# Ripplify: Music Taste Visualization for Nostalgia & Joy

Erik Kieran Boesen | Advised by Stephen Slade, Dept. of Computer Science, Yale University


## Abstract

I present a data visualization platform that enables users to examine their entire Spotify music listening history at a glance. I document the challenges encountered and best practices identified while building the service and its infrastructure.


## Methodology

There are many websites that allow users to view their top artists and songs on Spotify, but existing approaches only scratch the surface of the data Spotify has, using only the company's public API. Through the Spotify Privacy Settings, users can request their complete Extended Listening History.

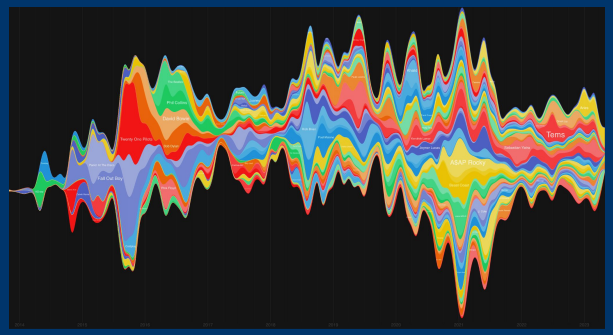
### Spotify's dataset

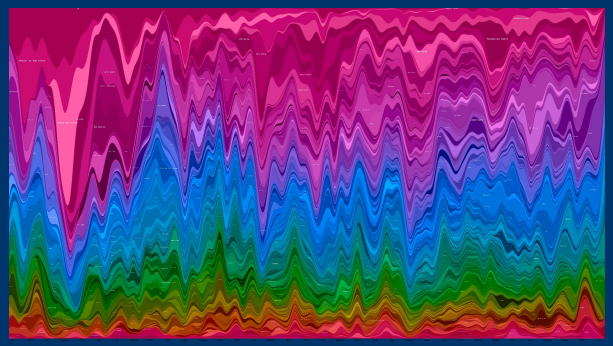
Every time a user plays a song, Spotify stores:

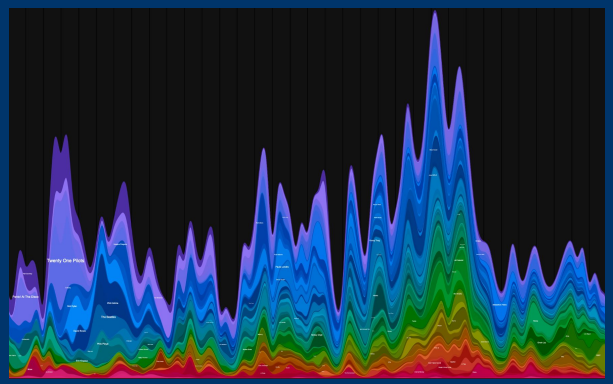
> Track name, ID, album, artist
> Timestamp of play & ms for which it was played
> Country, IP, operating system, & user agent of device
> How the song played: directly clicked, after previous song, at start of playlist selection, after a skip, etc.
> How the song stopped: ended, paused, skipped, other song selected, Spotify quit, etc.
> In shuffle mode, offline, in private listening session?
> So much more!

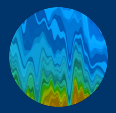
The goal of making this data of this nature intuitively legible in graphical form is achieved using "stream graphs" as discussed in Byron & Wattenberg (2008). I used D3 to render stream graphs wherein the X axis represents time and Y the amount the user listened to a each artist (or, optionally, each song by a single artist) during the history of their account.

This graph format allows individuals to quickly observe how their taste in music has evolved over time. Beta testers have described spending long periods of time looking closely at their results and noticing moments in their music listening history that they associate with memorable life phases, friendships, romances, and more. Some artists and songs enter one's life and then fade to the background, others may persist across many years, but all of them form a unique and deeply personal look at how music interacts with the human experience.



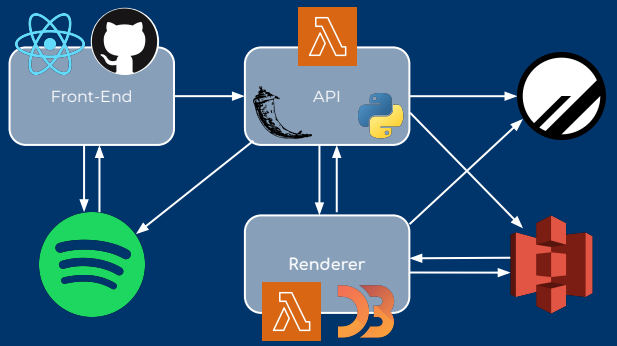






Visualize your own taste at

**ripplify.io!**




## Architecture

Ripplify has three main components:

> **Front End**: written in React, hosted with GitHub Pages. Interacts with Spotify API to perform auth, obtains OAuth tokens then forwards it to backend to obtain long-living JWT token. JWT & identity saved to localStorage. When logged in, displays instructions for uploading data then shows interface for choosing rendering configuration. That interface allows customization of graph mode (three options are shown left), color scheme, time resolution, start and end dates, whether to show artists or songs from a specific artist, how many artists/songs to show, a list of "guilty pleasure" artists to exclude from the graph, and more. Options sent to API as JSON.

> **API**: uses Python & Flask on AWS Lambda. Generates JWT tokens for auth, stores data in PlanetScale SQL DB. Accepts POST request to /history_files with .zip files of listening history from Spotify, which are stored in AWS S3. Runs task to unzip and compute summary data for rendering & stores in JSON file on S3. Accepts POST to /render which creates record in DB then triggers renderer.

> **Renderer**: separate Lambda microservice written in JS. Receives request from API w/ JSON rendering config. Uses JSDOM to emulate DOM then D3 and custom modification of d3-area-label to render graph. JSDOM then used to fetch the internal HTML of SVG which service then saves to .svg file on AWS S3. URL of this file then stored in DB. After front end sends render request, it makes API request every 5s until render has completed, then displays resulting SVG.

## References


Byron, Lee. *GitHub: Leebyron/Streamgraph*. 2010, https://github.com/leebyron/streamgraph.

Byron, Lee & Wattenberg, Martin. *Stacked Graphs – Geometry & Aesthetics*. 2008, https://leebyron.com/streamgraph/stackedgraphs_byron_wattenberg.pdf.