

SEO FOR HUBSPOT CMS DEVELOPERS

Video: Introduction to SEO for Developers

Hey, it's Ben from HubSpot Academy.

As a developer, you may sometimes be tasked with improving one or multiple aspects of SEO on a website. Working with marketers and SEO professionals, sometimes known as SEOs or SEO specialists, is an important part of the work you do. It's also possible that you'll sometimes be solely responsible for optimizing a website for search engines.

Whatever the case, by learning about SEO, you can help to ensure that users can discover your site in Search and, as a result, benefit from all the development work you've put into your website. Discoverability in a search is critical because 68% of online experiences begin with a search engine, and 53% of all website traffic comes from organic search.

While it's clear that having knowledge about SEO is important, it's also important to ask: What is the role of a developer when it comes to SEO? And what are the things a developer needs to know?

Before we attempt to answer these questions, we should also note that, as a developer, you're already having a positive impact on SEO when you apply web best practices. By accounting for things like site speed, accessibility, security, and mobile-friendliness, you're already solving for SEO. In general, having a site that adheres to modern web best practices increases its chances of ranking higher in search engines.

Let's now move into a deeper discussion of what SEO is and how you can solve for it on your website. In this video, we'll answer the following questions:

- What is SEO?
- How do search engines work?
- What is the role of a developer when it comes to SEO?

SEO, or Search Engine Optimization, is the practice of increasing the quantity and quality of traffic to a website through organic search engine results.

Part of ensuring that a website is search-friendly is making sure that it's accessible and comprehensible to web crawlers. Web crawlers, or just crawlers, are bots that browse

the web in order to index its contents. Many companies and individuals, including Google, use crawlers to browse and index the web.

Because Google is the most popular search engine in the world by a large margin, it is widely regarded as an authority on SEO best practices, and much of the guidance you see around SEO is about optimizing for Google's crawlers and Google Search.

Google's main crawler is known as Googlebot. Google maintains both mobile and desktop versions of its main crawler, known as Googlebot Smartphone and Googlebot Desktop. Google also maintains other crawlers that serve different functions, but when we talk about optimizing for crawlers, we're usually referring to optimizing for Googlebot Smartphone and, to a lesser degree, Googlebot Desktop.

Let's now focus on how a web page ends up in Google's search results. What Google Search does can be broken up into 3 main stages: crawling, which also includes rendering, indexing, and serving search results, sometimes known as ranking.

In the crawling stage, Google downloads the text, images, and videos on your page. As a part of crawling, Google also renders the page and runs any JavaScript it finds. This is important for pages that use JavaScript to add content to a page. Without a rendering stage, Google wouldn't be able to see this kind of content. This is common in the case of single-page apps where the app and its contents are generated client side instead of on the server.

After crawling and rendering, there's the indexing stage, where Google processes and analyzes the page and stores it in the Google index, which is Google's database used to serve search results.

The final stage is serving search results, a stage sometimes known as ranking. This stage occurs when a user actually performs a Google search. At this stage, Google searches its index and returns what it believes are the highest quality and most relevant results.

This image from Google's documentation depicts the crawling, rendering, indexing, and ranking stages.

With these stages in mind, we might say that the goal of SEO is to make crawling, rendering, indexing, and ranking of your site as easy as possible for Google and other search engines.

Of course, developers won't usually be responsible for all aspects of SEO. But the work you do is crucial to improving your site's ranking in search engines. Let's talk about what aspects of SEO you'll usually need to account for.

SEO is often divided into three main categories: off-page SEO, on-page SEO, and technical SEO. Off-page SEO includes things that you can do off of your website to improve its SEO; on-page SEO refers to things that you can do on your site to improve its ranking, like including an optimized title tag and including alt text on images; finally, technical SEO involves additional things you can do to improve the ranking of your site, like ensuring your site is mobile-friendly; creating and maintaining a sitemap; or improving site speed.

As a developer, the work you'll do for SEO will include both on-page and technical SEO considerations. You won't likely be involved in creating SEO-friendly content or developing an overarching SEO strategy, but you may be called upon to optimize certain aspects of a website. One of the great parts about the work you'll do with SEO is that many SEO recommendations are easily measurable and straightforward to implement.

For comprehensive coverage of SEO, check out HubSpot Academy's SEO Certification. The link can be found in the resources section.

By learning about SEO, you can communicate and work with other professionals more effectively, and help to ensure that users can actually benefit from the work you've done by discovering your site in search. Check the resources section for more information on SEO and your role in it.

