Intro - Splunk Web Framework Fundamentals

So you've installed Splunk, got things running, and now what? Hopefully, that is where this guide will come in and help you get the ball rolling, making fresh, interactive, useful, and dynamic applications using the Splunk Web Framework. We are hoping that we can actually get you creating some interesting applications without the usual log, index, search, graph, and report documentation that seems to be out in abundance.

Introducing the Splunk Web Framework

Welcome to the Splunk Web Framework, which has been set up as an essential support structure for Splunk users to build custom reports, dashboards, and apps on Splunk and with Splunk. This means that there is a supporting environment that can be used to develop end-to-end applications with no need to install anything other than Splunk. The Splunk Web Framework allows the user to start from the basics using a drag-and-drop interface, and makes them able to get underneath the hood and interact and customize the code directly. Further still, developers don't even need to develop with Splunk as their platform of choice to display their data. They are free to simply interface with Splunk API calls, search for data, and then display this returned data directly on their own websites and applications.

As of Splunk version 6, there was a major overhaul to the Splunk Web Framework. The framework is now integrated directly into Splunk Enterprise 6, so now you don't need to install anything else to start using the web framework. Previously, in Splunk 5, you needed to use a standalone version of the web framework. So unless you're using an old version of Splunk, you will be able to get going and working with the framework straight away. All your apps from previous versions of Splunk should work on Splunk 6, including apps created in Advanced XML, so it is well worth the upgrade to get an improved interface and functionality that it brings.

A quick note about advanced XML

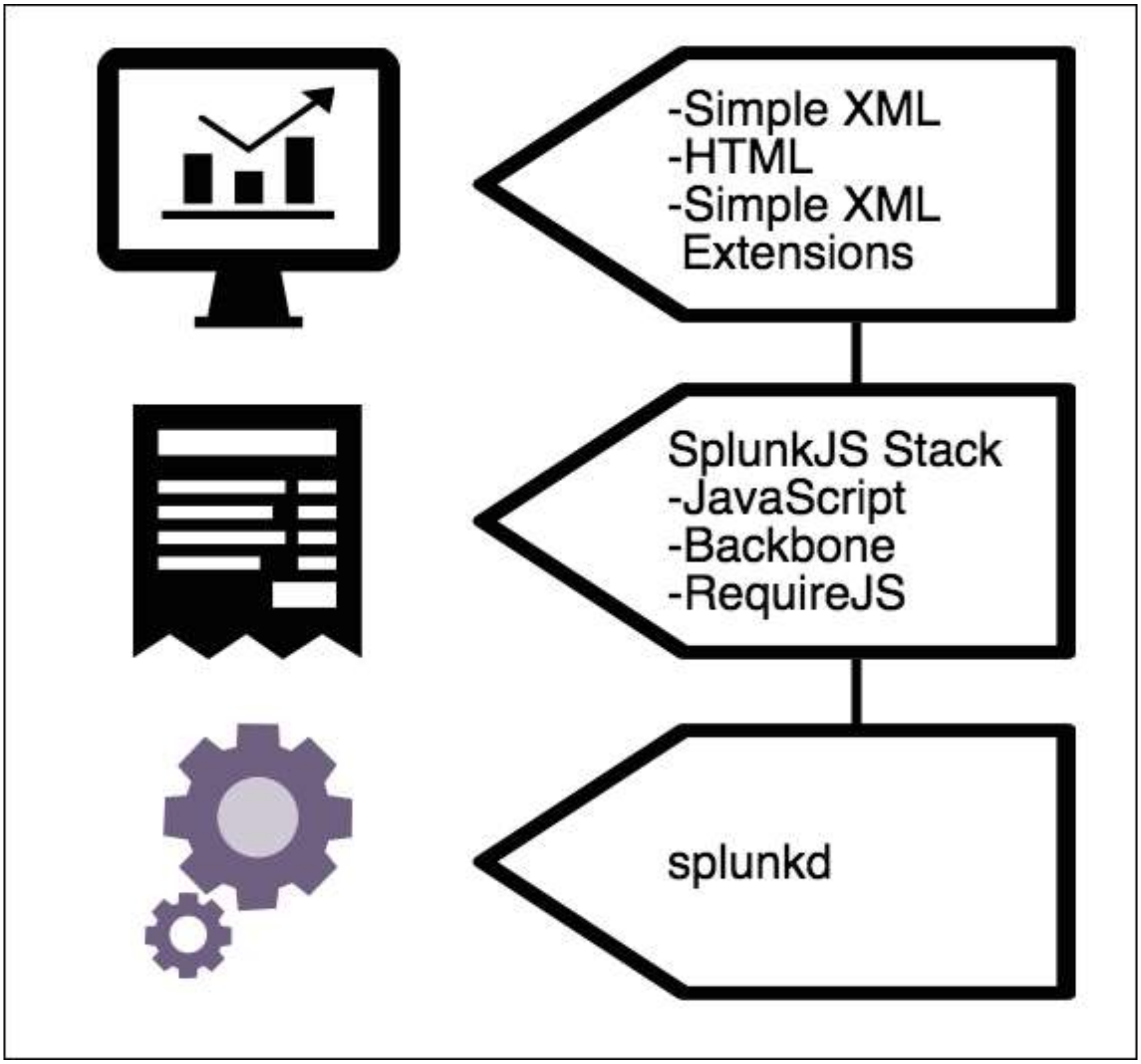
Let's get this out of the way early. You may have heard about Advanced XML, or you may have even seen some dashboards or views created in your environment that have been set up using Advanced XML. As of Splunk Enterprise 6.3, the Advanced XML feature has been deprecated. Although apps and dashboards using Advanced XML will continue to work and Splunk will continue to support and fix bugs, there will no longer be any feature enhancements to the Advanced XML feature of the Splunk Web Framework.

