

Create a playlist

Stated Requirements

I have decided to make a playlist featuring short songs from three albums from three different decades. Given a list of albums and their corresponding songs, output a list of songs in order from shortest to longest, where each song is less than or equal to the median song in terms of length. Finally, output the total time of the playlist.

Solution Snapshot

Solution has been implemented in JavaScript. I decided to render the result of the playlist in a KendoUI grid, so I create an index.html that the javascript parsing logic and rendering logic to present the playlist. In <head> tag I added the required resources for KendoUI grid and jQuery. The implementation logic lies in the script tag of index.html file.

How to run the index file

In order to run index.html without hosting on a web server, we will have to launch an instance of chrome browser with web-security disabled. This needs to be done to have chrome access to your local file system and override browser errors related to cross domain access to resource files.

Windows:

```
C:\Users\pvirk>"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --user-data-dir="C:/Chrome dev session" --disable-web-security
```

Mac

```
alias cchrome='/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --user-data-dir="/tmp/chrome_dev_session" --disable-web-security'
```

Please make sure that Albums.txt and index.html are in the same folder. Launch the index.html file by dragging to the browser or by choosing to open the file using file menu.

Solution Breakdown

1. By making a jQuery 'get' call, get all the contents of the Album.txt.
2. Parse the file on delimiter \n\n get an array of string representation of Albums.
3. For item (i.e. string representing album) in the list created in step 2, parse the strings on delimiter \n to get a collection of strings that represent songs.
4. The first item in list created in step 3 would represent Artist, Album, Year information, so we will parse this on " / " delimiter to get the corresponding fields.
Rest of the items in the list represents songs under the given album and we can extract title and duration information by parsing the items on " - " delimiter.
5. After going through each album we have a complete list of songs in Album.txt as an array representation of songs objects. Each song object have following attributes:
 - a. Artist

- b. Album
 - c. Year
 - d. Title
 - e. Duration
 - f. Duration in Minutes (we need this field for sorting, finding the median and calculating duration of playlist)
6. Perform sort on the array of songs. I did this by passing an anonymous compare function to sort method of songs array.
 7. Calculate the median duration of songs using sorted array of songs created in step 6. I defined function `get_median_duration` to calculate median duration.
 8. Filter the array of the songs so that we only have items that have song duration less than the median value. To achieve this I passed an anonymous function as a parameter of filter method of songs array. This returns me the filtered playlist that contains the songs that are sorted from shortest to longest duration and also less than or equal to the duration of median song length.
 9. Calculate total length of Playlist by passing an anonymous function to the reduce method on playlist array.
 10. Finally I rendered the results in a KendoUI grid and display the total duration of playlist at the bottom of the grid.
 11. Index.html page should look like the following screen shot

:

Index.html					
file:///C:/Users/pvirk/Documents/My%20Documents/Apple/CodingChallenge/CodingChallenge/Index.html					
Artist	Album	Year	Song Title	Duration	Duration In Seconds
Vampire Weekend	Contra	2010	Holiday	2:18	138
Love	Forever Changes	1967	Bummer In The Summer	2:24	144
Vampire Weekend	Contra	2010	Cousins	2:25	145
XTC	Drums and Wires	1979	Limelight	2:27	147
Vampire Weekend	Contra	2010	California English	2:30	150
XTC	Drums and Wires	1979	Chain Of Command	2:34	154
XTC	Drums and Wires	1979	Outside World	2:41	161
XTC	Drums and Wires	1979	That Is The Way	2:57	177
Vampire Weekend	Contra	2010	White Sky	2:59	179
Love	Forever Changes	1967	Old Man	3:02	182
Love	Forever Changes	1967	The Good Humor Man He Sees Everything Like This	3:08	188
XTC	Drums and Wires	1979	Day In Day Out	3:08	188
XTC	Drums and Wires	1979	Ten Feet Tall	3:17	197
Love	Forever Changes	1967	Alone Again Or	3:17	197
Love	Forever Changes	1967	Andmoreagain	3:18	198
Total Playlist Duration: 56:16					