ELEMENT MUSIC
JAVA Backend Interface Documentation

Table of contents

1. interface information

1.1.Song

1.1.1. add song

Interface: "/song/add"
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
file	MultipartFile

Body:

parameter name	type of data
songName _	String _
musicianId	String
musicianName	String
description	String
pic	String
lyric	String

Example:
Param: The Longest Movie.mp3
Body: {
songName: "The Longest Movie "
musicianId: "2"
musicianName: "Jay Chou"
description: " Included in "I'm Busy""
pic: "/img/songPic/tubiao.jpg"
lyric: " ...give me two more minutes to freeze my memory..."
}

1.1.2. back to all songs

interface: "/ song /all"
Calling method: RequestMethod.GET
Returns: song details [json objects]

1.1.3. by song id

interface: "/ song Id / detail"
Calling method: RequestMethod.GET
parameter:
Body:

parameter name	type of data

id	String

Example:
Body: {
id: " 2 "
}

Returns: song details json object returns null if there is no song

1.1.4. Find songs by song title

Interface: "/songName / detail"
Calling method: RequestMethod.GET
parameter:
Body:

parameter name	type of data
name	String

Example:
Body: {
name : " echo "
}

Returns: song details json object returns null if there is no song

1.1.5. delete song

interface: "/ song /delete"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
id	String

Example:
Body: {
id: " 2 "
}
returns: boolean (false or true)

1.1.6. Update song information

Interface: "/song/update"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
id	String
songName _	String _
musicianId	String
description	String
lyric	String

Example:
Body: {
id: "1"
songName : " Gangnam "
musicianId: "2"

description: "…"
lyric: "... circle circle circle circle ..."
}
Returns: json object ({ "msg": "modification successful", "code": 1} or { "msg": "modification failed", "code": 0})

1.1.7. Update song image

Interface: "/song/img/update"
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
file	MultipartFile
id	long

Example:
Param: {
File: test.jpg
Id: 1
}
Returns: json object ({ "msg": "Upload successful", "code": 1, "portraitPath": "/img/singerPic/xxx.jpg"} or { "msg": "Upload failed", "code": 0})

1.1.8. Update song URL

Interface: "/song/url/update"
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
file	MultipartFile

id	long

Example:
Param: {
File : test.txt
Id: 1
}
Returns: json object ({ "msg": "Upload successful", "code": 1, "urlPath": "/img/singerPic/xxx.jpg"} or { "msg": "Upload failed", "code": 0})

1.1.9. Returns all songs by the specified artist

Interface: "/song/songsByMusicianId"
Calling method: RequestMethod.GET
parameter:
Param:

parameter name	type of data
id	long

Example:
Param: {
Id: 1
}
Returns: song details json object returns null if there is no song

1.1.10. Return to singer's avatar

Interface: "/song/songPicture"
Calling method: RequestMethod.GET
parameter:
Param:

parameter name	type of data
id	long

Example:
Param: {
id: "2"
}
Returns: string (Example: "/img/songPic/1615938237513Sksolari.jpg")

1.1.11. Download songs (this function is not open for now)

Interface: "/song/ download "
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data

fileName	String

1.2.Musician _

1.2.1. Add singer

Interface: "/musician/add"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
name (cannot be empty)	String
sex	String
pic	String
birth	String
location	String
description	String
portrait	String
musicType (not null)	String (choose among the following types) • POP • R OCK • ELECTRONIC • FOLK • CLASSICAL • JAZZ • ABSOLUTE_MUSIC • RAP • METAL • WORLD_MUSIC • NEW_AGE • AMBIENT_MUSIC • INDIE

Example:
Body: {
name: " Jay Chou"
sex: " m ale" (only " male " and "female")
pic: "/img/songPic/tubiao.jpg"
birth: "1979-01-18" (fixed format)
location: “Taiwan”
description: " Taiwanese pop singer, original musician, actor, director, screenwriter"
portrait: “ zhoujielun.jpg”
musicType: " J AZZ" (JAZZ / RAP all caps)
}
return: json object ({"msg":"Uploaded successfully","code":1,"urlPath":"/song/test5.mp3"} or ({"msg":"Upload failed","code": 0 ,"})

1.2.2. back to all singers

Interface: "/musician/all"
Calling method: RequestMethod.GET
Returns: singer details [json objects]

