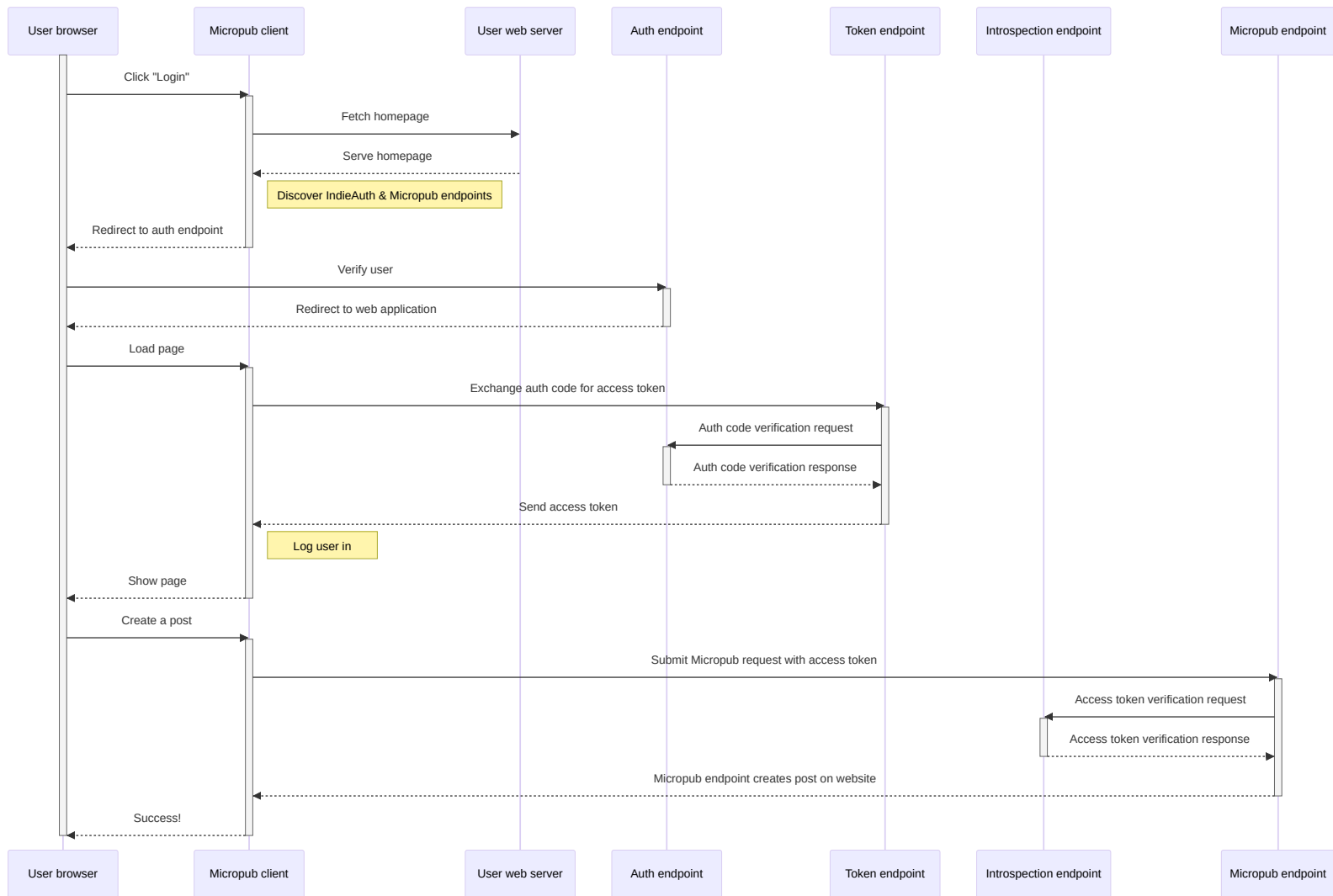
- Discovers your IndieAuth endpoints and your Micropub endpoints.
- Authenticates you (e.g. using RelMeAuth or IndieAuth).
- Submits your posts to the appropriate Micropub endpoint.

```
Link: <https://micropub.fly.dev/micropub>; rel="micropub"
```

```
<link rel="micropub" href="https://micropub.fly.dev/micropub">
```

A **Micropub server** is a set of endpoints:

- A `/micropub` endpoint for create / update / delete / undelete **actions** on a post (e.g. a note).
- A `/media` endpoint for actions on some media (e.g. an image, some audio).





