Listening to the Data

Adding and Analyzing Music Data in Splunk

Sarah Moir, Program Manager

October 2018

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> .conf18

SARAH MOIR

Program Manager at Splunk

Former Radio DJ & Station Manager at WESN 88.1FM

splunk> .conf18

Adding and Analyzing Music Data in Splunk

You could do this with any kind of data

Where to start

Get data in

Search and visualize

What's next

Where to start

Getting data in…

What do I want to know?

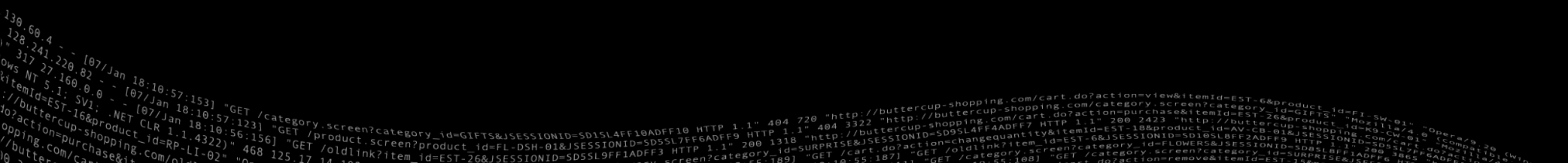
Use cases matter for data ingest

- Start by identifying use cases!
  - What's my listening pattern around a concert?
  - What artists did I discover in a particular year?
  - What songs did I listen to by a certain artist first, and how does that compare with my favorite songs by that artist?
  - Do I listen to more artists now? Do I listen to mostly different artists?
  - What is my favorite music venue in CA? In IL?
  - Am I more likely to skip a concert if I bought the tickets far in advance?

What data do I need to get these insights?

Identify data types based on use cases

- Listening pattern around a concert?
  - Listening data and concert data
- Artist discovery? Track discovery? Track popularity?
  - Listening data
- Favorite music venue?
  - Concert data
- Skipping concerts according to ticket purchase date?
  - Concert data, ticket purchase data



What data is out there?

Assess the data available to you

Music data I "created"

- Listening data tracked by Last.fm
- Concerts attended tracked by me
- iTunes library
- Spotify library
- SoundCloud library

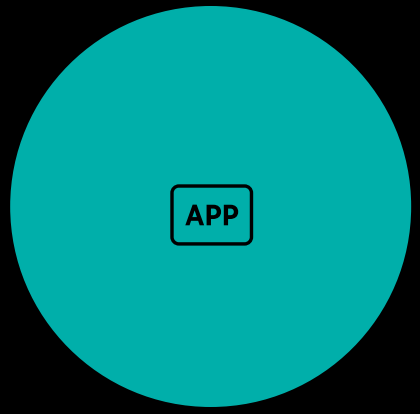
Music data externally available

- MusicBrainz metadata database
- Spotify audio features data
- Songkick concert data

How I got the data in

Figure out the easiest way to get data in

File Upload



Add-on Builder

Custom Alert Action

Lookup Editor App

Modular Input

File Upload: iTunes Library

XML party

- Uploaded the XML of the iTunes library file into Splunk.
- The XML of the iTunes library is not the XML format expected by Splunk
  - KV_mode=xml did not work for me, because the XML structure was not <field>VALUE</field> but rather <key>FIELD</key><key type>VALUE</key type>.
- Cue struggle.
- Cue asking people who know regex to write me regex.
- Cue celebratory gratitude.

splunk> .conf18

Add-on Builder: Last.fm Data

Hooray for unauthenticated APIs

- Add-on builder made it super easy to write a modular input.
- JSON-formatted data returned by the API
- Put together a little setup screen:
  - Request the username from the user
  - Ask for the API key (so that mine isn't hardcoded into it)
- Super easy UI for extracting fields from JSON events
- Checkpointing so that I can backfill data from when Splunk isn’t running
- Backfilled my data with file uploads so I didn’t have to wait for the input to churn through it all.

Custom Alert Action: Songkick Concert Data

Python 🐍

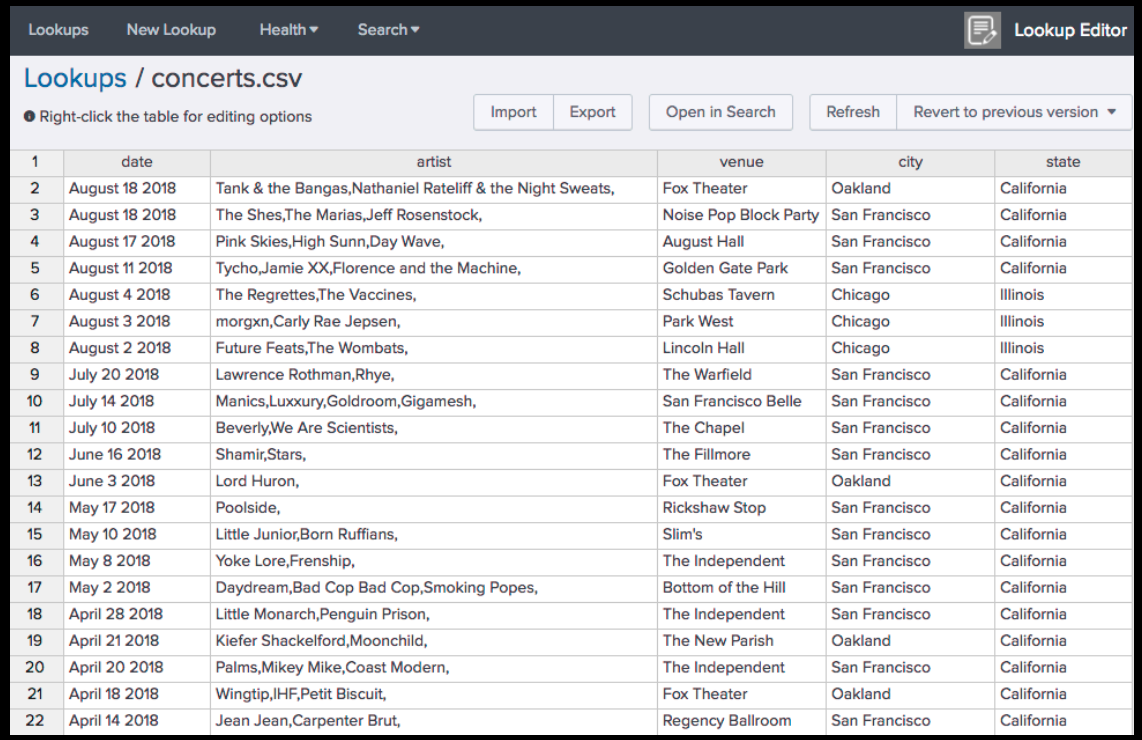
- Wanted to identify upcoming concerts for artists according to specific factors
  - Historical factors: which artists are above my average listen threshold and coming to town soon?
  - Trending factors: which artists have I recently discovered that are coming to town soon?
- Wanted to learn how to write an alert action

powered by songkick

Lookup Editor: Concert and Ticket Data

Manually-managed data

- Needed a way to keep track of external, time-ordered data manually
- Maintain 3 lookups
  - 2 concert lookups: 1 multi-value with artist field, 1 with added context of opener and headliner.
  - 1 ticket purchase lookup with date purchased, on sale date, face value, actual cost, and more!



