HW 4 – SoundCloud (jQuery & Ajax Wrap-up) - 30 pts

Introduction:

The goal of this assignment is to continue to build your JavaScript & jQuery skills. For this assignment we are going to recreate some of the functionality of music streaming site SoundCloud. Your task is to build a web page that supports the following interactions. Be sure to read through all of the required interactions before beginning. Use the **hw5-skeleton.js** file for the two functions listed in the instructions. This file can be found in the 'javascript' folder of the GitHub repo. You will also be using the Stratus library, which is a jQuery powered SoundCloud player - include the library using the CDN "**https://stratus.soundcloud.com/stratus.js**". Make sure to also include jQuery 1.7.

The two functions in hw5-sekeleton.js includes all the syntax and query parameters that you require. If interested, you may find concise documentation here:

- Stratus: https://stratus.soundcloud.com/
- Soundcloud: https://developers.soundcloud.com/docs/api/reference#tracks

Styling is not the primary focus for this assignment, so don't worry about making your page beautiful (of course, you can if you want). You're more than welcome to use any CSS framework.

Requirements (the things for which you'll be graded)

1. QUERYING SOUNDCLOUD'S API (9 total points)
    i. (2 point) User can input text to search SoundCloud's API
    ii. (1 point) Use the provided callAPI() function.
        a. Hint: You'll need to pass in the user's search query as the argument for the callAPI() function.
    iii. (6 points) The first 20 results from the response object are rendered on the page. Each song result displays the following information for that song:
        a. Song title
        b. Artist
        c. Picture

2. PLAYING SONGS VIA THE STRATUS PLUGIN (5 total points)
    i. (3 points) User can click a 'play' button attached to each song result to play that song via the Stratus player plugin.
    ii. (2 points) Use the provided changeTrack() function and pass in the song's permalink URL as the argument for changeTrack().
3. Hint: Find the 'permalink_url' field for each song in the SoundCloud API response object and pass this URL as the argument for the changeTrack() function.
    i. Cut and paste the following link to the Stratus player JS file in your HTML file: <script type='text/javascript' src="stratus-player.js"></script>

4. CREATING A PLAYLIST (12 total points)
    i. User can create a playlist of songs by picking individual songs from the search results.
        a. (2 points) User can click an 'add to playlist' button attached to each song result to add a song to the playlist.
        b. (2 point) Songs are added to the top of the playlist.
        c. (2 point) Songs are not 'removed' from the search results when they're added to the playlist. Hint: use the jQuery .clone() method to prevent them from being removed from the search results.
        d. (2 point) User can remove songs from the playlist. Hint: use the jQuery .remove() method.
        e. (2 points) User can move songs up or down in the playlist one spot at a time. Hint: use the jQuery .prev(), .next(), .insertBefore(), and .insertAfter() methods.
        f. (2 point) User can play songs that are in the playlist. This play functionality is the same as the play functionality in the search results.

5. CODING STYLE (4 total points)
    i. (1 point) Your site has a title and intuitive labeling or instructions on how to use it.
    ii. (1 point) Your JS and CSS code is separate from your HTML (i.e., there should be no inline CSS styling or JavaScript within your HTML file).
    iii. (1 point) All of your JavaScript is commented appropriately.
    iv. (1 point) Your code is clean and concise (minimal repetition and unused code).

6. Bonus (You can get up to 4 extra points for doing both of these)
    i. (2 points) 'RELATED' SONGS
        a. When a user plays a song, a set of 'related' songs are loaded into a new section of the page.
        b. These 'related' songs are found via taking the first 'tag' from the tag_list field found in each song in the SoundCloud API response object and then querying the SoundCloud API using this 'tag' as the 'q' attribute of the data object of the $.get() method.
        c. These 'related' songs can be played or added to the playlist just like how the regular search results.
        d. The 'related' songs section is loaded independently from the main search results.
    ii. (2 points) 'INFINITE' SCROLL
        a. Use a jQuery library of your choosing to load the next 20 results from the SoundCloud API response object each time the user scrolls to the bottom of the main results. There will only be 200 total results, so

don’t be alarmed if the ‘infinite’ scroll stops after 10 additional result loads.