A date has not yet been set for the removal of Advanced XML from Splunk Enterprise. All future development should be done using other features of the Splunk Web Framework, and all existing apps or dashboards that use Advanced XML should be migrated away from Advanced XML and onto one of the other options available in the Splunk Web Framework.

Architecture of the Splunk Web Framework

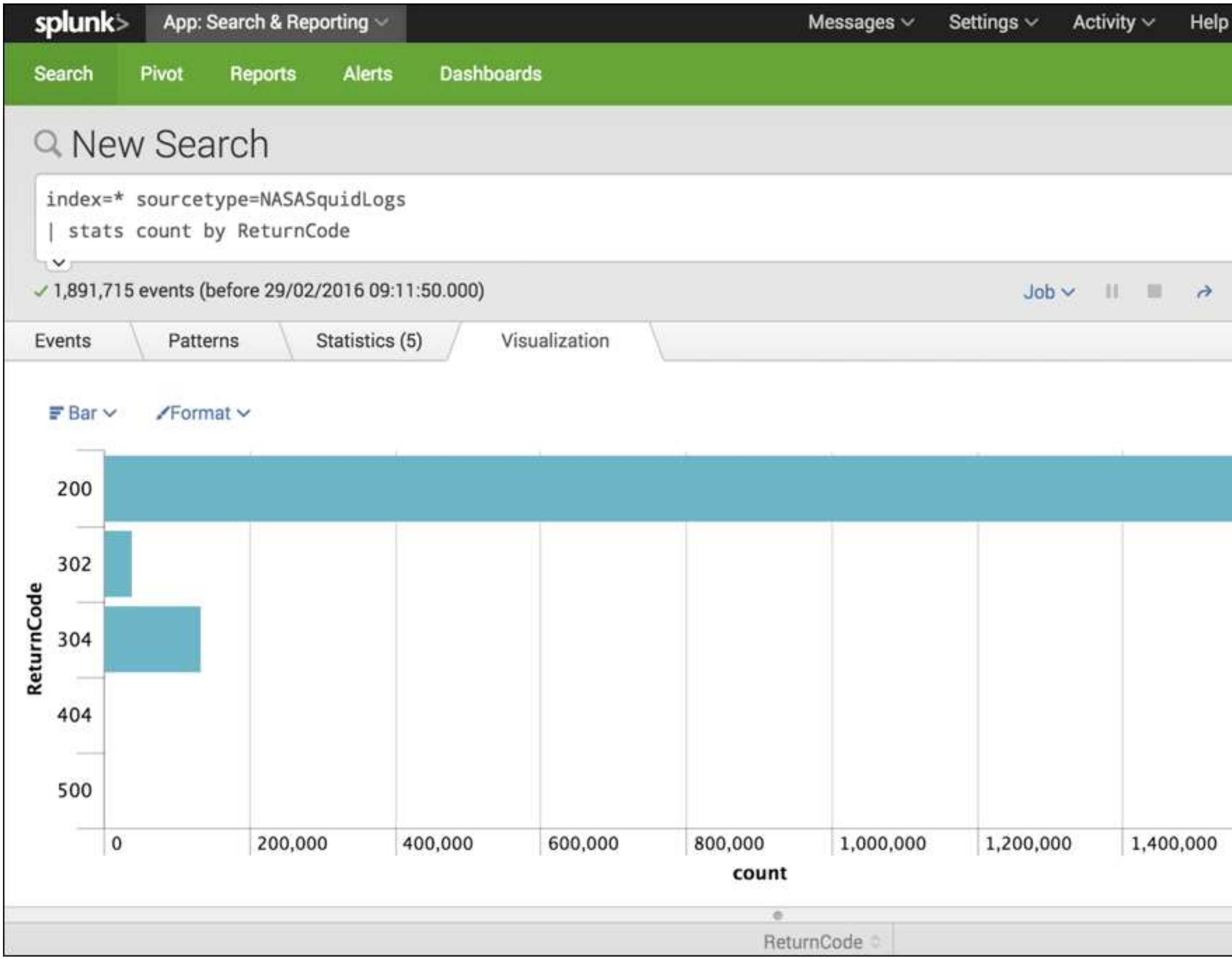
The Splunk Web Framework is now built directly on the core Splunk daemon, splunkd. Originally, splunkd only handled indexing, searching, and forwarding, but as of version 6.2, it also operates the Splunk Web Interface. Making this change was practical because it gave the framework the tools you need to build web applications directly on Splunk, as well as use the data that Splunk provides to display on your own website.

