

Simple Analytics

This pdf is generated on April 27, 2024. Go to [the documentation page](#) for the most up-to-date version.

Proxy all requests

Concerned about your website visitors' privacy when using Simple Analytics? We understand. Let's be clear: we never collect your visitors' IP addresses. And with our proxy setup option using Caddy, NGINX, or Netlify, you have the power to ensure those IP addresses never reach us. This setup acts as a privacy shield, offering you and your visitors added peace of mind.

If Caddy is your tool of choice, you can quickly set it up as a proxy with just a few lines in your Caddyfile. If you're using NGINX, you can do something similar by adding some configuration to your server directive. And if you're on Netlify, you can set up a proxy by adding redirect rules to your site's configuration. This lets analytics traffic go through your site before reaching us.

Setting up a proxy means no visitor IPs get to our servers. It's an easy and effective step for protecting privacy. We highly recommend it if you can do it.

Server proxies

Set up proxy in NGINX

Trailing slashes are very important here. Keep them as they are in this example.

```
location /simple/ { proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Proto-Version $http2; proxy_set_header Host
$http_host; proxy_set_header X-NginX-Proxy true; proxy_set_header Connection
""; proxy_pass_request_headers on; proxy_pass
https://queue.simpleanalyticscdn.com/; } location = /proxy.js { expires 7d;
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains"
always; proxy_pass https://simpleanalyticsexternal.com/proxy.js?
hostname=example.com&path=/simple; }
```

Change `example.com` to the domain you run the proxy on. This can be the same domain as your own website is hosted on.

You can change the path that you proxy from (`/simple/`) to something else. Make sure to also update that in the second `proxy_pass` (`&path=/simple`).

Reload your NGINX config with `sudo nginx -t && sudo nginx -s reload`.

Set up proxy in Caddy

```
example.com { handle /simple/* { uri strip_prefix /simple reverse_proxy
https://queue.simpleanalyticscdn.com { header_up X-Caddy-Proxy "true" header_up
-X-Forwarded-For } } handle /proxy.js { rewrite * /proxy.js?
{query}&hostname=example.com&path=/simple reverse_proxy
https://simpleanalyticsexternal.com { header_up -X-Forwarded-For } } }
```

By default, Caddy adds the X-Forwarded-For header. We stop this behaviour by adding `header_up -X-Forwarded-For`.

Change `example.com` to the domain you run the proxy on. This can be the same domain as your own website is hosted on.

Thanks to captcha.eu to provide us the Caddy configuration.

Set up proxy in Netlify

Add this to your `_redirects` config file:

```
/proxy.js https://simpleanalyticsexternal.com/proxy.js?
hostname=example.com&path=/simple 200 /simple/*
https://queue.simpleanalyticscdn.com/:splat 200
```

Change `example.com` to the domain you run the proxy on. This can be the same domain as your own website is hosted on.

Feel free to change `/simple` to something else.

Thanks to Brian LaLonde servantsystems.com to provide us the Netlify configuration.

You can adapt this setup for other server configurations too. If you're using a server setup we haven't mentioned, let us know so we can add it to this documentation page.

Update embed script

The embed script should be changed into this:

```
<script async defer src="https://example.com/proxy.js"></script>
```

If you really need to noscript-tag (we recommend against it)

We recommend against using the `<noscript>` -tag, because it's adding more bot traffic.

But if you want, you can include it like this:

```
<script async defer src="https://example.com/proxy.js"></script> <noscript >
</noscript>
```

Note the `/simple` prefix in the `noscript.gif` image and *not* in `proxy.js` .

This will send all traffic via your proxy on the endpoint `/simple/...` .

Validate your config

After these changes you should be able to visit the script at `https://example.com/proxy.js` .

It should return our embed script, starting with:

```
/* Simple Analytics - Privacy friendly analytics ... */
```

The `simple.gif` request (in your developer tools) should come back as a GIF with HTTP status code 202. If that's the case, it works.

[Let us know](#) if we can help you with anything!

This pdf is generated on April 27, 2024. Go to [the documentation page](#) for the most up-to-date version.