**ACTION MAP – soundcloud**

**CONTROLLERS**

1. **UserController – 11**

**POST – 4**

@/login (body – LoginDTO)

@/logout (user from session)

@/register (body – RegisterDTO)

@/followuser/{user-ID} (user from session)

**DELETE – 2**

@/remove-user (user from session)

@/unfollow-user/{user-ID} (user from session)

**GET – 3**

@/my-profile (user from session)

* // Get all user information for the currently logged user

@/users/{username}

* // Get specific user and public information for their profile

@/users

* // Return list of all users with no parameters, or search using @/users?**query**=”**USERNAME**”
* [https://@/users?**sort-param**=**value**] Returns list sorted by **sort-param** according to **value** (**DESC/ASC**)
* @RequestParam(required = false) String **query**
* @RequestParam(required = false) String **byFollowers**
* @RequestParam(required = false) String **bySongs**
* @RequestParam(required = false) String **byComments**
* @RequestParam(required = false) String **byKarma**

**PUT – 2**

@/users/{username}/picture (body – binary image)

* Update user profile picture

@/users/{username}/update-profile (body – UpdateDTO)

* Update user personal information

1. **SongController - 8**

**POST – 1**

@/add-song (body – SongDTO, user from session)

**DELETE** – 1

@/songs/{ID}/remove (user from session)

**GET – 5**

@/songs/{song-ID}/play

* //Retrieve the mp3 file for the song

@/songs/{song-ID}

* //Return information for {song-ID}

@/songs/liked (user from session)

* //Return all liked songs for current user

@/{username}/songs

* //Return all songs for {username}

@/songs

* // Return list of all songs with no parameters
* // Search using @/songs?**query**=”SONGNAME”&**sort-param=**VAL
* // Sort using any sort parameter=value pair. ASC / DESC for sorting order
* @RequestParam(required = false) String **query**
* @RequestParam(required = false) String **byComments**
* @RequestParam(required = false) String **byLikes**
* @RequestParam(required = false) String **byDislikes**

**PUT – 1**

@/songs/{song-ID}/set-like-status?**value**=<1 / 0 / -1> (user from session)

* // Set the current user’s like status for song {ID} according to the request parameter
* @RequestParam int **value**
* **value**=1 -> like song
* **value**=0 -> set song to neutral
* **value**=-1 -> unlike song

**3.CommentController - 5**

**POST –** 1

@/songs/{song-ID}/add-comment (body – comment text, user from session)

**DELETE –** 1

@/comments/{comment-ID}/remove (user from session)

**GET –** 2

@/songs/{song-ID}/comments

@/comments/{comment-ID}

Not needed:

@comments/mostlikes

@comments/mostdislikes

**PUT – 1**

@/comments/{comment-ID }/set-like-status?**value**=<1 / 0 / -1> (user from session)

* // Set the current user’s like status for {comment-ID} according to the request parameter
* @RequestParam int **value**
* **value**=1 -> like comment
* **value**=0 -> set comment to neutral
* **value**=-1 -> unlike comment

**4.PlaylistController - 7**

**POST –** 2

@/add-playlist (body – name, user from session)

@/playlists/{playlist-ID}/add-song (body – song\_id, user from session)

**DELETE –** 2

@/playlists/{playlist-ID}/remove-song (body – song\_id, user from session)

@/playlists/{playlist-ID}/delete-playlist (user from session)

**GET –** 3

@/playlists

* Return all playlists

@/playlists/{playlist-ID }

@/users/{username}/playlists

* Return all playlists by user

Not needed:

@/playlists/mostsongs

**DAOs**

1. UserDAO
2. SongDAO
3. CommentDAO
4. PlaylistDAO

**POJOs**

1. User
2. Song
3. Comment
4. Playlist

**DTOs**

1. LoginDTO (username, password)
2. RegisterDTO (username, password, email, age)
3. UpdateDTO (newPassword, newEmail, newAge, newCity)
4. SongDTO (title, description, tags, binary song file)