

Music App API

Author: Mark Jordan

Course: OSU CS493 – Cloud Application Development

Instructors: N. Chaudhry, S. Hedao

Assignment: Portfolio Project

Due: 2022-12-04

Description: A RESTful web API that serves JSON metadata for a simulated music player web application.

App URL

<https://jordmark-music-app-api.wl.r.appspot.com>

Table of Contents

- [Music App API](#)
 - [App URL](#)
 - [Table of Contents](#)
 - [Models](#)
 - [Users](#)
 - [Tracks](#)
 - [Playlists](#)
 - [Endpoints](#)
 - [Users](#)
 - [GET All Users](#)
 - [Tracks](#)
 - [CREATE Track](#)
 - [GET All Tracks](#)
 - [GET Track by ID](#)
 - [PATCH Track by ID](#)
 - [PATCH All Tracks](#)
 - [UPDATE Track by ID](#)
 - [UPDATE All Tracks](#)
 - [DELETE Track by ID](#)
 - [DELETE All Tracks](#)
 - [Playlists](#)
 - [CREATE Playlist](#)
 - [GET Playlists](#)
 - [GET Playlist by ID](#)
 - [PATCH Playlist by ID](#)
 - [PATCH All Playlists](#)
 - [UPDATE Playlist by ID](#)

- [ADD a Track to a Playlist](#)
- [UPDATE All Playlists](#)
- [DELETE Playlist by ID](#)
- [DELETE a Track from a Playlist](#)
- [DELETE All Playlists](#)
- [Partial Credit Explanations](#)
 - [Pagination](#)
 - [Wrong Error Codes](#)

Models

Users

A 'user' represents a music app user. When a 'user' signs into the music app, the 'user's' will be stored in the database.

Properties	Data Type	Required?	Description
name	String	Yes	The user's name associated with their Google account.
email	String	Yes	The user's email address associated with their Google account.
sub	String	Yes	The value of the 'sub' property encoded in the user's OAuth2.0 jwt token.

Tracks

A 'track' is a digital music record available on the music app.

This entity **IS NOT** a protected resource. In other words, the entity is not owned by any 'user'; any music app user can create, read, update and delete a 'track'. A 'track' can be found in more than one 'playlist' but any standalone 'playlist' cannot contain duplicate 'tracks'. A 'track' does not have to be in any 'playlist'.

Properties	Data Type	Required?	Description
album	String	Yes	The album on which the track was released. If the track was not released as part of an album, the value of the property should be set to 'NULL'.
artists	Array of Strings	Yes	A track can have more than one artist credited.
duration_s	Integer	Yes	The track length in seconds.
name	String	Yes	The name of the track.

Properties	Data Type	Required?	Description
playlists	Array of Objects	No	The playlists that contain this track. A track does not have to be in any playlists.

Playlists

A 'playlist' is a collection of 'tracks' created and managed by a music app 'user'.

This entity **IS** a protected resource. A user must be authenticated and authorized in order to create, read, update and delete a 'playlist'.

Properties	Data Type	Required?	Description
name	String	Yes	The name of the playlist.
public	Boolean	Yes	The playlist's public/private status.
owner_id	String	Yes	The user who owns the playlist.
tracks	Array of Objects	No	A collection of tracks in the playlist.

Endpoints

Users

GET All Users

Retrieves a collection of all users in the database.

Request

```
GET /users
```

```
Authorization: none
```

```
Parameters: none
```

```
Request Body: none
```

Responses

```
200 - OK
```

```
[  
  {  
    "email": "jordmark@fakeemail.edu",  
    "name": "Mark Jordan",  
    "sub": "107882060863705507732",
```

```
    "self_url": "https://jordmark-music-app-  
api.wl.r.appspot.com/users/5153611168350208",  
    "id": "5153611168350208"  
  },  
  {  
    "self_url": "https://jordmark-music-app-  
api.wl.r.appspot.com/users/5729450050191360",  
    "email": "markyjordan@fakeemail.com",  
    "name": "Marky",  
    "sub": "108620139713685040732",  
    "id": "5729450050191360"  
  }  
]
```

406 – NOT ACCEPTABLE

```
{  
  "Error": "The server only produces responses that conform to an  
'application/json' value provided in the request 'Accept' header."  
}
```

Tracks

Tracks are not a user's protected resources. The Tracks endpoints serve to manage tracks on the music web application, and anyone can create, read, update and delete a track.

CREATE Track

Add a new track to the music application database.