Within the framework, you have an app that will include numerous dashboard elements within the app. Within the dashboards, you will then have numerous panel and visualization elements that will make up your dashboard:



Description of the architecture

The preceding diagram provides a clear breakdown of the architecture, and its three distinct layers. It shows splunkd, which is built on C/C++ for speed and stability, as a server that provides the indexing and searching capabilities to the SplunkJS stack, which delivers the display and interface supporting the SimpleXML, HTML, and external web displays. Each layer builds on the others, providing further enhanced functionality.



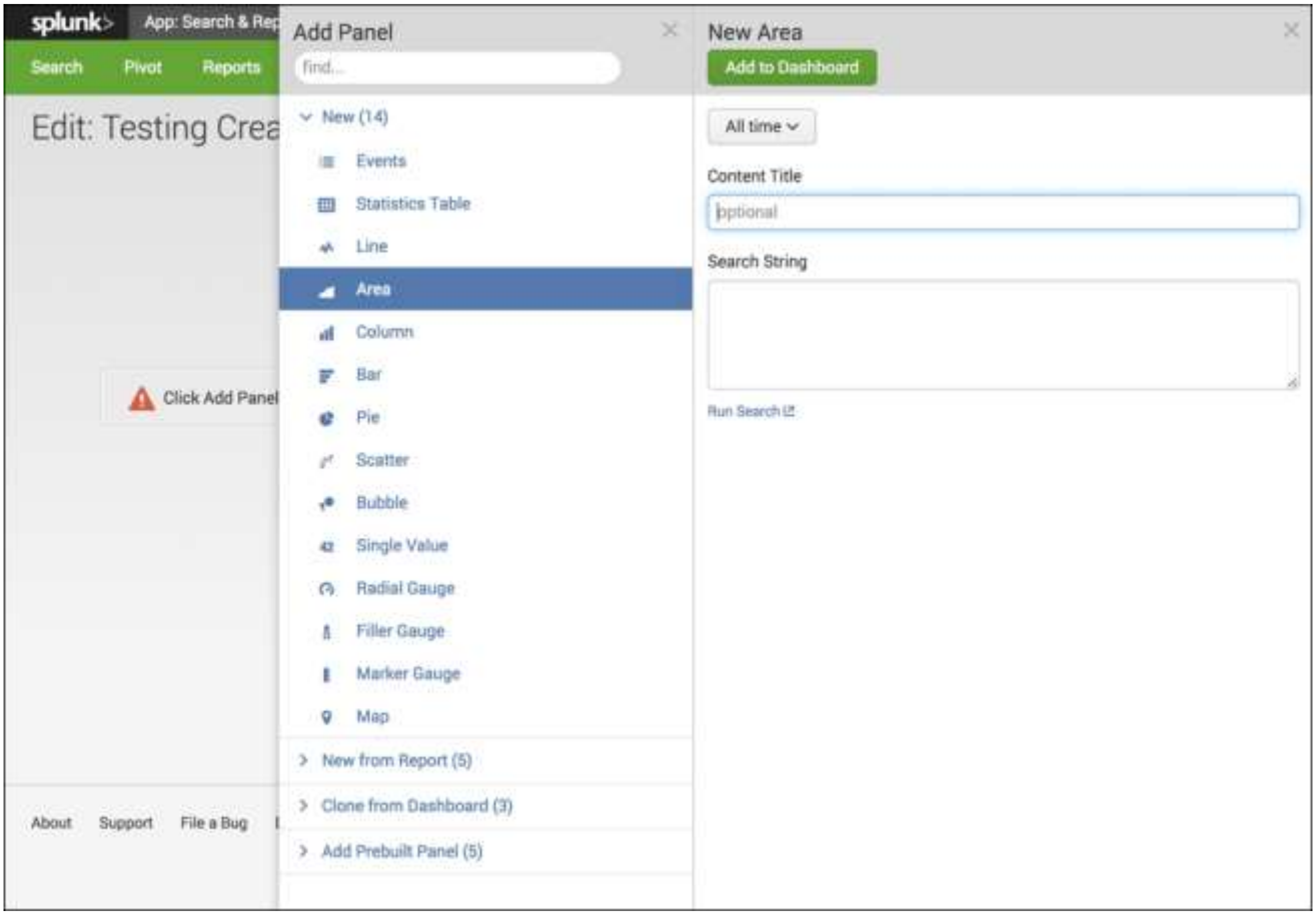
```
index=* sourcetype=NASASquidLogs
| stats count by ReturnCode
```

