[1] When the user is in *index.html* and search for a track, the *searchQuery* is forwarded to the *SearchController*. There we call the method *searchTrack(searchQuery)* on the *SpotifyResource* class, which will give us the *spotifyTracks*. If the *spotifyTracks* object is null, then we are redirected to *error.jsp*; else, the *spotifyTracks* are forwarded to *searchResults.jsp.*

[2] Once we are in *searchResults.jsp* and the user clicks on one of the shown tracks, the *spotifyTrackId* is forwarded to the *TrackController*. There we call the method *getTrack(spotifyTrackId)*, which will give us the *spotifyTrack*. If the *spotifyTrack* is null, then we are redirected to *error.jsp*. We also call the method *searchVideos(spotifyTrack.artists.name + spotifyTrack.name)*, which will give us the *youtubeVideos*. And finally, we call the method *getLyrics(spotifyTrack.name, spotifyTrack.artists.name)*, which will give us the *lyrics*. If the *youtubeVideos* or the *lyrics* are null, then we are redirected to *error.jsp*; else, the *youtubeVideos.videoId* and *lyrics.lyrics* are forwarded to *trackView.jsp*. There we have two different optional actions.

[3] The first optional action is adding a song to a Spotify account. If we click on the “add to Spotify” button, the *spotifyTrackId* is forwarded to the *SpotifyTrackLibraryController*. We proceed to authenticate the user’s Spotify account forwarding the *clientId* and the *clientSecret* to the *OAuthController* so that an *accessToken* is forwarded to the *SpotifyTrackLibraryController*. There we call the method *addTrack(spotifyTrackId)* on the *SpotifyResource*, which will give us either *true* or *false*. If the response is *false*, then we are redirected to *error.jsp*; else, the *spotifyTrackId* is forwarded again to the *TrackController* so we can get to the *trackView.jsp* where we were as we described before.

[4] The second optional action is sharing some lyrics on a Facebook account. If we choose a part of the lyrics and click on the “Share on Facebook” button, the *lyricsQuote* and the *spotifyTrackId* is forwarded to the *FacebookShareLyricsController*. We proceed to authenticate the user’s Facebook account forwarding the *clientId* and the *clientSecret* to the *OAuthController* so that an *accessToken* is forwarded to the *FacebookShareLyricsController*. There we call the method *publishPost(lyricsQuote)* on the *FacebookPostResource*, which will give us true. Then, the *spotifyTrackId* is forwarded again to the *TrackController* so we can get to the *tracView.jsp* where we were as we described before.