Lookups | New Lookup | Health | Search | Lookup Editor

Lookups / concerts.csv

Right-click the table for editing options | Import | Export | Open in Search | Refresh | Revert to previous version

1	date	artist	venue	city	state
2	August 18 2018	Tank & the Bangas,Nathaniel Rateliff & the Night Sweats,	Fox Theater	Oakland	California
3	August 18 2018	The Shes,The Marias,Jeff Rosenstock,	Noise Pop Block Party	San Francisco	California
4	August 17 2018	Pink Skies,High Sunn,Day Wave,	August Hall	San Francisco	California
5	August 11 2018	Tycho,Jamie XX,Florence and the Machine,	Golden Gate Park	San Francisco	California
6	August 4 2018	The Regrettes,The Vaccines,	Schubas Tavern	Chicago	Illinois
7	August 3 2018	morgxn,Carly Rae Jepsen,	Park West	Chicago	Illinois
8	August 2 2018	Future Feats,The Wombats,	Lincoln Hall	Chicago	Illinois
9	July 20 2018	Lawrence Rothman,Rhye,	The Warfield	San Francisco	California
10	July 14 2018	Manics,Luxxury,Goldroom,Gigamesh,	San Francisco Belle	San Francisco	California
11	July 10 2018	Beverly,We Are Scientists,	The Chapel	San Francisco	California
12	June 16 2018	Shamir,Stars,	The Fillmore	San Francisco	California
13	June 3 2018	Lord Huron,	Fox Theater	Oakland	California
14	May 17 2018	Poolside,	Rickshaw Stop	San Francisco	California
15	May 10 2018	Little Junior,Born Ruffians,	Slim's	San Francisco	California
16	May 8 2018	Yoke Lore,Frenship,	The Independent	San Francisco	California
17	May 2 2018	Daydream,Bad Cop Bad Cop,Smoking Popes,	Bottom of the Hill	San Francisco	California
18	April 28 2018	Little Monarch,Penguin Prison,	The Independent	San Francisco	California
19	April 21 2018	Kiefer Shackelford,Moonchild,	The New Parish	Oakland	California
20	April 20 2018	Palms,Mikey Mike,Coast Modern,	The Independent	San Francisco	California
21	April 18 2018	Wingtip,IHF,Petit Biscuit,	Fox Theater	Oakland	California
22	April 14 2018	Jean Jean,Carpenter Brut,	Regency Ballroom	San Francisco	California

Modular Input: Spotify Library Data

Modular all the things!

- Wrote a python script to pull Spotify library data
- Migration to a modular input is IN PROGRESS 📶
  - Spotify API uses OAUTH and an authorization flow
  - Still not sure how that authorization flow will look in Splunk.
  - Explored using HEC to push the data in, but there is no mechanism for push in the API.
  - Need more time to sit with this and get it going somewhere. 🤷‍♀️

Now what?

Analyze and visualize!

Search and Visualize

Iterate constantly to get the most precise answer

- Start with a list of questions
- Identify the data necessary to answer them
- Write searches to get the answers you want
  - Write better and faster searches the better you get at SPL
  - Identify spots where you can do calculations elsewhere
- Create visualizations to show off the answers you find
- Create better visualizations the more you understand inputs and annotations and custom visualizations

Artist discovery vs popularity

Hipster status: intact

Earliest Track Listens By Artist : "The Vaccines"

artist	track_name	earliestlisten
The Vaccines	Nørgaard	06/12/2011 04:38:06
The Vaccines	Post Break-Up Sex	04/04/2012 21:22:00
The Vaccines	Blow Your Mind	06/27/2013 14:47:39
The Vaccines	No Hope - Live in Brighton	06/27/2013 14:49:48
The Vaccines	No Hope	10/29/2014 17:11:18
The Vaccines	If You Wanna	09/04/2015 12:30:30
The Vaccines	Norgaard	09/06/2015 18:56:20
The Vaccines	I Always Knew	04/01/2016 16:18:30
The Vaccines	I Can't Quit	01/05/2018 13:00:25
The Vaccines	Nightclub	01/27/2018 17:46:59
The	Put It On a T-Shirt	03/12/2018

