

Video 1: Introduction to the HubSpot CMS for Developers

Hi there folks. Welcome to the CMS for developers course. I'm Jeff Ausura. I teach web development topics for HubSpot Academy, and I've been building websites on various CMSs for about 10 years.

The Hubspot CMS is a SaaS CMS with a special emphasis on enabling marketers. It includes all the tools you and your team need to build and maintain web assets including a professional IDE, asset management with a lightning-fast CDN, and a full-featured editor to quickly create and optimize web pages, blog posts, and emails.

The HubSpot CMS is designed to strike an optimal balance between flexibility for marketers and convenience and control for developers. It allows developers to build complex sites and hand them off to marketers with a minimum amount of startup and maintenance time. This reduces costs for both marketers and developers.

Throughout this CMS for developers course, we'll cover the tools and topics that are used in regular professional level CMS development work. Before you start working through this content, it's important to note: This course is for experienced web developers. Most of the topics in this course require a professional-level understanding of HTML, CSS, and JavaScript to fully comprehend.

That being said, even if you aren't a full-fledged web developer, feel free to use any or all of the content in this course for reference and context. There are many topics included here that can help marketers and designers decide how to approach projects, delegate responsibilities, and communicate with team members and contractors.

All the lessons and videos in this course have been designed to balance general information with detailed instruction on implementation. Whenever possible, we've tried to distill this content down to the most important details to help you get up to speed quickly. Feel free to consume these materials in whatever order you find most helpful. Also, make sure to check out the resources section of each lesson for links to documentation and important community channels, especially the Slack group for fellow CMS developers. As you know, web development is a deep topic, and no amount of instructional content can cover every last thing that comes up while building assets.

Finally, let us know what you think as you check out these videos and chapters. This is a work in progress, and we're always looking for opportunities to refine and iterate to maximize the value of this training.

Video 2: A developer's tour of the HubSpot CMS design manager and page editor

OK folks, let's talk about the two most important tools in the CMS. The design manager and the editor are the heart and soul of the CMS. These are the tools for generating templates and for building pages from templates.

Let's start with the design manager. The design manager is an IDE, or integrated development environment, where developers create and maintain templates, modules, CSS, and Javascript files. For web developers, this is the most important part of the CMS and the place where you'll spend most of your time.

Let's briefly walk through the UI of the design manager. On the leftmost side of the design manager, there's a finder for creating and organizing assets like templates and modules. The finder behaves a lot like a file system in a typical computer OS. You can create new files, clone files, and organize them in folders.

One important aspect of the finder is the "Actions" menu. This menu gives you access to some important file behaviors such as copying between portals, accessing revision history, and finding dependent assets.

To the right of the finder is the main workspace of the design manager. This area is contextual depending on the asset you're editing. If you're working on a drag and drop template, the workspace consists of a drag and drop editor and an inspector for working with individual components. If you're working in a CSS or JS file, this entire area becomes a text editor with syntax highlighting and error reporting. This text editor implements CodeMirror. Check out the resources tab for a link to their keyboard shortcuts documentation.

Also, check out some of our other videos on specific topics like templates and modules to get more in-depth details on using the design manager in these contexts.

Now that you have a basic understanding of the design manager, let's talk about the editor. The editor is where content editors build pages, emails, and blog posts from templates. To be clear, there's no single editing experience for website pages, emails, and blog posts. Each of these is accessed separately in the main navigation, and the editing UI varies slightly between them. That being said, the same basic relationship between templates and published content exists for each of these, so from a developer's perspective, it's helpful to reduce this relationship to templates and "the editor."

The main section of the editor UI for all file types consists of an inline editing area where editors can intuitively flow content into templates. As the template builder, you have a significant degree of control over how editors interact with this editing area. The modules you decide to use and the labels you apply to these modules can help make the editing experience more intuitive. Check out my lesson on templates for more information on this topic.

To the left of this area is a utility bar that gives quick access to preview options, a list of all the modules used in the template, and suggestions for optimizing the page. Depending on the type of file, there are also options for things like testing emails in different email clients and A/B testing.

Above the editing area is a tab-style navigation where editors can access things like settings and publishing schedules. The settings tab is especially important to take note of here. There are many settings in here for things like page metadata and even additional code snippets that you may want to consult with your editors about.

This brings me to my final point: The design manager and editor are optimized for creating templates and publishing pages respectively, but the actual execution of these activities isn't always clearly divided between web developers and marketers. It's up to everyone involved to define roles and assign responsibilities. As the web developer, it's in your best interest to work proactively with your editors to hand off intuitive editing experiences and establish conventions and guidelines for the technical aspects of creating and maintaining pages.

Video 3: A developer's tour of the HubSpot CMS file manager

OK, let's talk a little about the file manager. The file manager is where you and your content editors will store files like images, PDFs, and videos. There are a ton of supported file formats, and it's unlikely that you'll run into something you can't store there. That being said, even though it's technically possible, it isn't recommended to store CSS and JS files in the file manager. The vast majority of the time, you'll want to manage these with the design manager instead.

The file manager is accessed via the main menu by choosing "Marketing" and "Files and Templates" and then "Files." The file manager behaves like a typical file and folder management system. You can create folders and subfolders to organize your assets however you'd like. On top of this baseline behavior, there are a few aspects of the file manager that are of special importance to web developers building in the CMS. Most of these are related to the file details for individual assets.

The file details pane surfaces helpful file data like updated date, dimensions, and file size and gives you quick access to a file's URL. Perhaps most importantly, it lists all the templates and content where this file is used. This can be an invaluable feature for maintenance work and can save you a ton of time and help avoid disaster when updating assets used widely across a site.