Video: Accounting for SEO During Development

As a developer writing code, you have a direct impact on how well your site's pages are optimized for search engines. For example, developers are usually responsible for some aspects of on-page SEO, which refers to the set of things you can do on the pages of your site to optimize for SEO.

You won't be responsible for all elements of on-page SEO, like optimizing the page's content for search engines, but you will at least be responsible for implementing some SEO best practices in code.

Developers are also often responsible for some aspects of technical SEO, which includes things like creating and editing sitemaps, using URL redirects, using structured data, and optimizing for site speed.

In this video, we'll discuss the major elements of on-page SEO that developers should be aware of. This includes: using meta title and meta descriptions on pages, using canonical links correctly, and using descriptive text in links. We'll end with some tools you can use to evaluate your site's on-page SEO.

Some other elements of on-page SEO are related to accessibility, security, and performance. For example, including alternative text is an accessibility best practice that is also an element of on-page SEO. You can find links to other Academy videos covering these elements, as well as technical SEO, in the resources section of this video.

Let's start with meta titles and descriptions. As a developer, you probably won't be responsible for creating these, but you might be the person who creates pages in the editor, so you can be sure that they're always implemented correctly during the development and maintenance of your site.

A meta title, implemented as a `<title>` tag in HTML, should always be included on a page to provide a concise and accurate description of the page's content. The general recommendation is to keep these under 55 characters. A `<title>` tag will look something like this.

The meta description, implemented as a `<meta>` tag with `<name>` and `<content>` attributes in HTML, summarizes the page's content. The general recommendation for these is 155 characters or less. A meta description will look something like this.

Meta titles and descriptions are what appear on search engine results pages, or SERPs. At the top of this SERP, *HubSpot Academy - Homepage* is the page's title, and the text underneath is the page's meta description. This highlights the importance of including these elements on your site's pages.

You can implement meta titles and descriptions in HubSpot's page editor. First, click on the **Settings** tab and enter your title in the *Page title* field. Enter your meta description in the *Meta description* field.

Let's move on now to canonical URLs. Canonical URLs are useful if you have a single page that's accessible by multiple URLs, or if you have multiple pages with similar

content. Canonical URLs allow you to tell crawlers which version of the page is canonical, that is, the version of the page that should be crawled and indexed.

To tell crawlers which version of a page is canonical, in the page editor, click on **Settings**, then scroll down and click on **Advanced Options**. Scroll to the bottom where you'll find the *Canonical URL* field. Here, you can enter the canonical URL. By default, HubSpot will add self-referencing canonical URLs to all pages and blog posts, except in the case of blog listing pages.

You can modify these defaults by clicking on the settings icon in the main navigation, navigating to **Website**, then **Pages**, and clicking on the **SEO & Crawlers** tab. Modify these options under the *Canonical URLs* heading.

Finally, let's discuss using descriptive text in links. When creating the clickable word in hyperlinks, avoid generic link text, like "Click here" or "Learn more." Instead use specific descriptions that describe the content on the hyperlinked page.

For example, here on the HubSpot Academy homepage, the main call-to-action button contains the text "Sign up for free courses" which provides additional details about the linked page. We can assume that by clicking on this link, we'll be taken to a page where we can sign up to take Academy's courses.

Alright, let's now briefly discuss some tools you can use to evaluate on-page SEO, including some within HubSpot itself. For example, when creating a page, in the page editor, you can click on the **Optimize** tab to view SEO recommendations for the page.

Within HubSpot, you can also scan your entire site for SEO issues. In the main navigation, click on **Marketing**, then **Website**, and then **SEO**. Next, click on **Scan New URL** and enter your site's URL. After the scan is completed, you'll receive a list outlining recommendations in multiple categories including on-page SEO, performance, and security. Each recommendation will indicate its SEO impact, implementation difficulty, and who might typically address the recommendation – for example, a developer or marketer.

Check out our Knowledge base article entitled, "Understand SEO Recommendations" to get a complete breakdown of HubSpot's SEO recommendations.

You can also use Google Lighthouse within Google Chrome's Developer Tools to scan a page for SEO recommendations. Google's article on SEO audits breaks down Lighthouse's recommendations. There are many other tools available that you might

come across when working with SEO, including Screaming Frog, Semrush, Google Analytics, and Google Search Console.

You can also integrate Google Search Console with HubSpot by downloading the Google Search Console App from the marketplace, if you have a compatible plan. Check the resources section to learn more about these and other tools.

