Custom Domain is a fully-managed service that enables you to configure a first-party subdomain over HTTPS to track event requests through your domain instead of tracking events through Segment's default domain. For more information, see the Custom Domain documentation.

Segment's domain delegation solutions

Segment offers two domain delegation solutions: Custom Proxy and Custom Domain. If you use Custom Domain, you can choose to use either DNS delegation or a Cannonical Name (CNAME). Segment recommends using Custom Domain with DNS delegation, which leads to easy setup, maintenance, and monitoring.

SERVICE	HOW IT WORKS	INFRASTRUCTURE MANAGEMENT	AVAILABILITY
Custom Domain with DNS Delegation	A Segment service that allows your website to use your own subdomain to load Analytics.js securely over HTTPS and send event data. It is not limited to Analytics.js and is also compatible with server libraries. It uses a DNS subdomain that you delegate to Segment.*	**Segment manages all related infrastructure**, including applying security updates, managing the SSL certificate lifecycle, and monitoring.	Business Tier Recommended for reliable data collection.

SERVICE	HOW IT WORKS	INFRASTRUCTURE MANAGEMENT	AVAILABILITY
Custom Domain with CNAME	This approach uses a Canonical Name (CNAME) to map an alias name on your domain name to Analytics.js. It is not limited to Analytics.js and is also compatible with server libraries.	Customers are responsible for maintaining CNAME.	Business Tier Not recommended due to evolving and persistent browser privacy measures.
Custom Proxy	This approach uses a proxy or wrapper where all data is first collected by a proxy on your domain and then forwarded to Segment.	Customers are responsible for maintaining their own proxy infrastructure.	Available to all Segment users. Not recommended because it adds a point of failure, but remains an option if Custom Domain with sub-domain delegation is unavailable to you.

[*] If it's not possible for you to delegate subdomains to Segment, you can use a CNAME instead. Segment encourages users to delegate a DNS subdomain rather than use use CNAME aliasing due to the evolving privacy standards in browsers, but CNAME aliasing remains an option for users not interested in using nameservers.

Custom Proxy prerequisites

To set up a custom proxy, you need:

- Access to your site DNS settings
- A CDN you can serve assets from
- Access to the CDN settings
- A security certificate for the proxy domain

ℹ **Custom Proxy Troubleshooting**

If you experience issues configuring a custom proxy, contact your organization's IT department for help. Segment does not have access to the resources you need to configure a custom proxy.

This guide explains how to set up a custom proxy in CloudFront. You can apply these principles to almost any modern CDN that supports proxies.

You need to set up two important parts, regardless of the CDN provider you use:

- Proxy to Segment CDN (`cdn.segment.com`)
- Proxy to Segment tracking API (`api.segment.io`)

⚠ If you are using a Regional Workspace, please note that instead of using `api.segment.io` to proxy the Tracking API, you'll be using `events.eu1.segmentapis.com`

ℹ Segment only has the ability to enable the proxy setting for the Web (Analytics.js) source. Details for mobile source proxies are in the Analytics-iOS and Analytics-Android documentation. It is not currently possible to set up a proxy for server sources using the Segment UI.

ℹ **Segment loads most integrations through the proxy, except for third-party SDKs**

Third-party SDKs are loaded by a partner's CDN, even with a Segment proxy configured. For example, if you have a Segment custom proxy enabled and send data to a FullStory destination, FullStory's CDN would load the FullStory SDK.

Custom Proxy setup

There are two options you can choose from when you set up your custom domain proxy.

1. CloudFront
2. Custom CDN or API proxy

Follow the directions listed for CloudFront or use your own CDN setup. Once you complete those steps and verify that your proxy works for both `cdn.segment.com` and `api.segment.io`, contact Segment Product Support with the following template email:

```
Hi,

This is {person} from {company}. I would like to configure a proxy for the following source(s):

**Source URL**: link to the source in your Segment workspace (for example: https://app.segment.com/<your_slug>/source
s/<source>/overview)
**Source ID**: navigate to **API Keys** on the left-hand side of the source **Settings** and provide the Source ID
```

Double-check the Source URL and the Source ID.

A Segment Customer Success team member will respond that they have enabled this option for your account. When you receive this confirmation, open the source in your workspace, and navigate to Settings > Analytics.js. Update the **Host Address** setting from `api.segment.io/v1` to `[your proxy host]/v1`.