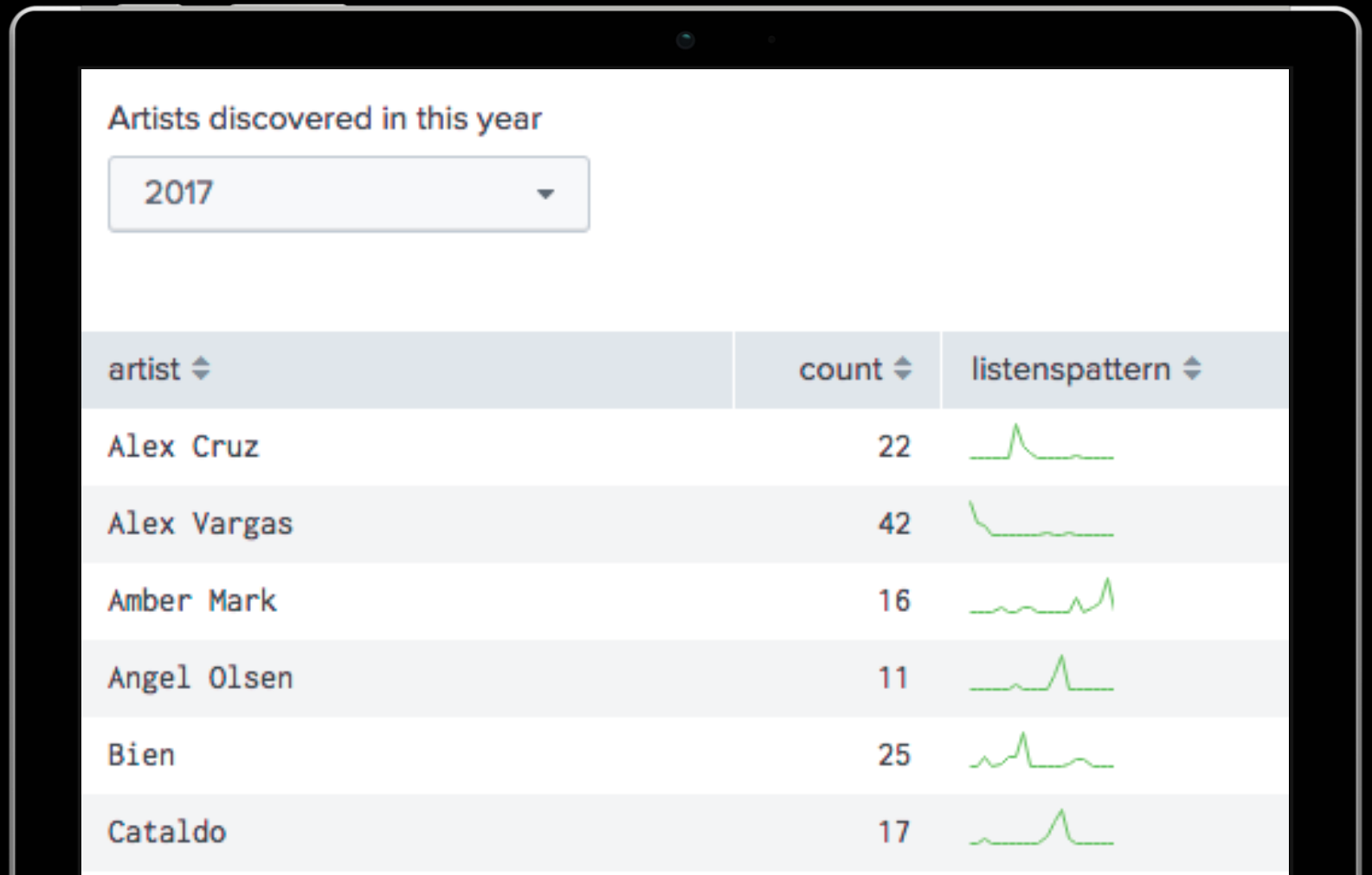
Top Listens by Artist: "The Vaccines"

track_name	album	artist	listens
Norgaard	What Did You Expect from The Vaccines?	The Vaccines	53
If You Wanna	What Did You Expect from The Vaccines?	The Vaccines	17
Wreckin' Bar (Ra Ra Ra)	What Did You Expect from The Vaccines?	The Vaccines	15
Wetsuit	What Did You Expect from The Vaccines?	The Vaccines	13
Under Your Thumb	What Did You Expect from The Vaccines?	The Vaccines	7
No Hope	No Hope	The Vaccines	6
Blow It Up	What Did You Expect from The Vaccines?	The Vaccines	4
Post Break-Up Sex	All Things Go West Mixtape	The Vaccines	4

- One dashboard input used by two panels
- One panel pulling earliest listen data
- One panel pulling top listens
- Both panels using Last.fm data

Artist discovery for a particular year

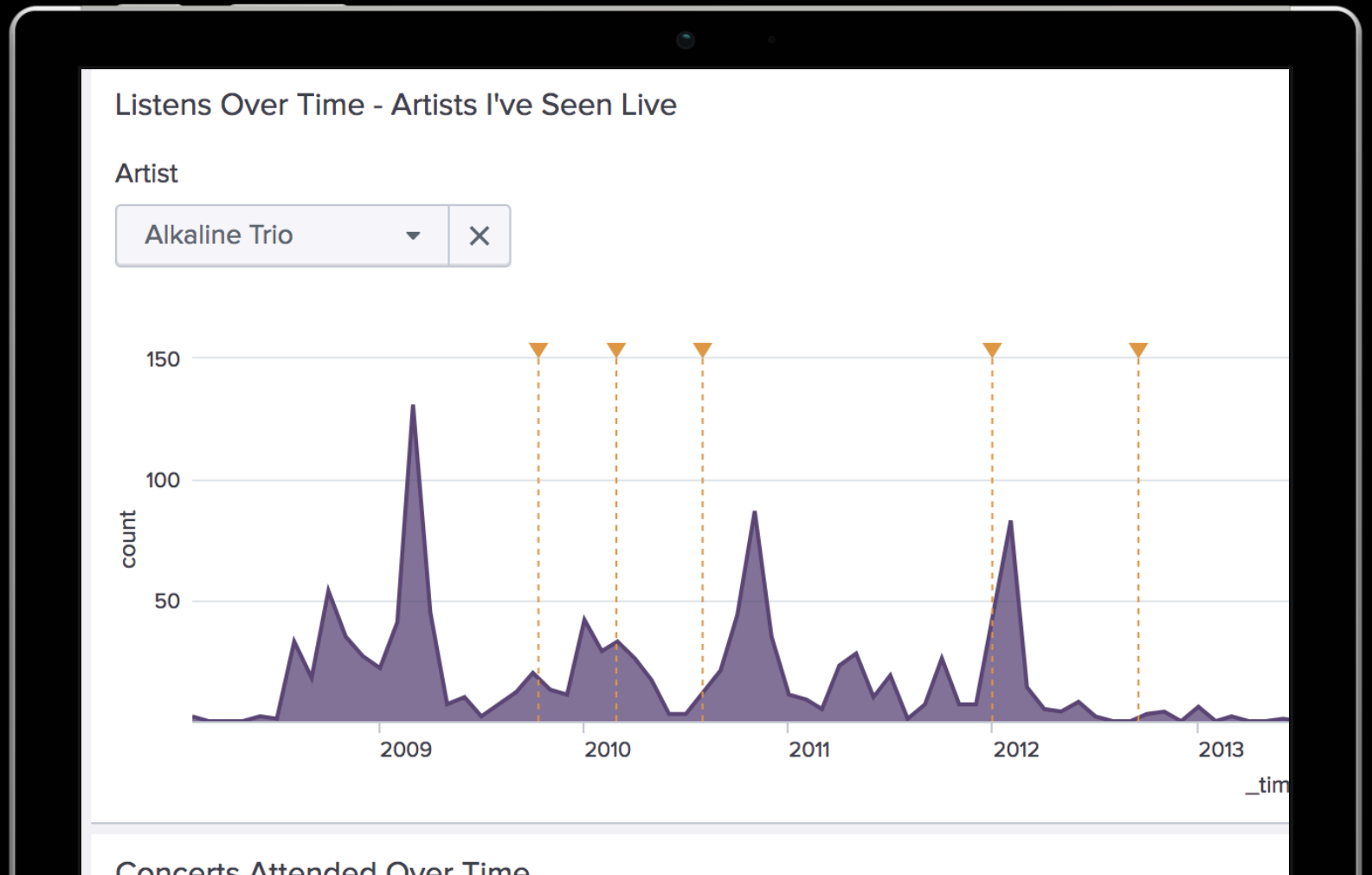
(usually lots)