1.2.3. Find an artist by artist name

Interface: "/musician/name/detail"
Calling method: RequestMethod.GET
parameter:
Body:

parameter name	type of data
name	String

Example:
Body: {
name: "Jay Chou"
}
Returns: singer details json object

1.2.4. Find singers by singer id

Interface: "/musician/ id /detail"
Calling method: RequestMethod.GET
parameter:
Body:

parameter name	type of data
musicianId	String

Example:
Body: {
musicianId: " 1 "
}
Returns: singer details json object

1.2.5. remove singer

Interface: "/musician/delete"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
id	String

Example:
Body: {
id: "2"
}
returns: boolean (false or true)

1.2.6. Update artist information

Interface: "/musician/update"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
id	String
name	String
sex	String
birth	String
location	String
description	String

Example:
Body: {
id: "2"
name: " Jay Chou"
sex: " male"
birth: "1979-01-18"
location: “Taiwan”
description: " Taiwanese pop singer, original musician, actor, director, screenwriter"
}
Returns: json object ({ "msg": "modification successful", "code": 1} or { "msg": "modification failed", "code": 0})

1.2.7. Update singer avatar

interface: "/musician/ por trait/update"
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
file	MultipartFile
id	long

Example:
Param: {
file: singer.jpg
id: "2"
}
Returns: json object ({ "msg": "Upload successful", "code": 1, "pic": "/img/singerPic/xxx.jpg"} or { "msg": "Upload failed", "code": 0})

1.2.8. Return to singer's avatar

Interface: "/musician/musicianPicture"
Calling method: RequestMethod.GET

parameter:
Param:

parameter name	type of data
id	long

Example:
Param: {
id: "2"
}
Returns: string (Example: "/img/singerPic/1615938237513Sksolari.jpg")

1.3. User (Consumer)

1.3.1. Add user

interface: "/ c onsumer/add"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
username	String
password	String
sex	String
phoneNum	String
email	String
birth	String
location	String
portrait	String

Example:
Body: {
username: " Element "
password: "123456"
sex: " m ale"
phoneNum : "13800008888"
email: element@163.com
birth: " 2020-12-01"
location: “ Hong Kong ”
portrait: "element.jpg"
}
Return: json object ({"msg":"registration successful","code":1} or ({"msg":"registration failed","code": 0 ,"})

1.3.2. Check whether the login is successful by email

Interface: "/ c onsumer/ login "
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
username	String
password	String

Example:
Body: {
username : “ Element@gmail.com ” _
password: "123456"
}
returns: ({
"msg": "Login successful",
"code": 0,
"sessionId": "27C31032D44DD47586BAFEC99A34F53C"
} or {
"msg": "Username and password do not match",
"code": 3
} or {
"msg": "The account is not registered",
"code": 2
}

1.3.3. return all users

interface: "/ consumer / all "
Calling method: RequestMethod.GET
Returns: user details [json objects]

1.3.4. Returns the specified id user

interface: "/ c onsumer/detail"
Calling method: RequestMethod.GET
Body:

parameter name	type of data
id	String

Example:
Body: {
id: "1"
}
Returns: user details json objects

1.3.5. delete users

interface: "/ consumer / delete "
Calling method: RequestMethod.POST
Body:

parameter name	type of data
id	String

Example:
Body: {
id: "1"
}

1.3.6. Update user information

interface: "/ consumer /update"
Calling method: RequestMethod.POST
parameter:
Body (all cannot be empty) :

parameter name	type of data
id	String
username	String
password	String
sex	String
phoneNum	String
email	String
birth	String
location	String
portrait	String

示例：
Body: {
id: “2”
username: “element”
password: “123456”
sex: “male”
phoneNum: “13800008888”
e mail: “element@163.com”
birth: "20201201"
location: “Taiwan”
portrait: "element.jpg"
}
return: json object ({"msg":"Modified successfully","code":1} or ({"msg":"Modified failed","code": 0 ,"})

1.3.7. Update user avatar

Interface: "/consumer/portrait/update"
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
file	MultipartFile
id	long

Example:
Param: {
file: consumer.jpg
id: "2"
}
Returns: json object ({ "msg": "Upload successful", "code": 1, "pic": "/img/singerPic/xxx.jpg"} or { "msg":
"Upload failed", "code": 0})

1.3.8. User logout login

Interface: "/consumer/logout "
Calling method: RequestMethod.POST
Returns: json object ({ "msg": "Successfully logged off", "code": 1 } or { "msg": "Failed to log off", "code": 0})

1.4.Song price (Price)

1.4.1. add price

interface: "/ price /add"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
Id _	String
originalPrice	String
rate	String

Example:
Body: {
id: "1"
originalPrice: "10"
rate: "0.15"
}
Return: json object ({"msg":"Success to increase song price","code": 1} or ({"msg":"Failure to increase song price","code": 0 ,"})

