Splunk Developer's Guide
Second Edition

This book will be your guide to develop a fully-fledged Splunk application—all the way from designing, to implementing, to publishing it. We will discover the fundamentals of building a Splunk application and then move on to creating one. During the course of the book, we will cover application data, objects, permissions, and more. After this, we will show you how to enhance the application, including branding, workflows, and enriched data. Views, dashboards, and web frameworks will also be covered.

This book will showcase everything new in the latest version of Splunk—including the latest data models, alert actions, XML forms, various dashboard enhancements, and visualization options (with D3). Finally, we will take a look at the latest Splunk cloud applications, advanced integrations, and development as per the latest release.

Who this book is written for
This book is for those who have some familiarity with Splunk and now want to learn how to develop an efficient Splunk application. Previous experience with Splunk, writing searches, and designing basic dashboards is expected.

What you will learn from this book
- Implement a modular input and a custom D3 data visualization
- Create a directory structure and set view permissions
- Create a search view and a dashboard view using advanced XML modules
- Enhance your application using event types, tags, and macros
- Package a Splunk application using best practices
- Publish a Splunk application to the Splunk community

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Application Design Fundamentals'
- A synopsis of the book’s content
About the Author

Kyle Smith is a self-proclaimed geek and has been working with Splunk extensively since 2010. He enjoys integrating Splunk with new sources of data and types of visualization. He has spoken numerous times at the Splunk User Conference (most recently in 2014 on *Lesser Known Search Commands*) and is an active contributor to the Splunk Answers community and also to the #splunk IRC channel. He was awarded membership into the SplunkTrust as a founding member. He has published several Splunk Apps and add-ons to Splunkbase, the Splunk community’s premier Apps and add-ons platform. He has worked in both higher education and private industry; he is currently working as an integration developer for Splunk’s longest running professional services partner. He lives in central Pennsylvania with his family.
Splunk is awesome. Not only can you consume virtually any data with it, you can also extend and integrate Splunk with virtually any external system. Splunk uses sets of configurations that are referred to as applications or add-ons, which is the primary focus of this book. Leveraging these applications and add-ons is what gives Splunk its unique ability to extend, learn, analyze, and visualize information.

Splunk helps users to determine the root cause of a failure, a quick overview of system health, and dive deep into SQL statements and messages, just to name a few. The aggregation and centralization of log and event management is a growing trend in the Big Data space. By leveraging the combined intelligence gathered from correlating disparate sets of data, businesses or individuals can make data-based decisions. This book will help a Splunk developer, or even just a curious end user, to develop different methods of consuming new data, design new types of visualization, or even just offer tips and tricks that help the software development lifecycle.

Overview of what this book isn't
Most developer guides will tell you what their book is and/or does. We aim to explain what this book isn't, and allow you to fill in the rest with your imagination! Thus, proceed to this list:

- Will not cover Splunk basics
- Will not cover creating dashboards via the GUI (other than HTML)
- Will not discuss how to code in Python
- Will not discuss statistics
- Will not cover SDKs
- Will not discuss making beer
Preface

Splunk basics will not be covered. These include concepts such as searching (finding data, using timecharts, stats, some eval commands, and so on), reporting (making basic pie charts or line charts via the GUI), data inputs (basic file monitoring, TCP and UDP inputs, Splunk forwarders, and so on), and configurations (GUI and web-based configuration editing), to name a few. Creating dashboards via the GUI? Nope. Python will be discussed and sample code will be provided, but this book will not cover the nuances of the code, nor will it teach you Python syntax. We will not cover statistical computation, other than how to practically apply some basic math to create value-based visualizations. We will not cover using the SDKs (software development kits) being used in custom Splunk applications that are external to Splunk (for example, Angular, PHP, .NET, and others). These are out of the scope of this book. Free as in beer? Nope, the choice of hops, starch, and oak-barrel aging for the creation of beer will not be discussed, but rather consumed during the writing and/or reading of this book.

Unless otherwise stated, this book uses Splunk version 6.3 as the development environment.

What this book is

This book will guide you through many the different areas of Splunk App and add-on creation. We will start by looking at the design aspects of an App or add-on, how to create them, what knowledge objects are available for use within the App, ways to enhance your App with metadata and external data, and some basic views and dashboards. From there, we will move into the Splunk Web Framework, modular inputs, jQuery, web framework programs, and then packaging and publishing Apps and add-ons. At the tail end, we will highlight some areas of the Splunk community that prove to be very useful.

