DataBase Project

Spotify System

Alireza Lorestani

Dr. Behshid Behkamal

2019-2020

1. users can register by calling:

   POST: http://localhost/spotify/auth/register.php

   the Body will be:

```
{
        "username" : "reza",
        "password" : "Ab123456",
        "password2" : "Ab123456"
        "recovery_question":"FavoriteColor?",
        "recovery_answer" : "red"

}
```

2. After that users should choose their rule (Artist or Listener) :

   Artist :

   POST: http://localhost/spotify/auth/register-artist.php

```
{
 "user_id" : 5,
 "name" : "AliReza Ghanari",
 "nationality" : "usa",
 "begin_year" : 1998
}
```

   Listener:

   POST: http://localhost/spotify/auth/register-listener.php

```
{
 "user_id" : 4,
 "first_name" : "kamal",
 "last_name" : "barzegar",
 "nationality" : "iran",
 "birthday_year" : "1348"
}
```

3. Now they can login into system :

   POST: http://localhost/spotify/auth/login.php

```
{
 "username" : "reza",
 "password" : "Ab123456"
}
```

4. admin can ban a user and remove his/her account :

    POST: http://localhost/spotify/admin/delete-user.php

```
{
 "user_id":44
}
```

5. also admin can remove a playlist or an album:

    POST: http://localhost/spotify/admin/delete-playlist.php

```
{
"playlist_id":13
}
```

    POST: http://localhost/spotify/admin/delete-album.php

```
{
 "album_id":2
}
```

6. admin can add new music to system :

    POST: http://localhost/spotify/admin/music.php

```
{
 "name": "music_name",
 "album_id" : 1,
 "duration" : 6
}
```

7. admin can remove a music :

    DELETE: http://localhost/spotify/admin/music.php?id=8

8. admin can see a list of reported musics :

    GET: http://localhost/spotify/admin/get-reports.php

9. admin can see a list of artists who are waiting for confirmation:

    GET: http://localhost/spotify/admin/verify-artists.php

10. admin can verify artists or reject them:

    POST: http://localhost/spotify/admin/verify-artists.php

```
{
 "artist_id":5,
 "verify":true
}
```

    if the verify parameter equals true, the artist will be verified. otherwise he/she will be rejected.

11. Artists can add their new album and its musics:

    POST: http://localhost/spotify/artist/album.php

```
{
        "name":"album mamad",
        "genre":"rap",
        "publish_datetime":"2020-10-16",
        "musics":[
        {
        "name":"music 1",
        "duration":100
        },{
        "name":"music 2",
        "duration":37
        }
        ]
}
```

12. Artists can remove their albums:

    DELETE: http://localhost/spotify/artist/album.php?album_id=5

13. Artists can remove their musics:

    DELETE: http://localhost/spotify/artist/album.php?music_id=19

14. Listeners can play maximum 5 musics per day by calling this API. there is no limit for premium members.

    GET: http://localhost/spotify/listener/play.php?music_id=100

15. Listener can change their account type to premium by calling :

POST: http://localhost/spotify/listener/buy-premium-account.php

```
{
 "card_number":"6037909038712345",
 "card_expire":"2024-11-23",
 "expiration_datetime":"2021-12-11T08:20:10"
}
```

16. Listener can make a playlist by calling:

POST: http://localhost/spotify/action/playlist/add-playlist.php

```
{
 "name":"playlist mamad2"
}
```

17. call this API to add the music to playlist:

POST: http://localhost/spotify/action/playlist/add-music-to-playlist.php

```
{
 "playlist_id":15,
 "music_id":1
}
```

18. and you can remove musics from playlist :

POST: http://localhost/spotify/action/playlist/delete-playlist.php

```
{
 "playlist_name":"playlist mamad2"
}
```

19. Listeners can like playlists:

POST: http://localhost/spotify/action/playlist/like-playlist.php

```
{
 "playlist_id":14
}
```

20. Listeners can add other listeners to their playlist:

    POST: http://localhost/spotify/action/playlist/add-user-to-playlist.php

```
{
 "playlist_id" : 14,
 "user_id":1
}
```

21. Users can report a music:

    POST: http://localhost/spotify/action/music/report-music.php

```
{
 "music_id" : 2,
     "description ":"قوانین کپی رایت رعایت نشده."
}
```

22. Users can like a music:

    POST: http://localhost/spotify/action/music/like-music.php

```
{
 "music_id":2
}
```

23. Users can follow other users and noticed what they are listening to:

    POST: http://localhost/spotify/action/follow.php

```
{
 "follower_id":5,
 "followee_id":4
}
```

24. Users can delete their account :

    GET: http://localhost/spotify/action/delete-account.php

Searches:

25. users can search a word, the system will search the word among other users, musics, albums, artists etc.

    GET: http://localhost/spotify/searches/search.php?search=music

26. Users can see other users profile:

`GET:` http://localhost/spotify/searches/profile.php?user_id=5

27. Users can see a list of their followings :

`GET:` http://localhost/spotify/searches/followers.php?action=following&user_id=5

28. Users can see a list of their followers:

`GET:` http://localhost/spotify/searches/followers.php?action=follower&user_id=5

29. Users can see musics of a playlist:

`GET:` http://localhost/spotify/searches/musics.php?source_type=playlist&source_id=14

30. Users can see musics of an album:

`GET:` http://localhost/spotify/searches/musics.php?source_type=album&source_id=1

31. Listener can see their following's last-played music:

`GET:` http://localhost/spotify/searches/last-played-music-by-following.php

32. admin can see premium accounts expiration date :

`GET:` http://localhost/spotify/searches/check-premium.php

33. Listener can see their following Artist's last 5 musics:

`GET:` http://localhost/spotify/searches/followed-artist-music.php

34. five most popular musics in last month:

`GET:` http://localhost/spotify/searches/week-popular-musics.php

35. by calling this API system offer 2 musics based on their favourite genre in a specific playlist:

`GET:` http://localhost/spotify/searches/offer-music-by-genre.php?playlist_id=14

36. offer recently released music and most popular music based on user’s activity:

`GET:` http://localhost/spotify/searches/offer-music-by-activity.php

37. Artists can see a list of their fans:

`GET:` http://localhost/spotify/searches/find-fans.php

38. Listeners can see a list of Artists with the same nationality:

`GET:` http://localhost/spotify/searches/same-nationality.php

39. admin can see a list of artist sorted by their activity:

`GET:` http://localhost/spotify/searches/artist-by-activity.php

40. admin can see a list of the most inactive artists:

`GET:` http://localhost/spotify/searches/least-activity.php

41. admin can see a list of un normal users:

`GET:` http://localhost/spotify/searches/unnormal-users.php