1.4.2. Get song price based on song id

Interface: "/ price /getPriceById"
Calling method: RequestMethod.GET
parameter:
Body:

parameter name	type of data
Id _	String

Example:
Body: {
id: "1"
}
Returns: string (song price)

1.4.3. back to all prices

Interface: "/ price /all"
Calling method: RequestMethod.GET
Returns: Song price details [json objects]

1.5. Like (wish list)

1.5.1. User adds likes to songs

Interface: "/consumer/addToWishlist "
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
songId	String
consumerId	String

Example:
Param: {
song Id: "1"
consumerId: "2"
}

Returns: json object ({ "msg": "Successfully added", "code": 1 } or { "msg": "Add failed", "code": 0})

1.5.2. Returns all user liked songs (favorites)

Interface: "/consumer/getWishlist "
Calling method: RequestMethod.GET
parameter:
Param:

parameter name	type of data
i d	String

Example:
Param: {
id: "2"
}

Returns: all songs liked by this user [json object]

1.5.3. Remove user-liked songs (favorites)

Interface: "/consumer/removeFromWishlist "
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
songId	String
consumerId	String

Example:
Param: {
song Id: "1"
consumerId: "2"
}

Returns: all songs liked by this user [json object]

1.6. User wallet (Purse)

1.6.1. Add value to specified user by user id

Interface: "/purse/addBalance "
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
id	String
addValue	String

Example:
Param: {
id : "1"
addValue: "100"
}

Returns: json object ({ "msg": "value added successfully", "code": 1 } or { "msg": "Failed to add value", "code": 0})

1.6.2. Returns the balance of the specified user by user id

Interface: "/purse/ get Balance "
Calling method: RequestMethod.GET
parameter:
Param:

parameter name	type of data
i d	String

Example:
Param: {
id: "2"
}

Returns: string balance

1.6.3. Pay by user id

Interface: "/purse/ withdraw Balance "
Calling method: RequestMethod.POST
parameter:
Param:

parameter name	type of data
i d	String
Withdraw Value _	String

Example:
Param: {
id: "2"
}

Returns: json object ({ "msg": "Successful payment", "code": 1 } or { "msg": "Payment failed", "code": 0})

1.7.Playlist

1.7.1. Add a playlist

Interface: "/palyList/add"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
name	String _
imagePath	String
description	String
songId	List of songId

Example:

http://localhost:8088/playList/add?name=test_playlist&description=haha&songId=3&songId=4&imagePath

Body: {
name: " Bedtime Playlist"
songId: [1,2,3]
description: " Included in "I'm Busy""
imagePath: "/img/songPic/tubiao.jpg"
}
Returns: json object ({ "msg": "New success", "code": 1 } or { "msg": "New failed", "code": 0})

1.7.2. Back to all playlists

Interface: "/ playList / getAll "
Calling method: RequestMethod.GET
Return: Playlist details [json objects]

1.7.3. Returns the playlist with the specified id

Interface: " / playList / getByld "
Calling method: RequestMethod.GET
parameter:
Body:

parameter name	type of data
id	String _

Example:
Body: {
id: " 1"
}

Return: playlist details json object

1.7.4. playlist with the specified id

Interface: " / playList / delete "
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
id	String _

Example:
Body: {
id: " 1"
}

Returns: json object ({ "msg": "Delete successful", "code": 1 } or { "msg": "Delete failed", "code": 0})

1.7.5. Update playlist image

Interface: "/ play List / image / update"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
id	String _
imagePath	File _

Example:
Body: {
id: " 1",
imagePath: "xxx.jpg"
}

Returns: json object ({ "msg": "Upload successful", "code": 1, "pic": "xxx.jpg" } or { "msg": "Upload failed", "code": 0})

1.8.trade (Order)

1.8.1. Add transaction

interface: "/ order /add"
Calling method: RequestMethod.POST
parameter:
Body:

parameter name	type of data
consumerId	String _
songId	String

Example:
Body: {
consumerId: " 1"
songId: "1"
}
Returns: json object ({ "msg": "Successful transaction", "code": 1 } or { "msg": "Insufficient balance", "code": 0} or { "msg": "Cannot add order", "code": 0})

1.8.2. Return all user's orders

interface: "/ order / all "
Calling method: RequestMethod.GET
Returns: transaction order details [json objects]

1.9. Purchase List (Collections)

1.9.1. Returns the songs purchased by the specified user

Interface: "/ consumer /getCollection"
Calling method: RequestMethod.GET
Returns: transaction song details [json objects]