- Use a search-driven lookup to store the “earliestlisten” of an artist
- Combine that with a dashboard input to search for a particular year
- Use sparklines to track listens throughout the year

What's my listening pattern around a concert?

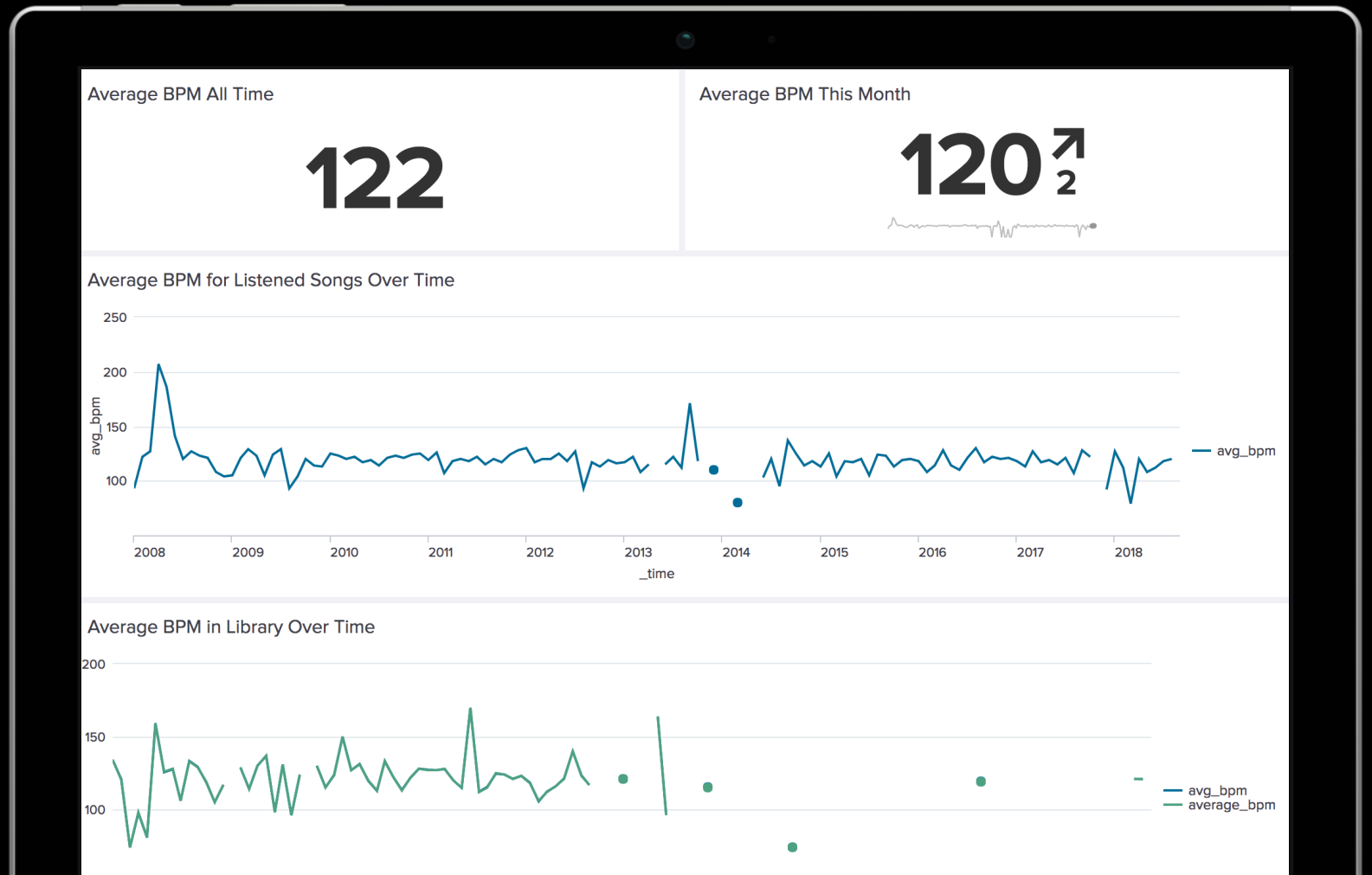
Sporadic correlations



- Used event annotations to compare concert dates with listening patterns
- Sporadic, inconsistent correlations
- See the appendix for the XML