Assumptions

There are a few basic assumptions that we are going to make. Having purchased or otherwise obtained this book, we assume that you are interested in developing with Splunk, and have a basic understanding of Splunk and how to navigate around the software. Knowledge of saving searches, reports, and basic dashboarding is a must, since most concepts and examples will be built upon the basics. We also assume that you have basic knowledge of HTML, CSS, JS, and some XML. Here, XML will be limited to the Splunk XML framework specifically. We would also recommend you to have knowledge of, or proficiency in, Python, RequireJS, and other web technologies such as Bower, npm, and Gulp. We will demonstrate how to use these web technologies within a Splunk application.
What this book covers

Chapter 1, Application Design Fundamentals, discusses fundamental questions and considerations before diving into an App or add-on configuration.

Chapter 2, Creating Applications, discusses the basic methods of App and add-on creation, along with an explanation of the structure of an App or add-on.

Chapter 3, Enhancing Applications, discusses a few different configurations that help to enrich your data with Splunk knowledge objects, along with some basic App and add-on branding guidelines.

Chapter 4, Basic Views and Dashboards, goes through the basics of SimpleXML dashboard creation and development.

Chapter 5, The Splunk Web Framework, details the various SplunkJS Stack components and shows examples of how to use them within an HTML dashboard.

Chapter 6, Advanced Integrations and Development, reviews modular inputs, data models, the KV Store, and modular D3 visualizations.

Chapter 7, Packaging Applications, lists the items needed to package an App or add-on, in order to get it ready for publishing.

Chapter 8, Publishing Applications, explains step by step how to upload an App to Splunkbase, and includes some information on the great support community.
Hello there, Splunk developer! If you are like us, we know you have a love of Splunk and all of the endless possibilities that present themselves! The Big Data world is exploding around us, and it always feels like a tireless battle when keeping up to date with advances in technologies, platforms, and concepts. Here, we will discuss none of those. This book is dedicated solely to Splunk and the development of applications for Splunk. Onward and upward!

**What is a Splunk application?**

All that being said, let's talk Splunk applications. A Splunk application is nothing more than a structured set of configurations and assets used to achieve an end goal of data collection, indexing, and visualization. Furthermore, in order to create a valid Splunk application, you must include the ability to navigate. Without navigation within the application, you would be working on an add-on. According to Splunk, applications:

- Contain at least one navigable view
- Can be opened from the Splunk Enterprise home page, from the App menu, or from the Apps section of Settings
- Focus on aspects of your data
- Are built around use cases
- Support diverse user groups and roles
- Run in tandem
• Contain any number of configurations and knowledge objects
• Are completely customizable, from frontend to backend
• Can include Web assets such as HTML, CSS, and JavaScript

This is taken from http://docs.splunk.com/Documentation/Splunk/latest/AdvancedDev/AppIntro.

Why applications?
Applications allow us to quickly share configurations, focus on the context of available data, limit data access to specific individuals or groups, and organize similar dashboards and views into a cohesive presentation of data within Splunk. Sharing applications can be as easy as just zipping it up and sending it out. Splunk applications could be said to be open source, due to the fact that almost all of the configurations, custom scripts, and any other knowledge object contained within the applications, are readable on the filesystem. This allows for customization for an individual instance while maintaining an overall master configuration.

Definitions
To get started, we should define a few naming conventions typically used when naming applications. Note that while we will use these naming conventions as the best practice, your application can really be named anything at all, which may conflict with other applications of the same name, or violate Splunk usage agreements or publishing guidelines. In particular, the name Splunk cannot be present in your application or add-on name. Additionally, in the past, Splunk has referred to add-ons as technology add-ons, and has since moved to just add-ons.
The following list of add-on types is our way to distinguish the different uses of each add-on:

• **Applications**: Applications could be named anything, as long as they are relevant to the content of the application and don't contain the name Splunk.
• **Domain add-ons (DA)**: Domain add-ons are not full applications, rather they contain the visualizations and presentation of the data for a broader application. No other configurations should be included (extracts, tags, event types, macros, line breaking configurations, and so on). Dashboards and views are the primary objects contained within this type of add-on.
Chapter 1

• **Supporting add-ons (SA):** Supporting add-ons are also not full applications; these contain *data definitions*, such as macros, saved searches, event types, and tags. These describe how to correlate the data, normalize the data, and consolidate the data to be usable in the domain add-on.

• **Technology add-ons (TA):** Technology add-ons provide extraction, data massage, and index-time configurations. These can also be referred to as technical add-ons. These contain the configuration options required to properly break events, extract search fields, and create timestamps, among other functions. These are the building blocks for the SA and DA add-ons, as well as full-blown applications.

Follow the Splunk application design guidelines. Using a custom naming scheme may cause conflicts.

