

Session	
Token	String
UserAccountID	Binary
Expiration	Date

#### Queried by:

Name

- Queried by:
  - TokenUserAccountID
  - Expiration

UserProfile		
Binary		
Binary		
String		

SongList		
ID	Binary	
UserProfileID	Binary	
Title	String	
LastModificationDate	Date	

SongListEntry		
ID	Binary	
SongListID	Binary	
Rank	Number	
Title	String	
Artist	String	
ContentURL	String	

# Queried by:

Name

UserAccountID

UserProfileID

Queried by:

SongListID

Queried by:

# Flows

## Create new song list

A user logs in. The user navigates to the page for song list creation. The user provides a title for the song list and presses "Create".

- The client sends a POST request to the /api/song-list API endpoint. The request contains the session token as a cookie and the song list's title in the body (as JSON).
- Verify that the received session token is valid. Take the UserAccountID field of the Session instance to find the corresponding UserProfile instance.
  Temporarily cache the ID field of the UserProfile instance for later use.
- A new SongList instance is created. The ID field is randomly generated and verified to be unique. The UserProfileID field is taken from the previously cached value. The Title field is taken from the request body. The LastModificationDate field is taken from the current date.

## Modify existing song list

A user logs in. The user modifies their song list by adding a new song to a specific location (rank) within the list, replacing an existing song, or changing the title of the song list. In the case that a new song is being added, the user provides an URL that represents the song (YouTube, Spotify, etc.).

- The client sends a PUT request to the /api/song-list API endpoint. The request contains the session token as a cookie and the new final state of the song list in the body (as JSON). Note that the body describes the final state, not the set of changes.
- Verify that the received session token is valid. Take the UserAccountID field of the Session instance to find the corresponding UserProfile instance.

  Temporarily cache the ID field of the UserProfile instance for later use.
- Use the cached ID value of the user's UserProfile instance to query for the corresponding SongList instance. The Title field of the SongList instance is modified to the received title. The LastModificationDate field is modified to the current date.
- Use the ID field of the SongList instance to query for all corresponding SongListEntry instances. All of these SongListEntry instances are deleted.
- A new SongListEntry instance is created for each song listed in the request body. The ID field is randomly generated and verified to be unique. The SongListID field is taken from the ID field of the user's SongList instance. The Rank field is determined by where the user has placed this new entry. The Title and Artist fields are extracted from the URL the user has provided. The ContentURL field is taken from the URL the user has provided.

#### View existing song list

A user looks up the song list for an existing user. Note that the user does not have to be logged in.

- The client sends a GET request to the /api/song-list API endpoint. The URL must look like the following. http://[domain]/api/song-list/[user's profile name]
- The UserProfile instance is queried for by the Name field, using the value provided in the path.
- The SongList instance is queried for by the UserProfileID field, using the ID field of the UserProfile instance.
- All relevant SongListEntry instances are queried for by the SongListID field, using the ID field of the SongList instance.
- All of the queried information is formatted (into JSON) and sent back to the client.