BPM Listen Analysis

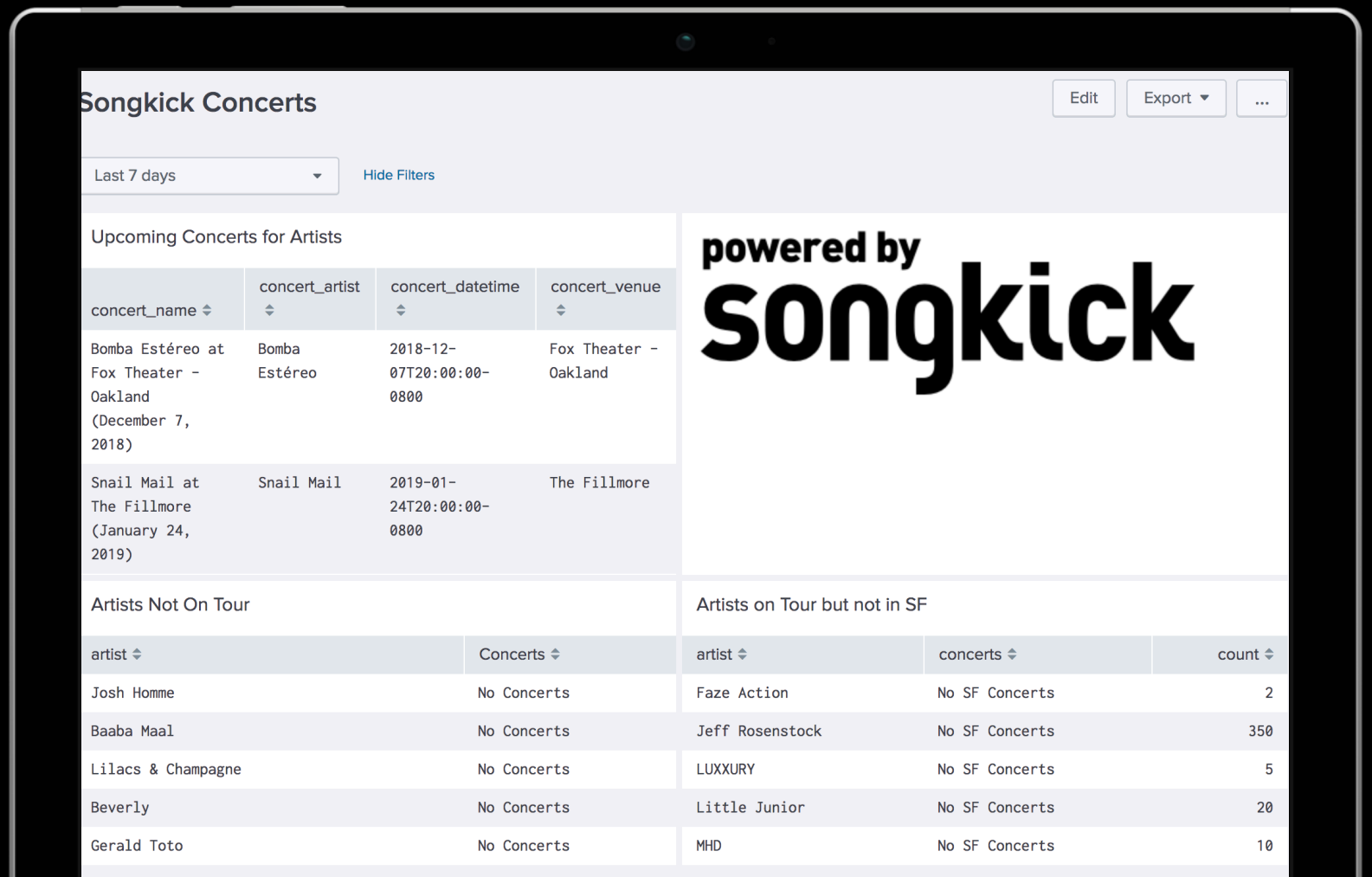
Oontz oontz oontz



- Analyze BPM trends by combining iTunes metadata with listening data
- BPM is in iTunes data
- Store that in a search-driven lookup for quick enrichment

Who Is On Tour?

Powered by Songkick



- Combines events created by the alert action with log messages generated by various types of failed responses from the Songkick API

What's in the future?

What do I still want to do?

Future use cases

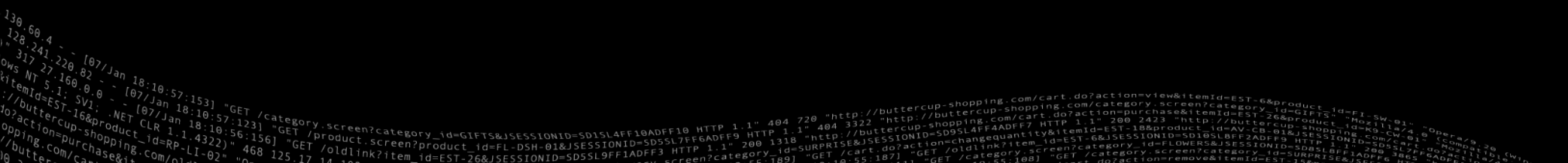
What I want to know that I need more data to know

- As I explored my data, I found new use cases that I wanted to answer with other data types:
  - What “types” of music do I listen to? How has that changed over time?
  - Calculate a popularity score for what I listen to and compare it to Spotify’s popularity score.
  - Personal one hit wonders?
  - Do I prefer music created in specific geographic regions?
  - Track music discovery across services, from first listen until iTunes library
  - What audio features are common across my most-listened-to songs?
  - Music taste prediction algorithm based on audio features and other features??

Future data types and analysis

Data growth

- MusicBrainz metadata about artists and tracks
  - Release dates of tracks and albums
  - Geographic location of artists
- Spotify audio feature data about tracks
  - Collect and analyze audio features
  - Can enhance existing library knowledge
  - Enrich beyond Spotify data



Key Takeaways

20 minute talks are short

1. Identify your use cases before you get data in
2. Identify the data you need to satisfy the use cases
3. Plan GDI according to data format and needs
4. Iterate searches and visualizations for precision
5. Record use cases that come up during data analysis

Thank you!

Don't forget to rate this session in the .conf18 mobile app

Appendix

All the configurations fit to include

Learn More

About my specific project

- See my Splunk blog post: https://www.splunk.com/blog/2018/01/04/10-years-of-listens-analyzing-my-music-data-with-splunk.html

splunk> .conf18

Learn More About Specific Configs

How did she do that?

- In the following slides, you can see the configurations I used
- Some configurations support getting data in
- Other configurations support visualizing the data
- In many cases I'm showing the raw configs or the raw XML, but you can often do these configs in the UI. It's simpler for me to show you the end product than the step-by-step UI instructions.

iTunes props.conf

XML library props.conf

```
45  [itunes_xml]
46  KV_MODE = none
47  LINE_BREAKER = (<dict>)
48  BREAK_ONLY_BEFORE = (<dict>)
49  DATETIME_CONFIG =
50  NO_BINARY_CHECK = true
51  SHOULD_LINEMERGE = false
52  TIME_PREFIX = <key>Date Added</key>
53  disabled = false
54  pulldown_type = true
55  TRUNCATE = 500000
56  category = Custom
57  REPORT-itunes_xml = itunes_xml
58
```