✓ 1,891,715 events (before 29/02/2016 09:11:50.000)

The Splunk web interface

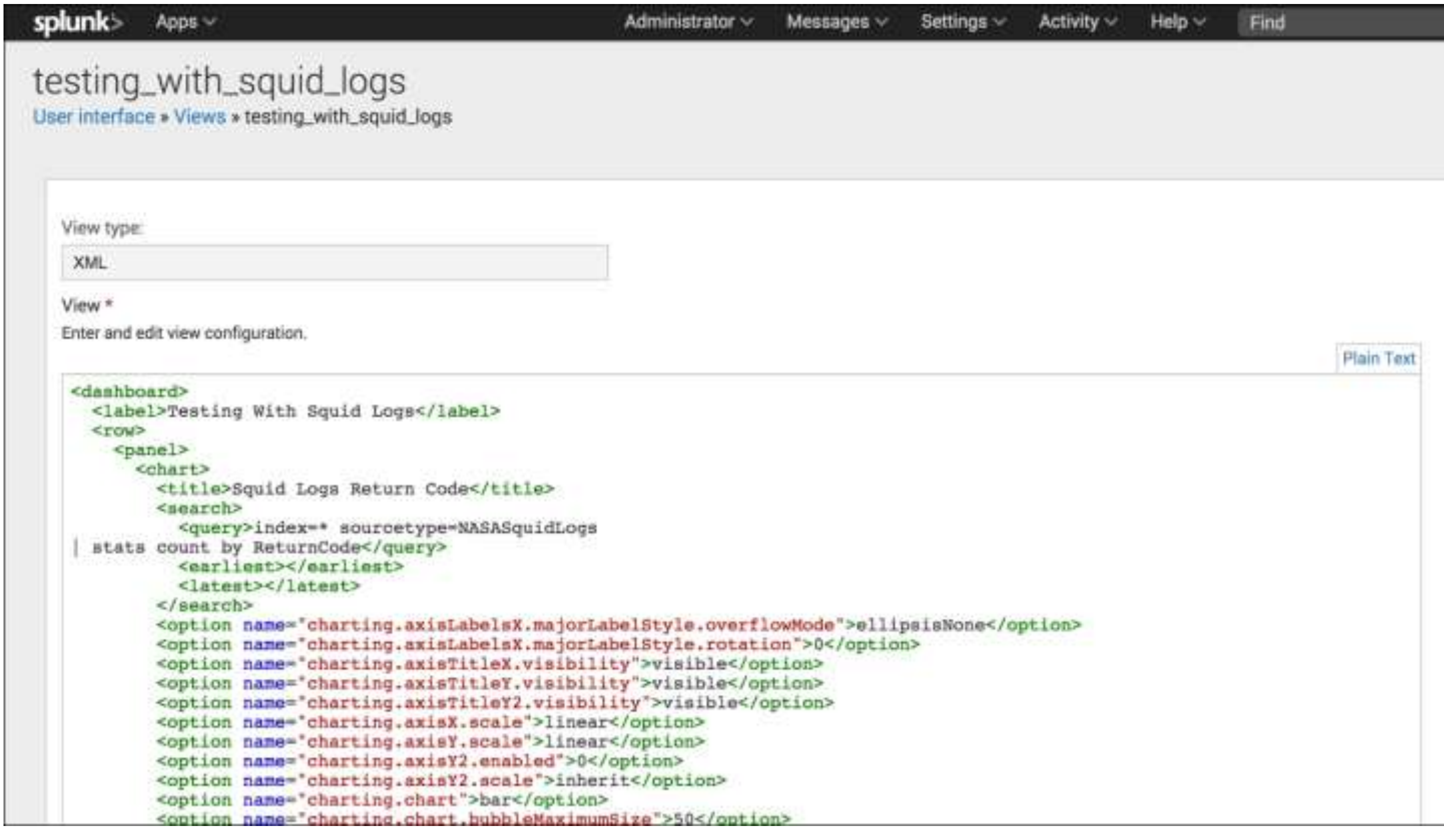
By now, I am sure you at least know that Splunk has a web interface. If you are competent with using Splunk, you would already be familiar with using the web interface for searching, configuring, and administration of Splunk. As part of the Splunk Web Framework, the web interface also provides an easy-to-use graphical user interface, which allows you to drag and drop tools and functionality with no prior programming knowledge or experience. It provides rapid development on the framework and allows you to visualize dashboard panels with ease.