Below these file details are buttons for replacing, archiving, cloning, and deleting files. The "Replace" button can be especially handy. Use this to swap out an asset while preserving its relationship to templates and other content.

Also on the topic of file relationships to templates and other content: It's important to note that the file manager maintains your links to assets even when you move files around. So you can feel free to reorganize your content without worrying about broken links.

Finally, there's also a stock images section of the file manager. These images are part of a partnership with Shutterstock. Check out the resources section for links to usage details, but essentially you're free to use these as you'd like inside of your HubSpot web assets.

Video 3: A developer's tour of settings in the HubSpot CMS

The settings section of the CMS is frequently overlooked when folks are first getting familiar with the toolset. That's unfortunate because this is an extremely powerful part of the CMS that connects a lot of the moving pieces. As a web developer, you should make it a priority to familiarize yourself with all of the available settings. Many aspects of templates and pages will pull data from settings, so you'll want to make sure you're aware of these connections for your own work and also to support content editors. Many of these settings are global, so they can be simultaneously powerful and dangerous.

Let's take a moment to step through the sections of settings that are most important for web developers. Also, check out our other CMS videos for more detailed information on configuration settings for system and blog templates.

OK, it only seems appropriate to start with website settings, so let's go there first.

Access your HubSpot account settings by navigating to the upper-right corner of your account and clicking the settings gear icon. The settings for all web pages can be found under website in a subsection called pages. There are also settings for blog and navigation in the website section, which we'll cover later.

There's a ton of stuff in website settings, and it would be exhausting to talk about each and every option, so let's just highlight a couple of things. The site header and footer sections in the "Templates" tab is an optional place to globally add code to the head and bottom of every website page. Like most settings, you can edit this for all domains or specific domains, but either way, this is a prime example of something that can be simultaneously powerful and dangerous. Keep this section of website settings in mind especially if you're inheriting sites that you didn't build since this can be a common source for troubleshooting things that can't be solved at the template level.

To the right of the "Templates" tab, the "Branding" tab contains global settings for logo and favicon. Like many site settings, there's a module and HubL tag for using this logo throughout websites. It's up to you to decide exactly how to implement logos throughout your templates, but you should familiarize yourself with this global option and the template and page-level options as well. This is a common source of confusion for page editors and an area where you'll want to establish some clear conventions with your co-workers and clients.

Moving on to blog settings— the templates tab here contains another set of header and footer HTML fields. These are separate from the website page snippets. This is worth noting as a general organizational convention for the entire CMS: There are varying divisions for global content, and website and blog pages generally have their own global controls. Perhaps the starkest difference is the assignment of templates themselves. Blog templates are assigned for an entire blog in this area of settings. Check out our videos on blog templates for more details on this relationship.

OK, let's leave the "Website" section of settings and take a look at email. These settings can be found under the "Marketing" section. There's a ton of stuff in here that can be very helpful for building sophisticated email templates that help editors spin up consistent emails very quickly. The "Subscriptions" tab here is probably the most important thing to take note of. This is where you'll connect your subscription preferences templates to the HubL tokens that are required in email templates for CAN SPAM compliance. If you're going to be working with email templates, make sure to check out the resources tab for more information on this important aspect of that process.

OK, let's take a quick look at "Domains & URLs." This is where you'll configure domains and create redirects. If you're coming to HubSpot from a CMS on the LAMP stack, this is where you'll take care of the things you may have used an htaccess file for. Either way, it's worth mentioning that this is another area where the CMS makes things easily accessible to both web developers and content editors.

This segways nicely into the last section of settings we'll cover in this video. The "Users & Teams" section can be very handy for controlling the access that individuals and teams have to various parts of the CMS. For instance, you may want to disable administrative access for users who only need to contribute content. Another related common configuration is allowing users to write content but not publish content.

Once again, you'll want to work with your clients and co-workers to determine appropriate levels of access for everyone using the CMS. As you may have noticed, this is a common theme for all the settings we've covered in this video. The more proactive you are about establishing roles and conventions for these administrative tasks, the easier your site management will be.

Video 3: A developer's tour of navigation options in the HubSpot CMS

OK, let's spend a little time looking at creating navigation in the HubSpot CMS. Technically speaking, there are many ways to create navigation, but there are two built-in options that you should probably consider first, especially if you're just getting started with this CMS.

The two built-in navigation options are modules called menu and simple menu. When you're building templates, you're free to use whichever of these suits your needs. Let's take a brief look at the differences.

Simple menus are, much like their name implies, simple. They only support one level of navigation, and they're created and edited in pages or templates. This means that they aren't usually a great option for navigation that will appear on multiple pages in a site. The main strength of simple menus is giving content editors a consistent and intuitive interface for building a list of links.

In contrast to simple menus, the standard menu module in HubSpot supports as many levels as you need. This is the module typically used to build site navigation. These menus cannot be edited in pages or templates. Menus are created and managed in settings. They're added to templates and can be selected at either the template or the page level.

Navigation can be found in the "Website" section of settings. The interface for creating new menus and adding and editing items is intuitive, but you'll want to get familiar with the process and the available options. This topic can sometimes be confusing for less technical folks, so you'll want to make informed decisions about who will build and maintain menus. Also, options like navigation items without links and with extra URL parameters can be used to build a variety of menus, but some complicated menu designs may not be possible to implement with this built-in module.

If you find yourself needing to build a menu that doesn't seem possible using the default navigation, you might consider using HubDB to create totally customizable menu structures. This can be a handy technique for menus that need complex language conditionals or that contain images or videos. Check out the resources tab for links to HubDB education and documentation.

Also, be sure to check out our videos on building HubSpot templates. Like logos, menus are frequently global site content, and you'll want to make informed decisions about how you'll build them into your templates to keep things clear and consistent for your fellow web developers and your content editors.