- This assumes that you have copy-pasted the tracks out of your library.xml file and into a dedicated file.
- The full library.xml file includes other details like podcasts and playlists that I didn't want to deal with yet.
- This is also missing a sed replacement for &, as those are not transformed from & to &

iTunes transforms.conf

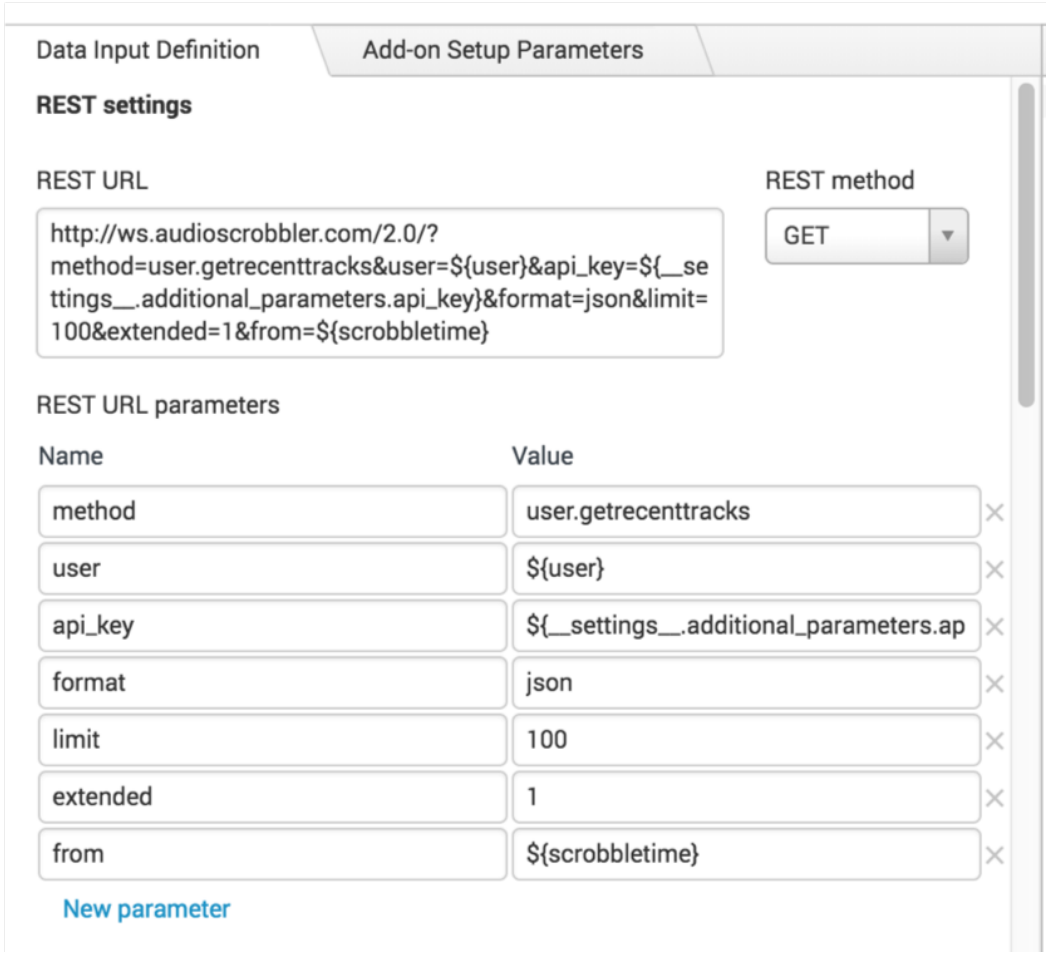
XML library field transformations

```
8  [itunes_xml]
9  REGEX = <key>([^<]+)</key><[^>]+>([^<]+)</
10 FORMAT = $1::$2
11 CLEAN_KEYS = true
12
13 [itunesxmlbool]
14 REGEX = <key>([^<]+)<\/key><(.+?)\/>
15 FORMAT = $1::$2
16 CLEAN_KEYS = true
```

- The first extraction is for most tracks
- The second extraction catches some with a slightly different format

Add-on builder data input settings

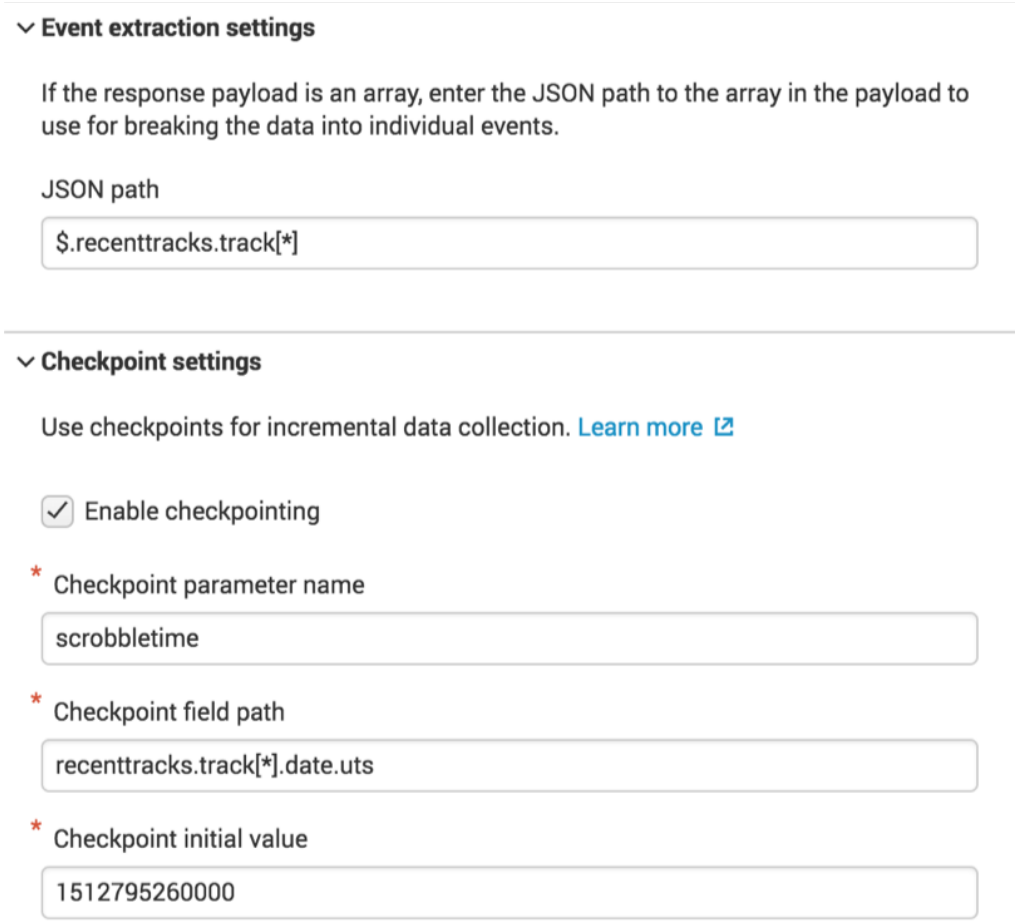
For the Last.fm API



- Make sure the REST URL references all the parameters you set up
- The value of the REST URL parameters can be from the script you write, the setup pages, or the checkpoint variable you set.