The dashboard editor is the main interface and is part of the SimpleXML layer of the Splunk Web Framework; it allows you to build dashboards within Splunk Web. Here you can visualize your events and statistical information as dashboard panels and views and provide charting functionality. It even allows you to start providing form-based controls and an interface with the user.



Simple XML

Simple XML expands the functionality of the framework further and allows the user to fine-tune the dashboard panels with more layout and display options. Splunk's **Extensible Markup Language**(**XML**) is the underlying code that is developed when using the web interface and dashboard editor. Simple XML code can be edited and manipulated directly from Splunk's built-in editor, or you can use your own code editor to configure the easy-to-learn syntax. The directory structure within Splunk is also straightforward and easy to learn, and it helps you manipulate the environment in ways that you can't actually do within the web interface.



testing_with_squid_logs

User interface » Views » testing_with_squid_logs

View type:

XML

View *

Enter and edit view configuration.

```
<dashboard>
  <label>Testing With Squid Logs</label>
  <row>
    <panel>
      <chart>
        <title>Squid Logs Return Code</title>
        <search>
          <query>index=* sourcetype=NASASquidLogs
| stats count by ReturnCode</query>
          <earliest></earliest>
          <latest></latest>
        </search>
        <option name="charting.axisLabelsX.majorLabelStyle.overflowMode">ellipsisNone</option>
        <option name="charting.axisLabelsX.majorLabelStyle.rotation">0</option>
        <option name="charting.axisTitleX.visibility">visible</option>
        <option name="charting.axisTitleY.visibility">visible</option>
        <option name="charting.axisTitleY2.visibility">visible</option>
        <option name="charting.axisX.scale">linear</option>
        <option name="charting.axisY.scale">linear</option>
        <option name="charting.axisY2.enabled">0</option>
        <option name="charting.axisY2.scale">inherit</option>
        <option name="charting.chart">bar</option>
        <option name="charting.chart.bubbleMaximumSize">50</option>
```

From the preceding example, you can see that the syntax of the Simple XML code is straightforward and relatively easy to learn. The code provides a multitude of options to tweak and fine-tune all aspects of the display of the different types of panels provided. It is definitely worth learning to use this function of the Splunk Web Framework. Although the drag-and-drop interface allows you to develop rich and interesting dashboard panels, sooner or later you will start to want to configure the display in a way that you can only do in SimpleXML.

Each visualization type has a long list of properties that can be managed and changed through SimpleXML code. Although simple, you still need to adhere to the white space and open and close tags within the code. If not, you could end up with no display provided.