As a developer, understanding how to implement SEO best practices is important because so much of the work you do has the potential to impact search ranking. By following the steps outlined in this video and using the many SEO tools available, you can create search-friendly websites that help your site get noticed.

Video: Prioritizing Mobile-Friendly Websites

Having a website that is optimized for mobile devices is a must in today's world. Mobile web usage has only increased over time, and, according to Statista, mobile devices were responsible for about 59% of global website traffic in the second quarter of 2022.

Additionally, according to Google, most users access Google search with a mobile device. For this reason, Google has enabled mobile-first indexing for all new websites. This means that Google primarily uses its smartphone agent to crawl and index pages.

Thus, the mobile version of your site is what is appearing in search results. With the increase in mobile usage over time and the advent of mobile-first indexing, optimizing your site for mobile is no longer optional but an important and critical part of your site's success.

In this video, we'll cover some ways you can create mobile-friendly sites and some tools you can use to make sure that your site is optimized for mobile.

We'll discuss: responsive design, using the viewport meta tag, using legible font sizes, using appropriately sized tap targets, and how to account for mobile-first indexing.

Let's start with responsive design. When implementing a responsive design, you'll use HTML and CSS to make your website look and function well on all devices, including desktop, mobile devices, and tablets.

Implementing a responsive design is the first step in optimizing your website for all devices. This process includes using media queries, responsive text sizes, and

responsive images to change the appearance of your site based on the device of your users. Responsive design is a broad topic and comprehensive coverage is outside the scope of this video, but it's important to note.

Let's now move on to three on-page SEO elements related to mobile-friendliness that are part of Lighthouse's SEO recommendations.

The first is to include the viewport meta tag in the head of your HTML. Most HTML boilerplate will include this tag, but it's still good to know since the mobile version of your site won't work correctly without it. This one is automatically included in your HTML in HubSpot when you use the `standard_header_includes` variable in your templates, like here in the CMS Boilerplate's `base.html` file.

Using legible font sizes is also important. Google recommends using a font size of at least 12px on at least 60% of the text on your page.

Finally, sizing tap targets appropriately is an important element of optimizing for mobile. Tap targets are the parts of your page that users on touch devices can interact with, like buttons, links, and form elements. Google recommends tap targets at least 48px wide and 48px tall and to be at least 8px apart from each other.

You can use padding to increase the size of tap targets without changing their appearance, and use margin to increase the space between tap targets.

Google also provides these general guidelines when creating a mobile site: Use the same content on your mobile site as on your desktop site. Remember that your mobile site will likely be used for almost all indexing, so it's what will appear in search results.

Also, make sure you're using the same meta tags, like meta descriptions and meta robots tags, on both the desktop and mobile versions of your sites. Check out Google's article on "Mobile-first indexing best practices" for more details.

Google also provides a mobile-friendly test that can help you identify issues that are impacting your site's mobile experience. We've included a link to this tool in the resources section.

Your mobile site should get the same level of attention as your desktop site. Use the tips and tools in this video to make sure that you're accounting for your mobile audience when designing and developing your website.

Video: What Is Structured Data?

Why is structured data important for SEO?

Structured data helps Google to better understand the contents of your pages and what your website is all about so it can provide more relevant, helpful results to searchers. Google's goal is to provide results that match a searcher's intent – results that are contextualized to their particular query, location, and preferences. Since one of the main goals of SEO is to optimize your site for search engines to more effectively crawl and index, structured data is another tool in your SEO toolset that helps you accomplish this.

What is structured data?

A standardized format for providing information about a page and classifying the page content (Source: Google).

Structured data is a markup language.

A markup language is computer code that can be written, read, and understood by humans. HTML is an example of a common markup language that most marketers have at least some familiarity with.

Like HTML, structured data communicates content in an organized way so that search engines can display that content in an attractive manner in SERPs.

When talking about structured data, the phrase "marking up" often comes up.

"Marking up" structured data means creating the structured data code.

How Google Uses Structured Data

"Google uses structured data that it finds on the web to understand the content of the page, as well as to gather information about the web and the world in general."

Source: Google Search Central

Google uses the information it finds in structured data to create rich results in SERPs.

Structured data makes it easier for Google to understand what images and content are about so it can accurately display that content in search results. The way in which these results are visually displayed are called "rich results" or "rich snippets." You can learn more about rich results in the resources section below.

There are no guarantees that Google will create a rich snippet of your page even when you've added structured data.

Using structured data only makes a web page eligible for rich results.

Structured data is a tool you can use to tell Google detailed information about the contents of your web pages.

