

User Apis:

Api-1 Add User:

Functionality: Add User in music app system.

Request

POST */user/add/*

Headers

Content-Type application/json API works on application/json

Request Body

name	mandatory	Data type	Description
name	Y	String only(100)	
city	Y	String only(100)	
state	y	String only(100)	
country	y	String only(100)	
email	y	alphanumeric(100)	
contactNo	y	numeric(10)	
description	Y	String with spaces(100)	Text to show when a user search this user.
type	Y	String only	Possible values: MUSICIAN NORMAL ADMIN

Request Body:

```
{
  "name": "Yogesh",
  "city": "karnal",
  "type": "MUSICIAN",
  "state": "haryana",
  "email": "yogiverma1991@gmail.com",
  "country": "INDIA",
  "contactNo": "9717017392",
  "description": "Artist hailing from small town karnal"
}
```

Response Body:

```
{
  "status": "SUCCESS",
  "responseCode": "00",
  "message": "User added successfully",
  "userId": 19
}
```

Api-2 Update User Status:

Functionality: Update User status in music app system.

Request

POST /user/updateStatus/

Headers

Content-Type application/json API works on application/json

Request Body

name	mandatory	Data type	Description
status	Y	String only	Status of user. Possible values ACTIVE INACTIVE
userId	Y	Integer	User id of user for which status is to be updated.

Sample Request Body:

```
{
  "status": "INACTIVE",
  "userId": "18"
}
```

Response Body:

```
{
  "name": "Yogesh",
  "city": "karnal",
  "type": "MUSICIAN",
  "state": "haryana",
  "email": "yogiverma1991@gmail.com",
  "country": "INDIA",
  "contactNo": "9717017392",
  "description": "Artist hailing from small town karnal",
  "status": "INACTIVE"
}
```

Music Apis:

Api-1 Add Music:

Functionality: Add music in music app system. Music added by this user id can only change status of music.

Request

POST */music/add/*

Headers

Content-Type application/json API works on application/json

Request Body

name	mandatory	Data type	
name	Y	alphanumeric(100)	Name of music
userId	Y	Integer	User id of the user which owns this music.
location	Y	String(500)	Location where music is stored.
type	Y	String	Possible value: AUDIO VIDEO
description	Y	String with spaces(100)	Text to show when a user search this music.

Sample Request Body:

```
{  
  "name": "miti di khusbu.mp3",  
  "userId": "19",  
  "location": "/data/log/",  
  "type": "AUDIO",  
  "description": "miti di khusbu"  
}
```

Response Body:

```
{  
  "status": "SUCCESS",  
  "responseCode": "00",  
}
```

```
"message": "Music added successfully",
"musicId": 6
}
```

Api-2 Update Music Status:

Functionality: Update music status in music app system.

Request

POST /music/updateStatus/

Headers

Content-Type application/json API works on application/json

Request Body

name	mandatory	Data type	Description
status	Y	String only	Status of user. Possible values: ACTIVE INACTIVE
userId	Y	Integer	User id of the user which owns this music.
musicId	Y	Integer	Music id which is to be updated.

Request Body:

```
{
  "status": "ACTIVE",
  "userId": "19",
  "musicId": "6"
}
```

Response Body:

```
{
  "userId": 19,
  "name": "miti di khusbu.mp3",
  "type": "AUDIO",
  "status": "ACTIVE",
  "location": "/data/log/",
  "description": "miti di khusbu"
}
```

Api-3 Get Music Data:

Functionality: Get music data which is already uploaded in system.

Request

GET /music/getData/

Request Param:

name	mandatory	Data type	Description
musicId	Y	Integer	Music id for which data is to be fetched.

Response Body:

```
{
  "status": "SUCCESS",
  "responseCode": "00",
  "message": "Success",
  "musicId": 6,
  "likesList": [
    {
      "id": 1,
      "userId": 19,
      "musicId": 6,
      "status": "ACTIVE"
    }
  ],
  "commentsList": [
    {
      "id": 1,
      "userId": "19",
      "data": "Mast song",
      "parentCommentId": "0",
      "musicId": "6",
      "status": "ACTIVE"
    },
    {
      "id": 2,
      "userId": "19",
      "data": "Mast song",
      "parentCommentId": "0",
      "musicId": "6",
      "status": "ACTIVE"
    }
  ]
}
```

Comments Apis:

Api-1 Add Comments:

Functionality: Add comments on music in music app system. Comments can be added by any user.

Request

POST */comments/add/*

Headers

Content-Type application/json API works on application/json

Request Body:

name	mandatory	Data type	Description
data	Y	String only(1000)	Comment data.
userId	Y	Integer	User id of the user which add comments on this music
musicId	Y	Integer	Music id on which comments is updated.
parentId	Y	Integer	Parent comment id.

Sample Request Body:

```
{
  "userId": "19",
  "data": "Mast song",
  "parentId": "2",
  "musicId": "6"
}
```

Response Body:

```
{
  "status": "SUCCESS",
  "responseCode": "00",
  "message": "Comment added successfully",
  "commentsId": "7"
}
```

Likes Apis:

Api-1 Add Likes:

Functionality: Add likes on music in music app system. One user can like only one time.

Request

POST */likes/add/*

Headers

Content-Type application/json API works on application/json

Request Body:

name	mandatory	Data type	Description
userId	Y	Integer	User id of the user which has liked this music.
musicId	Y	Integer	Music id on which likes is updated.

Sample Request Body:

```
{  
  "userId": "19",  
  "musicId": "10"  
}
```

Response Body:

```
{  
  "status": "SUCCESS",  
  "responseCode": "00",  
  "message": "Likes added successfully",  
  "likeId": 5  
}
```


Response Codes with there messages

Response message	Response code
INVALID_NAME	Err-0001
INVALID_CITY	Err-0002
INVALID_USER_TYPE	Err-0003
INVALID_STATE	Err-0004
INVALID_EMAIL	Err-0005
INVALID_COUNTRY	Err-0006
INVALID_CONTACT_NO	Err-0007
INVALID_DESCRIPTION	Err-0008
SOME_ERROR_OCCURRED	Err-0009
DUPLICATE_REQUEST	Err-0010
INVALID_USER_STATUS	Err-0011
INVALID_USER_ID	Err-0012
USER_NOT_FOUND	Err-0013
INVALID_LOCATION	Err-0014
INVALID_MUSIC_ID	Err-0015
INVALID_MUSIC_TYPE	Err-0016
INVALID_COMMENTS_DATA	Err-0017
FAILURE	Err-0018
INACTIVE_MUSIC	Err-0019
SUCCESS	Err-0020
internal server error	Err-0021
MUSIC_NOT_FOUND	Err-0022
UNAUTHORIZED_USER_ID.	Err-0023