Endpoints discovery

The **micropub endpoint** is discovered using either a HTTP Link header...

```
Link: <https://micropub.fly.dev/micropub>; rel="micropub"
```

...or a `<link rel="me">` tag in the `<head>`.

```
<link rel="micropub" href="https://micropub.fly.dev/micropub">
```

The **media endpoint** is discovered with a `GET` to the micropub endpoint.

```
GET /micropub?q=config
Authorization: Bearer xxxxxxxxx

{
  "media-endpoint": "https://media.example.com/micropub"
}
```



Sign In

indiebookclub is able to post to your website!

Click the button below to sign in and allow this app to post to your site.

[Sign In](#)

▼ Debugging Information:

indiebookclub found the following endpoints on your site:

- IndieAuth metadata endpoint: `https://giacomodebidda.com/.well-known/oauth-authorization-server`
- Authorization endpoint: `https://indieauth.com/auth`
- Token endpoint: `https://micropub.fly.dev/token`
- Micropub endpoint: `https://micropub.fly.dev/micropub`
- Revocation endpoint: `https://micropub.fly.dev/revocation`




Syndication (cross-posting)


1. Publish (on your) Own Site → Your site has the original post (i.e. the canonical URL)
2. Syndicate Elsewhere → syndication targets (e.g. social networks) get a copy of your post

By posting first on your own site, you create a direct ownership chain that can be traced back to you without any intervening 3rd party services (silos) TOS's getting in the way.



Source: POSSE, on indieweb.org 

- Your **micropub endpoint** should list all the syndication targets you want to send your post to.
- Your **micropub client** should have a way to publish to the syndication targets. For example, for posting on LinkedIn you will need to create a LinkedIn OAuth app.



As you can read in Rethinking syndication #581 , implementing syndication is not trivial.



Backfeed (reverse syndication)

A POSSE copy of your original content is published on a silo (e.g. a social network). People interact with that content. You want to capture these **interactions** so you can have **cross-site conversations**.

How can you do it?

1. Poll the silo API (or scrape the silo) to retrieve interactions like replies, likes, reposts, bookmarks
2. Link each interaction to the original content using original post discovery 
3. Convert interactions into webmentions 
4. Submit all webmentions to your webmention endpoint
5. Fetch your webmention endpoint and display the webmentions on your site

This process is called backfeed.

You can implement it yourself or use a backfeed proxy like Bridgy 

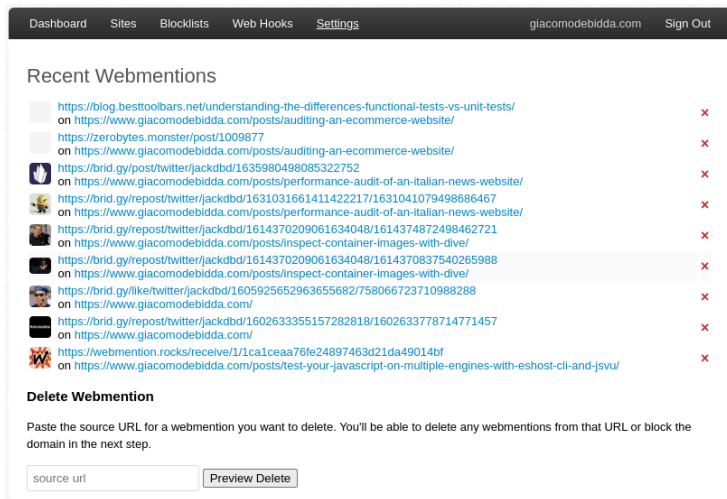
Receiving webmentions

You advertise your webmention endpoint in the `<head>` of your HTML pages.

```
<link rel="webmention" href="https://webmention.io/giacomodebidda.com/webmention">
<link rel="pingback" href="https://webmention.io/giacomodebidda.com/xmlrpc">
```

Bridgy sends webmentions to your webmention endpoint.