The primary source of structured data markup is Schema.org.

What is Schema.org?

Schema.org is the organization that creates the standardized language and rules used for structured data.

You can search Schema.org to find the specific markup you need for your particular page.

Schema.org contains a large collection of different code examples which website owners can then adapt to their particular preferences. For example, if you have a website that sells computers, you can find code on Schema.org that allows you to show the price, stock levels, reviews, and ratings for your products.

Schema.org is accepted and understood by Google, which is why we recommend using it.

Google supports adding schema markup using a few different formats.

The recommended format is called JSON-LD.

What is JSON-LD?

JSON-LD is a script that can be placed within a web page to communicate structured data to search engines.

There are other formats for adding structured data, but JSON-LD is recommended.

It's not only Google's preferred format, but it's also less complicated than other methods. That's because with JSON-LD, all you need to do place the script somewhere in the code of your web page.

Three Ways to Implement Structured Data

Work with a developer

Use a plugin

Add it manually

Structured data is primarily implemented by developers.

Since it involves writing and editing code, many marketers prefer to enlist the help of a web developer to do this job. However, if you're a marketer without any development resources, you can still take advantage of the benefits of structured data. Not all Schema markups are built the same: some are simpler to implement on your own, while others require more work with extra properties that are either required or recommended. A good guideline to follow is: the more recommended properties there are in a markup, the more likely you'll want to work with a developer.

Add structured data markup to your site using plugins.

If you use WordPress as your CMS, there are multiple plugins available that allow you to add structured data to your pages automatically.

If you don't have a developer or the option to use a plugin, you can also add schema markup to your pages manually.

This does require a bit of technical skill, so if you're very uncomfortable with coding then we recommend outsourcing to a developer. If you do feel comfortable getting your hands in a bit of code, implementing schema markup can be quite straightforward.

Steps to add structured data to a page:

In the head HTML of your web page, add a script element set to JSON-LD.

Inside the script element, tell Google you're using Schema.org structured data.

Based on the type of content you're describing, tell Google which kind of structured data you're using.

Add all the required and recommended properties to give Google more information about the content being described.

Here's an example of completed schema markup. (Source: Rice Media)

Adding structured data may sound complicated, but fortunately there are tools to help.

You can use an online schema generator to do a lot of this work for you: simply pick the type of markup you want to create, enter the relevant information, and copy and paste the code into the head HTML of your page. We've linked a schema generator in the resources section below which will help speed up the process and make it much easier.

Google has a helpful tutorial that guides you step-by-step through the process of manually adding structured data markup to a page, which is also linked in the resources section. We recommend going through that tutorial so you can get a feel for adding structured data before implementing it on your own site.

Test Your Structured Data

Make sure to test your page by using Google's Rich Results Test and Structured Data Testing Tool.

Using Google's structured data testing tool, make sure you've included the correct properties and you don't have any errors.

You can either enter a code snippet if your page is still in progress or provide a URL for a published page.

Google's Rich Results Test tool will tell you if you've included all the necessary information to make your page eligible for rich results. You can also use either a code snippet or URL for this test. One helpful feature of this tool is that it allows you to preview how your page will appear in search results.

Chapter: Structured Data

To learn more about using structured data in HubSpot, check out our blog post: [How to Use Structured Data in HubSpot to Improve SEO](#).

To learn about how to automatically enable structured data on your HubSpot blog posts, see this community forum post: [\[Beta\] Setting to apply BlogPosting JSON-LD schema to blog posts](#).

For more on structured data for web developers, check out this video: [Structured data for web developers | Search Central Lightning Talks](#).

Video: Preventing Content From Being Crawled and Indexed

By default, the pages on your website can be crawled and indexed by search engines and other bots. But there are a few ways to prevent your content from being crawled: [using a robots.txt file](#), using the noindex meta tag, password-protecting your page, or removing a page entirely.

Let's start with the robots.txt file. This file tells search engine crawlers which URLs they can crawl on your site. This is mostly to stop your site from being overloaded with

requests by crawlers. Using robots.txt is not a way to keep pages out of search engine results.

You can use a robots.txt file to manage crawler access to entire pages, media files, and other resource files like scripts or stylesheets. It's important to note that not all crawlers will respect the instructions you provide in your robots.txt file. Additionally, a page disallowed in robots.txt can still appear in search results if other sites link to it.

To view and edit your robots.txt file in HubSpot, click the **settings icon** in the main navigation bar, in the left sidebar menu, navigate to **Website, Pages**, and then click on the **SEO & Crawlers** tab. Here you can edit your robots.txt file.

