Splunk® Enterprise
Developing Views and Apps for Splunk Web 6.3.1

Setting up a scripted input

Generated: 12/01/2021 12:41 pm

Setting up a scripted input

This section describes how to set up a scripted input for an app. To illustrate the setup, it uses an example script that polls a database and writes the results to a file. A more detailed version of this example is in Example script that polls a database. That topic provides details on the example, including code examples in Python and Java.

You can write any number and types of scripts in various scripting languages that perform various functions. This example shows the framework for a commonly found script. Adapt this framework according to your needs.

Script to poll a database

This example script does the following.

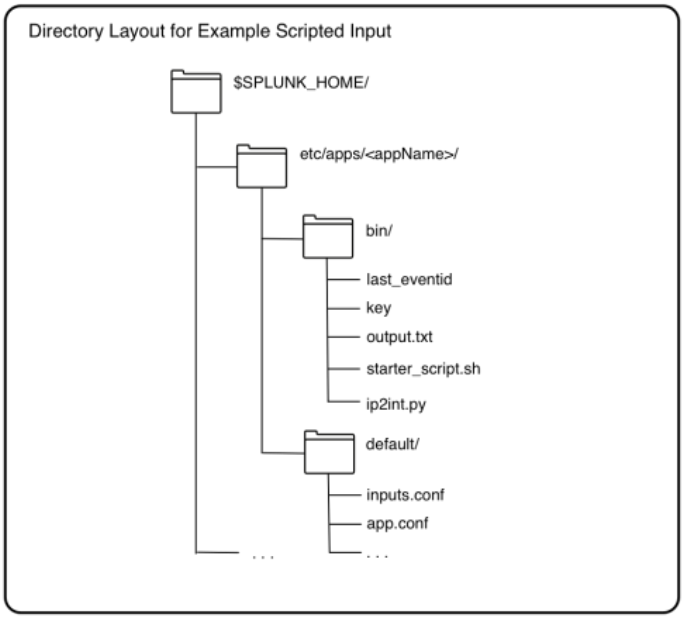
- Runs at a regular interval.
- Queries a database.
- Writes the output to a file in a format optimized for indexing.
- Splunk software indexes the file containing the results of the queries.

Directory structure

Place scripts in the `/bin` directory of your app.

```
$SPLUNK_HOME/etc/apps/<appName>/bin/
```

Here is the directory structure of the example script for this example. The directory structure for your app might differ.



Script files

. . ./etc/apps/<appName>/bin/my_db_poll.py

This is the script that retrieves information from the database. This script does the following:

- Queries the database and writes the query result to file.
- Defines the format of output data.
- Accesses a database using credentials stored in key.
- Reads last_eventid to determine the next event to read from the database.
- Queries the database at the next event and writes the output to a file.

. . ./etc/apps/<appName>/bin/starter_script.sh

Wrapper script that calls the `my_db_poll.py` script. In this example, it calls `my_db_poll.py` with the arguments needed to query the database.

In `. . ./etc/apps/<appName>/default/inputs.conf`, create a stanza that references this wrapper script. In this example, the stanza specifies how often to call the starter script to poll the database.

. . ./etc/apps/<appName>/bin/ip2int.py

A helper script to convert IP addresses from integer format to dotted format, and back. This is a type of helper script that formats data better for indexing. You often have helper scripts that aid the main script.

. . ./etc/apps/<appName>/bin/key

Text file containing username and password encoded in base64 using the python function `base64.b64encode()`. The Splunk Enterprise user has read and write access to this file.

Security for passwords is an issue when running scripts.

. . ./etc/apps/<appName>/bin/last_eventid

File containing a number for the last event received from the database. `my_db_poll.py` writes the last_eventid after querying the database. The Splunk user has read and write access to this file.

'. . ./etc/apps/<appName>/bin/output.txt'

A single event from the script, for reference. `my_db_poll.py` writes the actual output from querying the database to another directory.

. . ./etc/apps/<appName>/default/inputs.conf

Configure scripted data input in `$SPLUNK_HOME/etc/<appName>/default/inputs.conf`. Use the local directory for the app to overwrite behavior defined in the default directory. Here is an example:

```
[script://$SPLUNK_HOME/etc/apps/<appName>/bin/starter_script.sh]
disabled = true # change to false to start the input, requires restart
host = # enter hostname here
index = main
interval = 30          #frequency to run the script, in seconds
source = my_db
```

```
sourcetype = my_db_data
```

$SPLUNK_HOME/etc/system/local/props.conf

Configure properties for the script in the Splunk Enterprise system `props.conf`.

```
[my_db_data]
TIME_PREFIX=^[^\|]+\|
TIME_FORMAT=%Q
MAX_TIMESTAMP_LOOKAHEAD=10        #look ahead 10 characters
SHOULD_LINEMERGE=false
```

$SPLUNK_HOME/etc/system/local/transforms.conf

Define field transforms in `transforms.conf`.

```
[my_db_extractions]
DELIMS = "|"
FIELDS ="EventID","AlertTime","UserName",. . ."
```