Add-on Builder event breaking and checkpoint settings

For the Last.fm API



- Setting a JSON path for the event extraction is very convenient and I love it.
- When you add a checkpoint parameter name, remember to add it to the REST URL parameters!!
- Set a checkpoint initial value especially if you have done backfilling of your data.
- The add-on builder highlights the JSON response to help you determine if your field path is accurate.

splunk> .conf18

Songkick alert action script

The method that calls the API and queues events

```
    ## This method will do the actual work of calling the API itself
    def dowork(self, result):
        ## get api_key from TA setup
        api_key = self.configuration.get("api_key")
        ## get artist_mbid (default is none)
        artist_mbid = result.get("artist_mbid")
        ##
        if artist_mbid:
            url =
"https://api.songkick.com/api/3.0/artists/mbid:{0}/calendar.json?apikey={1}".format(artist_mbid,api_key)
            self.message('Calling url for artist {0}'.format(artist_mbid))
            ## build sourcetype
            sourcetype = 'songkick'
            ## make request
            r = requests.get(url)
            ## process successful request
            if r.status_code==200:
                self.message('Successfully asked Songkick for concerts', status='success')
                ## songkick returns a whole boatload of concerts
                ## all part of one json for each artist
                loadConcert=json.loads(r.text)
                ## Filter to return only results with actual upcoming concerts
                if loadConcert['resultsPage']['totalEntries'] > 0:
                    ## record the fact that there are actual concerts
                    ## self.message('There are concerts for artist {0}'.format(artist_mbid))
                    ## Determine if those concerts are in the SF Bay Area and if yes, add the concert
                    for locale in loadConcert['resultsPage']['results']['event']:
                        if locale['venue']['metroArea']['id']==26330:
                            concert=json.dumps(locale)
                            self.addevent(concert,sourcetype=sourcetype)
                            self.message('Successfully queued concert events for writing to index',
status='success')
                        ## If there are no concerts in the SF Bay Area, log that and where they are instead
                        else:
                            place=locale['venue']['metroArea']['displayName']
                            self.message('No concerts in SF Bay Area for artist {0}, instead in
{1}'.format(artist_mbid,place))
                ## If there aren't any events for the artist, log that fact.
                else:
                    self.message('No concerts for artist {0}'.format(artist_mbid))
            ## process unsuccessful requests
            else:
                self.message('Failed to ask Songkick for concerts',
                    status='failure',
                    status_code=r.status_code)
```

- Working to provide the full script online as well as a blog post.
- This is the functional part of the script I wrote for the Songkick alert action.
- Missing parts: imports, writing the events to Splunk, most error handling

Recommended lookup headers

Pick a consistent date format for easy transformation

- For ticket data

purchase_date, onsale_date, concert_date, artist, site, promoter, quantity, cost, face_value, discovery

- For concert data

Date, artist, venue, city, state

- But also

Date, opener1, opener2, opener3, headliner, venue, city, state, info, festival_name, band5, band6, band7

- Pick a consistent date format for easy transformation
- Choose which extraneous metadata about concerts you want to include
- Use the full state name instead of the two letter abbreviation to make KMZ lookups work
- I use “info” to designate if it’s a festival or a DJ set or something else as needed.

Spotify progress

Spoilers: not much

```
14  # Prompt the user for authorization in order to get an authorization token
15  token = util.prompt_for_user_token(username,scope,client_id,client_secret,redirect_uri)
16  # The redirect_uri must be whitelisted at https://beta.developer.spotify.com/dashboard/applications/
17
18  # Docs about the util above:
19  # Call util.prompt_for_user_token method with the username and the desired scope (see Using Scopes for
    .   information about scopes) and credentials. This will coordinate the user authorization via your web browser and
    .   ask for the SPOTIPY_REDIRECT_URI you were redirected to with the authorization token appended. The credentials
    .   are cached locally and are used to automatically re-authorized expired tokens.
20
21  ## This is where the magic happens as far as getting saved tracks with _time as date_added
22  ## Validate that there is a token provided
23  if token:
24      sp = spotipy.Spotify(auth=token)
25      ## Get results from the Spotify API for the saved tracks of the current user
26      ## Query parameters for the API endpoint go in the parentheses
27      results = sp.current_user_saved_tracks(limit=5)
28      for x in results['items']:
29          track=json.dumps(x)
30          print(track)
31  else:
32      print "Can't get token for", username
33
```

- Right now the script hardcodes the scope, username, client_id, client_secret, and redirect_uri
- The Spotipy library makes everything easier to use
- As you can tell, lots of work to do to convert this to a modular input still.

splunk> .conf18

Earliest listen search-driven lookup

- Test the search in the search pipeline first before scheduling it
- Do inputlookup append=t before you do the outputlookup

splunk> .conf18

Discovery vs Popularity XML

```
<form>
  <label>Music Inputs</label>
  <fieldset submitButton="false">
    <input type="text" token="artist">
      <label>artist</label>
      <prefix>"</prefix>
      <suffix>"</suffix>
      <initialValue>*</initialValue>
      <default>Geographer</default>
    </input>
  </fieldset>
  <row>
    <panel>
      <title>Earliest Track Listens By Artist : $artist$</title>
      <table>
        <search>
          <query>`lastfm` | search artist=$artist$ |stats earliest(_time) AS earliestlisten by artist track_name | fieldformat earliestlisten = strftime
            (earliestlisten, " %m/%d/%Y %H:%M:%S") |sort earliestlisten</query>
          <earliest>$earliest$</earliest>
          <latest>$latest$</latest>
          <sampleRatio>1</sampleRatio>
        </search>
```

```
    <panel>
      <title>Top Listens by Artist: $artist$</title>
      <table>
        <search>
          <query>`lastfm` | search artist=$artist$ | stats count(track_name) as listens by track_name album artist | sort -listens</query>
          <earliest>0</earliest>
          <latest></latest>
          <sampleRatio>1</sampleRatio>
        </search>
```

- Dashboard input and two searches for each panel. The panels each use the same token as the dashboard input.

splunk> .conf18

What's my listening pattern around a concert?