In a robots.txt file, you'll enter a field, a colon, and a value. Possible fields include user-agent, disallow, allow, and sitemap.

The user-agent field is used to identify the crawler that the instructions are for. The disallow field is used to tell the user-agent which URLs, in the form of relative paths, it cannot crawl. Relative paths must begin with a forward slash and are relative to the root of your site.

The allow field is used to allow crawling of certain pages that you may have disallowed previously in the file. Finally, the sitemap field is used to tell crawlers the location of your site's sitemap in the form of an absolute URL.

Comments in a robots.txt file should start with a hash symbol. User-agent and allow and disallow fields will appear in groups. A group begins with a user-agent field and value, and is followed by any number of allow and disallow directives. A robots.txt file can contain more than one group.

Let's consider an example. In this robots.txt file, we'll first enter the user-agent. To do this, we'll type *user-agent*, colon, and then an asterisk. An asterisk means "all user-agents."

You can also enter different values for different user-agents. For example, Googlebot, BingBot, and LinkedInBot are all names of bots that crawl websites. This value is case-insensitive.

We'll follow this up with a disallow directive. Type *disallow: /* to disallow crawling of every page on the site. Typing */cms* will tell the crawler to not crawl this page and also any pages that begin with */cms*, which allows you to block multiple pages with a single directive. The path values after the colon are case-sensitive.

Check out Google's article on the robots.txt specification for more information on robots.txt syntax and how their crawlers interpret this file. The link can be found in the resources section.

Remember that a page that's disallowed in robots.txt can still be indexed. To prevent a page from appearing in Google, use the noindex meta tag: `<meta name="robots" content="noindex">`. This will prevent Google from indexing the page, though there's no guarantee that all search engines and crawlers will recognize or respect this tag.

A quick warning. A page with a noindex meta tag should never be included in your robots.txt file. If a page appears in robots.txt, Google will not crawl that page. So even if the page has a noindex meta tag, Google will never see that tag, meaning the page could still appear in search results.

In short, either include a page in robots.txt, or add a noindex meta tag to the page, but never do both.

To add this tag to a page, in the page editor, click **Settings**, scroll down and click on **Advanced Options**, and paste this tag into the **Head HTML** field.

Remember that not every crawler will respect a noindex meta tag, so if you want to block your page from all crawlers, you'd need to password-protect, unpublish, or delete the page.

To password-protect a page using HubSpot's CMS, in the page editor, click on **Settings**, scroll down to **Advanced Options**, and under *Control audience access for page*, select **Private - password required** to set the page's password. Check our knowledge base article on password-protecting a page for the most up-to-date information on what products and subscription levels have access to this feature.

And that's it! Work with a marketer or SEO to determine when to block crawler access to pages and files. Also check the resources section for more on the robots.txt file and the noindex meta tag.

Video: Creating and Managing URL Redirects

URL redirects do exactly what you might think: they allow you to redirect an old URL to a new URL. URL redirects help you make sure that both users and crawlers can find the information they're looking for on your site without encountering a 404 page.

You can use a redirect when you: are moving your site to a new domain, have multiple URLs for your site, are merging two websites, or have moved a page to a new URL.

Redirects can be broadly categorized into two types: permanent and temporary. Permanent redirects signal that a page has been moved permanently. Use a permanent redirect if you want the new URL to be shown in search results.

Temporary redirects indicate that a page has been moved temporarily. Using a temporary redirect ensures that the old URL will appear in search results for a longer time.

There are a few different kinds of permanent and temporary redirects, but we'll just focus on 301 and 302 redirects. 301 and 302 refer to the status codes returned by the server when the URL is accessed. A 301 status code indicates a permanent redirect, and a 302 status code indicates a temporary redirect.

To add redirects in HubSpot, click the **settings icon** in the main navigation bar. In the left sidebar menu, navigate to **Website** and then **Domains & URLs**. Finally, click the **URL Redirects** tab.

Here, you can view existing redirects and add new URL redirects. You can import a csv file to add multiple redirects at once, add a single redirect, and also export your redirects as a csv file.

Let's add a new URL redirect by clicking **Add URL redirect** in the upper right. Choose **Standard** to add a traditional Permanent (301) or Temporary (302) redirect.

Enter the original URL you'll be redirecting from in the *Original URL* field, and the new URL you're redirecting to in the *Redirect to* field.