Thus end the *official* naming conventions as normally seen in a Splunk installation. We will now discuss some other naming conventions that have been found to help in the wild west of various Splunk installations. These two naming conventions are of the author's own design, which have helped in some of his deployments:

• **Input add-ons (IA):** Input add-ons are just that—configurations that assist in the collection of data, known as inputs. These add-ons are most likely found on a deployment server and are used to collect data from universal forwarders. One of the advantages to splitting your IAs from your TAs is a reduced size in the add-on being sent to the universal forwarder. This is especially useful if your TA contains lookups that aren't needed on the universal forwarder but are several megabytes in size.

• **Admin add-ons (ADMIN):** This add-on is a very special add-on. It would typically contain *administrative* configurations that might be needed in a variety of locations. Such configurations could be the web server SSL port, deployment client information, or anything in *web.conf* or *server.conf* format. It can be used to send index information to a set of non-clustered indexers, or possibly to scale the addition of more search heads by setting all relevant settings from a central location.
While this may not be a complete list of naming conventions, it should be enough to recognize any that are seen in the wild. An additional aspect of the naming conventions that we recommend is the addition of company information. This will help your Splunk admins differentiate between Splunk add-ons and custom add-ons. Just as an example, let's say you built a TA for Cisco, specific to your company (the ACME company). Splunk's provided add-on is entitled *TA-cisco*, but you don't want to modify a vendor's offering. So, your new add-on's name could be *A-ACME-TA-cisco*. This gives you two things: an easy-to-see custom TA that relates to Cisco and the ability to override any *TA-cisco* settings based on application precedence.

Let's discuss application precedence for a moment. Splunk uses a *merged configuration* when applying configurations that are installed via the applications. The methodology that Splunk chose to implement conflict resolution is pretty simple. There are two different methods of precedence. The first is directory structure. If you have an input located in the *default* folder of an application (more on *default* in the later chapters), you can place a matching configuration in the *local* folder of the application to override the *default* configuration. The same method is applied to the applications themselves. Splunk uses the ASCII values of the names to determine precedence. On *nix, you can sort the applications in the *apps* folder of Splunk using the *LC_COLLATE=C ls* command. This will show you the ASCII-sorted order of the applications, and the first in the list will be highest priority. A has a higher priority than Z, but Z has a higher priority than a. So, the A at the beginning of the add-on name gives your add-on the highest precedence, so you can override any setting as needed.

From this point forward, both Splunk applications and add-ons will be referred to formally as Apps purely as a convenience.

**Designing the App**

So you've decided that you need an App? Congratulations! Now that you know that you need one, you need to decide on a few more items as well. It is important to do a little bit of planning, as even the simplest Apps can evolve into super-complicated Apps, with dashboards, saved searches, workflows, and more. Never assume "well, this'll just be a quick development", as, most of the time, it is not.
Identifying the use case

First and foremost, try to determine the scope of your App. Once you have the scope planned out, try to limit the amount of scope creep that occurs, if possible. You may just be trying to perform extractions on your data, and if that is your current end goal, stop there. Don't try to build a full-blown suite on your first attempt. Build the IA, then the TA, and then move on from there. Ask yourself these questions as you try to determine your scope:

- What am I trying to accomplish? Search-time extractions? Index-time parsing? Dashboards to share?
- What users need access to my App? Everybody? Specific roles?
- What kind of information will I be presenting? Server based? Metric based?
- Who is my target audience? Business users who don't understand Splunk Search Processing Language (SPL), or technical users who will notice that I didn't convert MB to GB properly?

These questions can help you spark an idea of what internal resources would need to be engaged, as well as any kind of documentation and educational requirements.

Identifying what you want to consume

Once you have determined the scope of the App, you will need to decide how and from where you will consume the data. Getting data into Splunk can happen in a very wide variety of ways. There is no set manner of input that will work on all data sources. You may have to develop a new script or modular input. Being aware of where your data is coming from is the key to getting it consumed correctly the first time. A few questions you may ask yourself could be:

- Why do I need this data? Is it all completely relevant to my use case?
- Where is the data? Cloud, SaaS provider, internal network?
- How do I get the data? Do I already have a collector script, or do I need to engage an internal resource to write a collector/modular input?
- What format is the data? Is it already extracted (or well known, like syslog), or do I need to write custom extractions?