> The **Host Address** field does not appear in source settings until it's enabled by Segment Customer Success.

There should be no downtime once the setup is complete, as the default Segment domains continue to work alongside the customer's domains.

Custom CDN / API Proxy

Follow these instructions after setting up a proxy such as CloudFront. Choose between the snippet instructions or the npm instructions.

> If you've followed the instructions above to have a Segment team member enable the apiHost settings in the UI, you can skip the instructions in this section.

Snippet instructions

If you're a snippet user, modify the analytics snippet located inside the `<head>` of your website:

To proxy CDN settings and destination requests that typically go to `https://cdn.segment.com`, replace:

```
- t.src="https://cdn.segment.com/analytics.js/v1/" + key + "/analytics.min.js"
+ t.src="https://MY-CUSTOM-CDN-PROXY.com/analytics.js/v1/" + key + "/analytics.min.js"
```

To proxy API tracking calls that typically go to `api.segment.io/v1`, replace:

```
- analytics.load("<MY_WRITE_KEY>")
+ analytics.load("<MY_WRITE_KEY>", { integrations: { "Segment.io": { apiHost: "MY-CUSTOM-API-PROXY.com/v1" }}})
```

npm instructions

If you're using the npm library, make the following changes directly in your code:

To proxy settings and destination requests that typically go to `https://cdn.segment.com` through a custom proxy:

```
const analytics = AnalyticsBrowser.load({
  writeKey,
  // GET https://MY-CUSTOM-CDN-PROXY.com/v1/projects/<writekey>/settings --> proxies to
  // https://cdn.segment.com/v1/projects/<writekey>/settings

  // GET https://MY-CUSTOM-CDN-PROXY.com/next-integrations/actions/...js  --> proxies to
  // https://cdn.segment.com/next-integrations/actions/...js
  cdnURL: 'https://MY-CUSTOM-CDN-PROXY.com'
 })
```

To proxy tracking calls that typically go to `api.segment.io/v1`, configure the `integrations['Segment.io'].apiHost`:

```
const analytics = AnalyticsBrowser.load(
    {
      writeKey,
      cdnURL: 'https://MY-CUSTOM-CDN-PROXY.com'
    },
    {
      integrations: {
        'Segment.io': {
          // POST https://MY-CUSTOM-API-PROXY.com/v1/t --> proxies to
          // https://api.segment.io/v1/t
          apiHost: 'MY-CUSTOM-API-PROXY.com/v1',
          protocol: 'https' // optional
        }
      }
    }
  )
```

Custom Proxy CloudFront

These instructions refer to Amazon CloudFront, but apply more generally to other providers as well. Before changing the Segment Tracking API or the Segment snippet (Segment CDN) to use your new proxy, complete the custom domain proxy setup on your side to avoid any unexpected behavior.

CDN Proxy

To set up your CDN Proxy:

1. Log in to the AWS console and navigate to CloudFront.
2. Click **Create Distribution**.
3. Configure the distribution settings. In the Origin section, update the following values:
   - **Origin Domain Name**: `cdn.segment.com`
   - **Protocol**: `HTTPS only`
4. In the Default cache behavior section, configure the following values:
   - **Viewer protocol policy**: `Redirect HTTP to HTTPS`
   - **Allowed HTTP Methods**: `GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE`
5. In the Settings section, configure the following values:
   - **Alternate domain name (CNAME)**: `analytics.<yourdomain>.com`
   - **Custom SSL certificate**: Select an existing or new certificate to validate the authorization to use the **Alternate domain name (CNAME)** value. For more information, see Amazon's documentation Requirements for using alternate domain names.
6. Click **Create Distribution**.

Take note of the Domain Name for use in the next step.

Add CNAME Record to DNS

To add a CNAME record for the Segment proxy to your organizations DNS settings:

Use a name that makes it clear what you are using the subdomain for, for example `analytics.mysite.com`.

1. Use a name that makes it clear what you are using the subdomain for, for example `analytics.mysite.com`.
2. Go to your domain registrar and add a new record to your DNS of type “CNAME”.
3. Configure these values:
   - **Name**: `<subdomain_name>.yourdomain.com`
   - **Value**: The Domain Name value from CloudFront
4. Save your record. This might take some time to take effect, depending on your TTL settings.
5. Make a `curl` request to your domain to verify that the proxy works.