XML

```
<panel>
  <title>Listens Over Time - Artists I've Seen Live</title>
  <input type="dropdown" token="artist" searchWhenChanged="true">
    <label>Artist</label>
    <initialValue>Alkaline Trio</initialValue>
    <fieldForLabel>artist</fieldForLabel>
    <fieldForValue>artist</fieldForValue>
    <search>
      <query>|inputlookup concerts.csv | makemv artist delim="," | mvexpand artist | dedup artist | sort artist</query
        >
      <earliest>0</earliest>
      <latest></latest>
    </search>
  </input>
  <chart>
    <search type="annotation">
      <query>|inputlookup concerts.csv | makemv artist delim="," | mvexpand artist | search artist="$artist$" | eval
        _time=strptime(date, "%B %d %Y") | rename annotation_label as Concert</query>
      <earliest>0</earliest>
      <latest></latest>
    </search>
    <search>
      <query>`lastfm` | search artist="$artist$" | timechart count</query>
      <earliest>0</earliest>
      <latest></latest>
    </search>
    <option name="charting.chart">area</option>
    <option name="charting.drilldown">none</option>
    <option name="refresh.display">progressbar</option>
  </chart>
</panel>
```

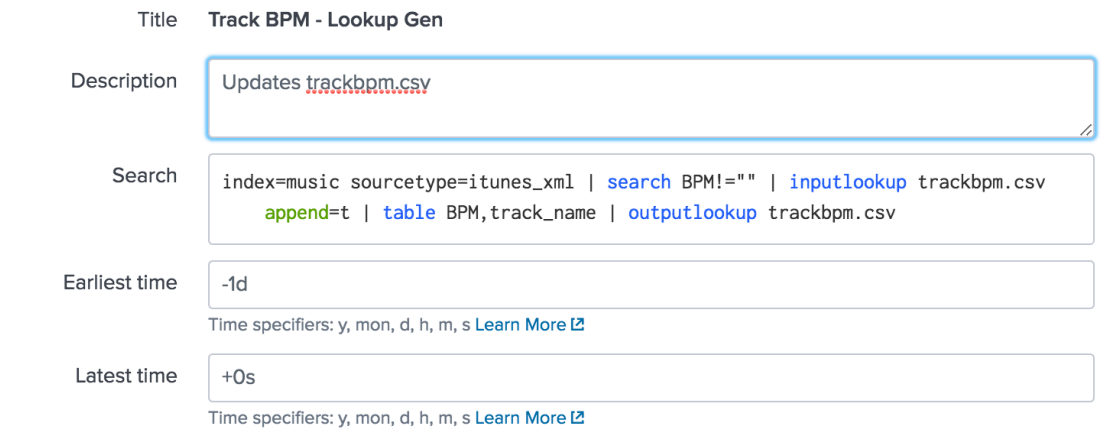
- The dashboard input is populated by a search into the lookup.
- The event annotation search has a type="annotation"
- The other search is real basic.

splunk> .conf18

BPM Listen Analysis

XML for the trending BPM over time indicator

```
<panel>
  <title>Average BPM This Month</title>
  <single>
    <search>
      <query>`lastfm` | table track_name,artist,_time | lookup trackbpm.csv track_name OUTPUT BPM | search BPM!="" | table BPM, track_name, _time | where BPM
        &lt; 200 | timechart avg(BPM) as avg_bpm  | eval avg_bpm=round(avg_bpm)</query>
      <earliest>0</earliest>
      <latest></latest>
    </search>
    <option name="drilldown">all</option>
    <option name="refresh.display">progressbar</option>
    <drilldown>
      <set token="BPM">$click.value2$</set>
      <link target="_blank">https://www.youtube.com/results?search_query=$BPM$+bpm</link>
    </drilldown>
  </single>
</panel>
```



- Used a search-driven lookup to combine tracks with bpm to make it easy to enrich at search-time
- Removed empty values from the results
- Tracked the average BPM per month over time.
- Added a drilldown for the BPM value so I could look up songs with that BPM.

splunk> .conf18

Songkick dashboard XML

```
<row>
  <panel>
    <title>Upcoming Concerts for Artists</title>
    <table>
      <search>
        <query>index=music sourcetype=songkick | mvexpand concert_artist | table concert_name, concert_artist,concert_datetime, concert_venue | dedup
          concert_artist</query>
        <earliest>$field1.earliest$</earliest>
        <latest>$field1.latest$</latest>
        <sampleRatio>1</sampleRatio>
      </search>
      <option name="count">100</option>
      <option name="dataOverlayMode">none</option>
      <option name="drilldown">none</option>
      <option name="percentagesRow">false</option>
      <option name="refresh.display">progressbar</option>
      <option name="rowNumbers">false</option>
      <option name="totalsRow">false</option>
      <option name="wrap">true</option>
    </table>
  </panel>
  <panel>
    <html>
    <img src="/static/app/TA-songkick-add-on-for-splunk/images/powered-by-songkick-black.png"/>
  </html>
  </panel>
</row>
```

```
  <panel>
    <title>Artists Not On Tour</title>
    <table>
      <search>
        <query>index=cim_modactions sourcetype=modular_alerts:get_concerts artist_mbid!="" | table artist_mbid,signature |lookup artist2mbid.csv artist_mbid
          OUTPUT artist | eval Concerts=if(like(signature, "No concerts for artist%"), "No Concerts", "unknown") | search Concerts="No Concerts" | table artist
          ,Concerts</query>
        <earliest>$field1.earliest$</earliest>
        <latest>$field1.latest$</latest>
        <sampleRatio>1</sampleRatio>
      </search>
      <option name="count">25</option>
      <option name="dataOverlayMode">none</option>
      <option name="drilldown">none</option>
      <option name="percentagesRow">false</option>
      <option name="refresh.display">progressbar</option>
      <option name="rowNumbers">false</option>
      <option name="totalsRow">false</option>
      <option name="wrap">true</option>
    </table>
  </panel>
```

- The first row has one panel that is using the events created by the Songkick alert action and the other panel displays the attribution image.
- The second row is a panel that is taking data from the log, uses a lookup to get the artist display names, and display the artists that do not have concerts.

splunk> .conf18