There is a lot of data out in the wild, but not all of it may be relevant to your use case. You may find that of a service that has 100 endpoints available for data collection, you only need 10. Not only will you save on license usage, but your indexers will thank you for it as well.
Identifying what you want to brand

Another key thought process in App development is how far you want to brand your App. Splunk has a very robust architecture and framework, providing you with the ability to customize your Apps extensively. You can override any individual piece of CSS and extend SplunkJS Stack to include any number of different visualizations or third-party libraries. Additional questions you might ponder on would include:

- Do I want to brand anything at all, or just stay with native Splunk?
- Do I need to engage an internal graphics resource to design and create App icons? App logos?
- Am I going for mobile or static desktops? What desktop size is typical of incoming users?
- To what extent should I customize my App? Do I just change a few colors using native Splunk options or do I override CSS?
- Do I need to engage a web designer to build custom CSS or HTML layouts?

There are so many options available to brand your App, but all customizations should conform to the Splunk branding guidelines for developers. Visit [http://www.splunk.com/view/SP-CAAAFT9](http://www.splunk.com/view/SP-CAAAFT9) to read through Splunk's guidelines.

Identifying what you want to display

Once you have the whats and hows of the data you're going to collect, you need to figure out visualizations. How you display the information is just as important as what data you collect. Splunk comes with a variety of graphs and displays right out of the box, and can be extended quite easily to include some really cool presentations. Some of the questions posed to you might be:

- Do you need a programmer to write custom modules or extend SplunkJS views and managers?
- What third-party graphing or graphic libraries do you need to document, develop, or get permission to use?
- Do you need to engage a statistician to determine the best and most effective way to display your data? Some stats (such as max, mean, and min) are easy, others (such as confidence intervals and trendlines) are not.
Such a small list of questions hardly precludes any other relevant discussion within your organization. The more internal discussion that can take place, the better and more thought-out your App may turn out.

## Installing Apps

As a Splunk developer, you should be aware of the three methods to install Apps. There are advantages and disadvantages to each method, but no required method. It is mostly personal preference as to which method is used by the end user, but, typically, newer Splunk users will use the Web interface, while advanced users will use the command line. Let's review those methods, just to keep them fresh in your mind.

### Splunk Web

Installing Apps via Splunk Web is simple. Once you have downloaded the App from its source, you navigate to the **Manage Apps** section of Splunk. You will find this at the top-left of Splunk Web, as shown in the following screenshot:

![Splunk Web Manage Apps](image)

Once you have clicked on **Manage Apps**, you will see a button to install the app from a file. You can also browse the Splunk App store, using the first button:

![Splunk Web App Store](image)
This brings you to a form that you can use to actually install the App. Simply click on the **Browse** button, select the file you downloaded, check the **Upgrade** button if this App has already been installed, and then click on **Upload**. That’s it! Splunk takes the App, installs it, and prompts to restart if needed:

### Upload an app

If you have a .spl or .tgz app file to install, you can upload it using this form.

You can replace an existing app via the Splunk CLI. Learn more.

File

[Choose File](#) No file chosen

- [Upgrade app. Checking this will overwrite the app if it already exists.](#)

[Cancel] [Upload]

---

### The Splunk command line

CLI holds a special place in many *nix admins' hearts. It is entirely possible to install Apps via the command line alone. Doing so requires having the following: access to the physical (or virtual) server and enough permissions to perform CLI commands with Splunk. All commands are going to be executed from `$SPLUNK_HOME`, which normally defaults to `/opt/splunk`. Follow these steps to install an App via CLI:

1. Copy the App file (either a *.tgz or *.spl file) to the filesystem.
2. Run the `./bin/splunk install app <path_to_file>` command.

Splunk will install the App. You may be prompted to restart, depending on the contents of the App. Index-time configurations require a restart, whereas search-time configurations do not.

### Unzipping using the command line

The final methodology is to perform an unzip/untar. If the App was constructed properly, the only steps you need to perform are as follows:

1. Copy the file to `$SPLUNK_HOME/etc/apps`.
2. Change the file extension from `.spl` to `.zip`.
3. Use your favorite utility and unzipped the file into the folder.
Caution! This will overwrite any other settings you have configured, including local configurations (if present in the zip file). We will cover directory structure in the next chapter.

**Summary**

In this chapter, we covered the basic fundamentals of designing and installing Splunk Apps. Apps can be broken down into *domains*, each with a naming convention that allows you to quickly determine what the App can do, and what is contained within it, so that new users to your environment don't have to look for configurations. We learned how to approach App design to make sure that the App is planned beforehand, which will eliminate the need to refactor major portions of the App later, when it may already be in production. We also went over the three different methodologies available to install Apps to give a basic understanding of user experience related to the installation of any App you may build.

Now that you've acquired an understanding of what an App consists of, in the coming chapters, we will discuss creating, enhancing, and customizing them.
Where to buy this book


[Click here](https://www.packtpub.com) for ordering and shipping details.