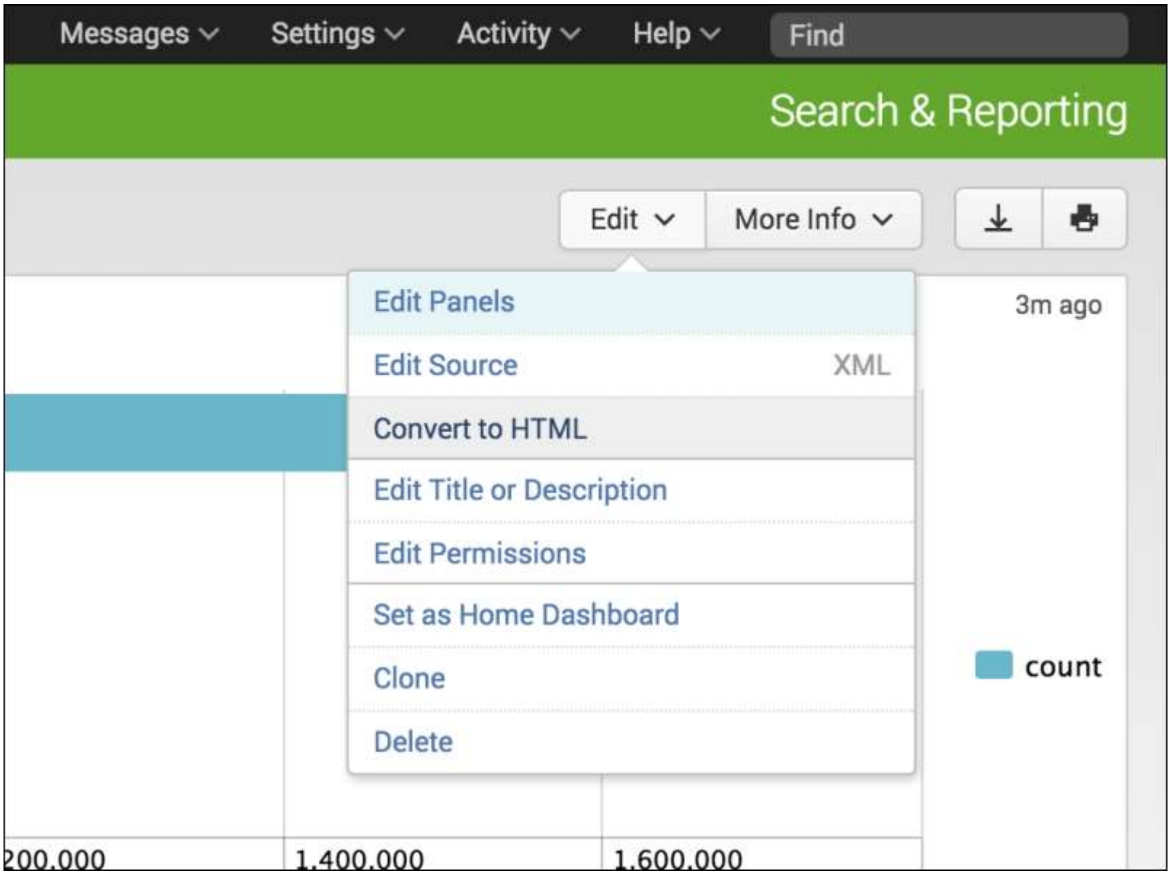
SimpleXML extensions

SimpleXML also allows you to create extensions to utilize CSS and JavaScript files so that you can further modify and enhance the behavior and appearance of a dashboard that was created through the code editor or web interface. You can modify the layouts further, add new visualizations, and customize the way that the end user interacts with the dashboards.

Working with SimpleXML extensions will get you working directly in the server directory structure of your Splunk deployment. Once you have added your CSS or JavaScript files to the server, it is simply a matter of editing your code to then use the files needed.

HTML

By now, you can see that each level of the framework expands functionality of the interface further, and utilizing HTML dashboards allows you to expand your functionality even further. Splunk comes with a converter that allows you to convert your Simple XML dashboards into HTML, and allows you to use the built-in code editor, edit, and configure the HTML dashboard further. As with Simple XML, you are also able to use your favorite code editor, allowing developers with knowledge of HTML, CSS, and JavaScript to transfer their knowledge and work directly in Splunk by using it as a platform to generate their HTML-based environment.



SplunkJS libraries

`SplunkJS` provides a framework of tools and libraries that allows developers to build and manage dashboards and organize dependencies, as well as integrate Splunk components into their own web applications. The libraries allow you to manage views and search managers to allow you to work with searches and interact with Splunk data. `SplunkJS` removes the developer from the Splunk Web Interface but gives the ability to both build Splunk Apps for Splunk and build web applications using Splunk data.

splunkd

This is the main system process that Splunk uses to handle all of the indexing, searching, forwarding, and web interface that you work with in Splunk Enterprise. Although we will need to restart Splunk and the splunkd process occasionally, this book will not be focusing on splunkd, as this would be more of a server administration focus.