

Feature request: include preview sample.

December 27th 2017

OBJECTIVE

To give users the ability to listen to a preview sample.

BACKGROUND

In the current version of the app, the user can search for songs and add them to a playlist. It would enrich the user experience if they have the option to listen to a preview of a song to ensure it is the song they desire to add or to simply listen if the selected song is what they desire to add to their playlist.

The feature would accomplish the following:

- Display a play/pause button next to the tracks in the search results and in the playlist.
- Enabling the above mentioned button would play the audio of sample of the selected song.
- Disabling it would pause the played song.
- Enabling the button next to another song while a sample is being played will disable the current preview.

TECHNICAL DESIGN

Play or pause a track

A new feature to play audio files, retrieved from the Spotify API, will be created that will be implemented in the Track and App components. On render it will display a play button for each track. After enabling it for a particular track the button displays a pause option so that the user can pause the sample.

In Track.js the state will be initialised to contain a boolean to give information about the displaying mode, which is either the state of play or pause and initially will be play. If a user

decides to play a sample song the state of play or pause will be set to pause for the particular track chosen to be played.

To activate the playback from a song, a new method called `Spotify.playTrack(trackUri)` in `Spotify.js` will retrieve the playback function from the Spotify API at the <https://api.spotify.com/v1/me/player/play> endpoint. In a PUT request the track uri will be sent for playback. The `trackUri` object that is given upon calling `Spotify.playTrack(trackUri)` will be a string containing the uri that will be sent in the PUT request.

Two new methods will be added to `App.js`, `playTrack(track)` and `pauzeTrack(track)`. Both will be given a track object which is used to either be played or paused.

To retrieve the track uri of the given track object we will iterate the array of both the search results and playlist tracks to match the URI of the selected track to which the track URI will be stored to a `trackURI` variable. This variable will be used in the PUT request to call upon the playback in the Spotify API.

After retrieving the track uri the method `Spotify.playTrack(trackUri)` will be called in the `playTrack(track)` function or `Spotify.pauseTrack(trackUri)` for the `pauzeTrack(track)` function in `App.js`.

To call the `playTrack(track)` method or `pauseTrack(track)` method in `App.js` at the time a user clicks on the play or pause button, we should pass `this.playTrack` and `this.pauseTrack` to an `onPlay` and `onPause` attribute to the component render of `SearchResult` and `Playlist`. In both `SearchResult.js` and `Playlist.js` the attributes `onPlay` and `onPause` that are given the `this.props.onPlay` and `this.props.onPause` objects and will be passed into the `TrackList` components. In `TrackList.js` the same will be passed to the `Track` component.

In `Track.js` two methods will be added: `playTrack()` and `pauzeTrack()`. The `playTrack()` method will activate the `playTrack` method with an track object from `App.js` that is given as a prop: `this.props.playTrack(this.props.track)`. The `pauzeTrack()` method will activate the `pauzeTrack` method with an track object from `App.js` that is given as a prop: `this.props.pauzeTrack(this.props.track)`. Both methods will be bound in the constructor of `Track.js`.

To render the button that displays a play sign or a pauze sign the `renderAction()` method in `Track.js` will be modified to display the additional button. In the constructor of `Track.js` a boolean `displayPlay` will be created an initially set to true.

The `renderAction()` method will check if the `this.displayPlay` boolean from the class is true or false. When true the method will render a play button with an element that contains the `onClick` attribute and triggers `this.playTrack` when a user clicks the button element. When

false the method will render a pause button with an element that contains the onClick attribute that triggers this.pauzeTrack when a user clicks on the element.

Upon rendering the app all tracks will show a play button. To the playTrack() method in Track.js a new line of code will be added after calling upon this.props.playTrack(this.props.track): this.displayPlay = false, which will set the Track class boolean to false, so that upon a users action to play a track, the app will render a pause button for the selected track. This will happen by the else statement in the renderAction() method.

CAVEATS

Different design

In the current implementation of the design it will be unclear for the user what the duration of a song is and how many seconds they have listened to it so far. This information is normally always available in apps that play audio files. To improve the user experience it should be evaluated if this should be added to the design.