Tracking API Proxy

As events travel through the proxy before reaching the tracking API, set up a proxy for the tracking API so that all calls proxy through your domain. To do this, set up a CloudFront distribution that’s similar to the one in the previous section, with the exception of the Origin Domain Name:

FIELD	VALUE	DESCRIPTION
Origin Domain Name	`api.segment.io`	The domain name to which the proxy is served

Add CNAME Record to DNS

To add a CNAME record to your DNS settings:

1. Go to your domain registrar and add a new record to your DNS of type “CNAME”. This time use the CloudFront distribution for the tracking API proxy.
2. Enter values for these fields:
   - **Name**: `<subdomain_name>.yourdomain.com`
   - **Value**: Tracking API CloudFront Distribution Domain Name
3. Save your record. This might take some time to take effect, depending on your TTL settings.
4. Run `curl` on your domain to check if the proxy is working correctly.

Common issues

These are some common issues that occur for customers implementing a custom proxy. This is not an exhaustive list, and these CloudFront or Cloudflare settings may change.

Cloudflare returning a 403 error

A 403 error can mean that you’ve misconfigured your Cloudflare CDN distribution. Try one of the following options to fix the error:

1. If you have a Cloudflare enterprise plan, create a Page Rule in Cloudflare so that Segment’s CDN doesn’t refuse the requests made through the Cloudflare Proxy. If `cdn.segment.com` is another CNAME that resolves to `xxx.cloudfront.net`, you will need to use a Page Rule in Cloudflare to override the host header to match the hostname for proxy requests. For more information about overriding the host header, see Cloudflare’s Rewrite Host headers docs.
2. For customers who are not on the Cloudflare Enterprise plan, use Cloudflare Workers. Workers usually run on the main domain (for example, `www.domain.com`), but if you want Workers to run on a subdomain, like `http://segment.domain.com`, you must record the subdomain in your DNS. For more information, see Cloudflare’s Routes and domains documentation.

When creating a Worker you can use this example provided by Cloudflare in their Bulk origin override documentation with the origins set to:

```
const ORIGINS = {
"yourcdndomain.com": "cdn.segment.com",
}
```

Cloudflare CORS issue

In order to resolve a CORS OPTIONS pre-request fetch error, you must specify “Strict (SSL-Only Origin Pull)” as a Cloudflare Page rule for the `api.segment.io` proxy. Please see Cloudflare’s Encryption modes documentation for more details.

CloudFront Proxy returning a 403 error

If your CloudFront Proxy is returing a 403 error, the following change in CloudFront might resolve the issue:

```
Before:
Cache Based on Selected Request Headers: All

After:
Cache Based on Selected Request Headers: None
```

Alternatively, this setting may solve your issue:

```
Before:
Origin request policy: AllViewer

After:
Origin request policy: None
```

CloudFront CORS issue

To resolve a CORS issue, you might need to add a referrer header in the request you send to Segment. Follow AWS’s How do I resolve the “No ‘Access-Control-Allow-Origin’ header is present on the requested resource” error from CloudFront? guide, which explains how to add a referrer header.

Self-hosting Analytics.js

To reduce fetching assets from Segment’s CDN, you can bundle Analytics.js with your own code.

To bundle Analytics.js with your own code, you can:

- Use Analytics.js as an npm package.
- Use npm to install your destinations.
- Hardcode your settings instead of fetching from the CDN (Segment doesn’t recommend this as it completely bypasses the Segment source GUI).

```
// npm-only
export const analytics = new AnalyticsBrowser()
analytics.load({
 ...
 cdnSettings: {...} // object from https://cdn.segment.com/v1/projects/<YOUR_WRITE_KEY>/settings'
 })
```

Restore the API host to the Segment default

If you wish to restore the proxied API host to it’s original value:

1. Navigate to the **Source > Settings > Analytis.js tab**
2. Scroll down until you see the Host address field.
3. Under the field, there is a small blue text that says ‘Restore to a default value’. Click **Restore** and then **Save**.

Any changes made to the CDN host must be update manually in your code.

This page was last modified: 27 Jan 2025

Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

Visit our Support page

Help improve these docs!

Edit this page

Request docs change

Was this page helpful?

👍 Yes

👎 No

Get started with Segment

Segment is the easiest way to integrate your websites & mobile apps data to over 300 analytics and growth tools.

Your work e-mail

Request Demo

or

Create free account