Then choose your redirect type. Besides a *Permanent* or *Temporary* redirect, you can also choose **Proxy**, which will show a new page's content without redirecting to that page - **the URL field will remain the same**. This type of redirect is often used to share a file that's hosted on a specific directory outside your main site.

You can also select the redirect's priority and other options. Check out our knowledge base article on "Creating and Managing URL Redirects" for more information.

In this *Add URL redirect* section, you can also add a flexible redirect by choosing **Flexible**. Using flexible redirects, **you can add variables** to both your original and

new URLs. This makes redirecting multiple pages much easier. Note that only certain products and subscription levels have access to flexible redirects.

For example, let's say you're moving your entire blog to a new subdomain.

If your blog was originally located at `https://example.com/posts` and you wanted to redirect every blog post to your new subdomain at `https://blog.example.com/posts`, you could use a flexible redirect. This way, you wouldn't have to create a redirect for every single blog post.

In this example, you would include a variable in the original URL field, like this, `https://example.com/posts/*rest-of-url`. The asterisk before `rest-of-url` denotes a variable at the end of a URL.

Then, in the redirect field, you'd want to enter `https://blog.example.com/posts/{rest-of-url}`. `Rest-of-url` in curly brackets indicates that it's a variable at the end of a URL. Of course, the variable names in the original URL and redirect fields have to match in order for a flexible redirect to work.

With this redirect, all blog posts should now redirect to their version on the new subdomain, as long as they have the same value at the end of the URL.

For example, all of these example posts will redirect to their version on the new subdomain:

- `https://example.com/posts/post-1` redirects to `https://blog.example.com/posts/post-1`.
- `https://example.com/posts/post-1` redirects to `https://blog.example.com/posts/post-2`.
- `https://example.com/posts/post-1` redirects to `https://blog.example.com/posts/post-3`.

Using flexible redirects, you can also create variables for individual components of a URL, like a year value that appears in a URL, redirect specific query parameters to another page, or use a combination of all of these to create complex URL redirects.

Check out the knowledge base article on [Setting up a flexible pattern URL redirect](#) for more information on creating these redirects, and for the most up-to-date information on what products and subscription levels are needed to create these kinds of redirects.

Creating redirects is important to ensure that your users and crawlers don't end up on your 404 page and that the most up-to-date information appears in search results. Check out the resources section for more information on URL redirects.

Video: Working With Sitemaps

Sitemaps are XML files that provide search engines information about the structure of your site. You can use sitemaps to tell search engines about what pages and files you think are important on your website, which might help them crawl your site more efficiently.

Google can normally crawl your site fairly well without a sitemap if your site has 500 pages or less and your pages are properly linked. Properly linked pages are pages that can be accessed through links on your site, either through your site's navigation or in the body of your site's pages. Still, Google says that, in most cases, your site will benefit from having a sitemap.

Sitemaps are also useful if your site contains a lot of media content, like videos and images, or if it has a lot of news articles. You can help Google find and understand this kind of content by including it in your sitemap.

Using a sitemap will not guarantee that all items will be crawled. Similarly, removing items from your sitemap won't prevent them from being crawled and indexed. Check out our video on "Preventing Content From Being Crawled and Indexed" for more on how to do that.

If you're using HubSpot's CMS, a sitemap is automatically generated for your site. It includes live pages and blog posts, but not landing pages. HubDB dynamic child pages will be included in your sitemap only if the parent page is in the sitemap. Also note that these dynamic child pages will not appear in the sitemap within HubSpot, but they will appear in the sitemap XML file.

You can always check your site's XML file by adding /sitemap.xml to the end of your site's URL and navigating there in the browser. For example, visiting *academy.hubspot.com/sitemap.xml* shows us the Academy website's sitemap.

To edit your sitemap in HubSpot, click on the **gear settings icon** in the main navigation and in the left sidebar menu, navigate to **Website**, and then **Domains & URLs**. Finally, click on the **Sitemap** tab. Here you can view your XML sitemap, add and remove pages from your sitemap, and update options for pages. Work with an

SEO or marketer to determine what changes you might want to make to your sitemap.

Once you're happy with your sitemap, you can make it available to Google in one of three ways: using Google Search Console, using Google's ping tool, or by including it in your site's robots.txt file.

Include the following line anywhere in your robots.txt file to let crawlers know about your sitemap when they land on your site. Sitemap: <Sitemap absolute URL>. Google's crawler will notice it the next time it visits your site.

Using and submitting your sitemap is a useful way of letting search engines know which content you'd like them to prioritize when crawling your site. Check the resources section for more information on working with sitemaps.