Webmentions appear on your webmention.io dashboard.



The screenshot shows the webmention.io dashboard for the user 'giacomodebidda.com'. The top navigation bar includes links for Dashboard, Sites, Blocklists, Web Hooks, and Settings. The main section is titled 'Recent Webmentions' and displays a list of incoming mentions. Each entry shows the source URL, the target URL (https://www.giacomodebidda.com), and a red 'x' icon indicating a failed mention. Below the list is a 'Delete Webmention' section with instructions on how to delete a mention by pasting the source URL. A 'Preview Delete' button is visible at the bottom of this section.

Source URL	Target URL	Status
https://blog.besttoolbars.net/understanding-the-differences-functional-tests-vs-unit-tests/	on https://www.giacomodebidda.com/posts/auditing-an-e-commerce-website/	×
https://zerobytes.monster/post/1009877	on https://www.giacomodebidda.com/posts/auditing-an-e-commerce-website/	×
https://bridgy/post/twitter/jackdbd/1635980498085322752	on https://www.giacomodebidda.com/posts/performance-audit-of-an-italian-news-website/	×
https://bridgy/repost/twitter/jackdbd/1631031661411422217/1631041079498686467	on https://www.giacomodebidda.com/posts/performance-audit-of-an-italian-news-website/	×
https://bridgy/repost/twitter/jackdbd/1614370209061634048/1614374872498462721	on https://www.giacomodebidda.com/posts/inspect-container-images-with-dive/	×
https://bridgy/repost/twitter/jackdbd/1614370209061634048/1614370837540265988	on https://www.giacomodebidda.com/posts/inspect-container-images-with-dive/	×
https://bridgy/like/twitter/jackdbd/1605925652963655682/758066723710988288	on https://www.giacomodebidda.com/	×
https://bridgy/repost/twitter/jackdbd/1602633355157282818/1602633778714771457	on https://www.giacomodebidda.com/	×
https://webmention.rocks/receive/1/1ca1ceaa76fe24897463d21da49014bf	on https://www.giacomodebidda.com/posts/test-your-javascript-on-multiple-engines-with-eshost-cil-and-jsvu/	×

Delete Webmention

Paste the source URL for a webmention you want to delete. You'll be able to delete any webmentions from that URL or block the domain in the next step.

Fetching webmentions

You can fetch webmentions at runtime, with client-side JS, or at build time.

I fetch webmentions whenever I build my Eleventy site.

```
import EleventyFetch from '@11ty/eleventy-fetch'

const format = 'jf2' // https://www.w3.org/wiki/JF2
const endpoint = `https://webmention.io/api/mentions.${format}`

const response = await EleventyFetch(
  `${endpoint}?page=1&per-page=25&token=${process.env.WEBMENTION_IO_TOKEN}`,
  {
    directory: '.cache-webmentions',
    duration: '30d',
    type: 'json',
    verbose: true
  }
)

const { likes, replies, reposts } = responseToWebmentions(response)
```

Don't forget to sanitize the webmentions before displaying them on your site.

Sending webmentions

You want to send me a webmention? You can cite my article on your site and either:

- Go to webmention.io/giacomodebidda.com/webmention and fill in the form.
- Fill in the form I include on my site.
- Let a service like webmention.app send your outgoing webmentions automatically.

Webmentions










*Did you mention this blog post on your website? Let me know the URL of **your article** using the form below.*

Send Webmention

Upon form submission, your [webmention](#) will be sent to [Webmention.io](https://webmention.io).

Webmentions collected by [Bridgy](#).

Recap

-  Read Getting Started
-  Buy a domain
-  Build a website and host it on your domain
-  Write your content with microformats2
-  Publish your content on your domain
-  Syndicate content to other platforms
-  Backfeed (reverse syndicate) content from other platforms to your site
-  Use IndieAuth to have full control of your identity
-  Use IndieWebify.me and IndieMark as checklists to verify your progress

Thanks

[jackdbd/sfscon-2024-indieweb](#) 