



SLAP!



by *M&M Team*

Intro

Architecture

Demo

Intro

Members

6110545619 Mai Norapong

@MaiNorapong || @MaiMee1

6110546011 Tharathorn Bunrattanasathian @th-bunratta

Motivation

We loved Elle's Game of Song Association, wanted to play it, and no current API facilitates it, so we made our own.

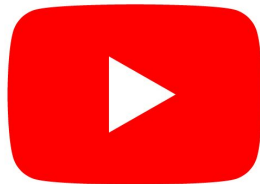
Pain Points

People can search lyrics, can see the popularity of songs (barely), but not popularity of words.

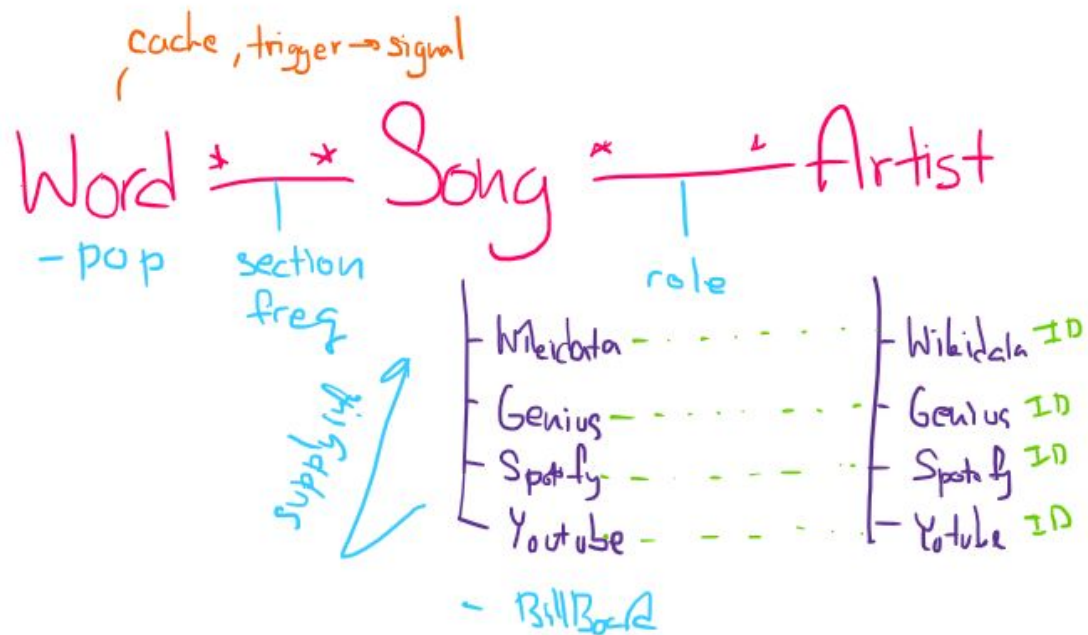
Data Sources

- Billboard
- Spotify
- Genius
- YouTube
- Wikidata

billboard



Data Model



Data Model

- Song joins data together and chooses what to supply to client / API

```
@property
def title(self) -> str:
    if self.wikidata_song:
        if self.wikidata_song.title:
            return self.wikidata_song.title
    if self.genius_song:
        if self.genius_song.title:
            return self.genius_song.title
    if self.spotify_song:
        if self.spotify_song.title:
            return self.spotify_song.title
    raise ValueError("No title available for this Song.")
```

Data Model

- Data sources are slottable

```
def link_to_spotify(self):
    from app_v2.models.spotify import SpotifySong
    if self.spotify_song is not None:
        # already linked
        return
    spotify_song = SpotifySong.retrieve_song(self.title, [self.artists.first().name])
    self.spotify_song = spotify_song
    self.save()

def link_to_genius(self):
    from app_v2.models.genius import GeniusSong
    if self.genius_song is not None:
        # already linked
        return
    # Use spotify title bc service is related
    genius_song = GeniusSong.retrieve_song(self.spotify_song.title, [self.artists.first().name])
    self.genius_song = genius_song
    self.save()
```

Components

- Django API server
 - Django REST framework
- Vue visualization app