Request

POST /tracks

Authorization: none

Parameters: none

Request Body: required

Request Body Format: JSON

Request Body Attributes:

- album:
 - type: string
 - required: yes
 - description: the name of album containing the track
- artists:
 - type: array of strings
 - required: yes
 - description: the track's artists
- duration_s:

- type: integer
- required: yes
- description: the track duration in seconds
- name:
 - type: string
 - required: yes
 - description: the track name

Sample Request

```
{
  "album": "null",
  "artists": [ "IU" ],
  "duration_s": 167,
  "name": "eight"
}
```

Responses

200 – OK

```
{
  "id": "5675594515742720",
  "album": "null",
  "artists": [
    "IU"
  ],
  "duration_s": 167,
  "name": "eight",
  "playlists": [],
  "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/tracks/5675594515742720"
}
```

400 – BAD REQUEST

```
{
  "Error": "The request object is missing one or more of the required
attributes."
}
```

406 – NOT ACCEPTABLE

```
{
```

```
"Error": "The server only produces responses that conform to an  
'application/json' value provided in the request 'Accept' header."  
}
```

500 – INTERNAL SERVER ERROR

```
{  
  "Error": "An unexpected server error occurred."  
}
```

GET All Tracks

Retrieves a collection of all tracks from the database.

Request

GET /tracks

Authorization: none

Parameters: none

Request Body: none

Responses

200 – OK

```
[  
  {  
    "duration_s": 210,  
    "album": "The ReVe Festival 2022 – Feel My Rhythm",  
    "name": "Feel My Rhythm",  
    "playlists": [],  
    "self_url": "https://jordmark-music-app-  
api.wl.r.appspot.com/tracks/4872136191639552",  
    "artists": [  
      "Red Velvet"  
    ],  
    "id": "4872136191639552"  
  },  
  {  
    "playlists": [],  
    "duration_s": 177,  
    "artists": [  
      "ITZY"  
    ],  
    "self_url": "https://jordmark-music-app-  
api.wl.r.appspot.com/tracks/5192424284487680",  
  }  
]
```

```

    "name": "Not Shy",
    "album": "Not Shy",
    "id": "5192424284487680"
  },
  {
    "artists": [
      "Twice"
    ],
    "duration_s": 177,
    "album": "Between 1&2",
    "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/tracks/5643550469390336",
    "playlists": [],
    "name": "Talk that Talk",
    "id": "5643550469390336"
  },
  {
    "playlists": [],
    "artists": [
      "IU",
      "SUGA"
    ],
    "album": "null",
    "duration_s": 167,
    "name": "eight",
    "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/tracks/5645760532054016",
    "id": "5645760532054016"
  },
  {
    "name": "After LIKE",
    "duration_s": 176,
    "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/tracks/5708765252812800",
    "playlists": [],
    "artists": [
      "IVE"
    ],
    "album": "After LIKE",
    "id": "5708765252812800"
  },
  {
    "album": "IM NAYEON",
    "playlists": [],
    "duration_s": 166,
    "name": "Pop!",
    "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/tracks/5750704903815168",
    "artists": [
      "NAYEON"
    ],
    "id": "5750704903815168"
  },
  {

```

```
    "self_url": "https://jordmark-music-app-  
api.wl.r.appspot.com/tracks/6279511075192832",  
    "duration_s": 212,  
    "playlists": [],  
    "name": "WHISTLE",  
    "artists": [  
        "BLACKPINK"  
    ],  
    "album": "SQUARE ONE",  
    "id": "6279511075192832"  
  }  
]
```

406 – NOT ACCEPTABLE

```
{  
  "Error": "The server only produces responses that conform to an  
'application/json' value provided in the request 'Accept' header."  
}
```

500 – INTERNAL SERVER ERROR

```
{  
  "Error": "An unexpected server error occurred."  
}
```

GET Track by ID

Retrieve a single track from the database.

Request

```
GET /tracks/:track_id  
  
Authorization: none  
Parameters:  
- track_id:  
  - type: string  
  - required: yes  
  - description: the id of the track  
Request Body: none
```

Responses

200 – OK

```
{
  "name": "eight",
  "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/tracks/6232102555090944",
  "album": "null",
  "artists": [
    "IU"
  ],
  "playlists": [],
  "duration_s": 167,
  "id": "6232102555090944"
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

406 – NOT ACCEPTABLE

```
{
  "Error": "The server only produces responses that conform to an
'application/json' value provided in the request 'Accept' header."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

PATCH Track by ID

Update a single track in the database. This endpoint doesn't allow you to update the track's 'playlists' property.

Request

PATCH /tracks/:track_id

```
Authorization: none
Parameters:
- track_id:
  - type: string
  - required: yes
  - description: the id of the track to patch
Request Body: JSON
Request Body Attributes:
- album:
  - type: string
  - required: no
  - description: the name of album containing the track
- artists:
  - type: array of strings
  - required: no
  - description: the track's artists
- duration_s:
  - type: integer
  - required: no
  - description: the track duration in seconds
- name:
  - type: string
  - required: no
  - description: the track name
```

Sample Request

```
Updating 'artists' Property

{
  "artists": [ "IU", "SUGA" ]
}
```

Responses

```
200 - OK

{
  "id": "6232102555090944",
  "album": "null",
  "artists": [
    "IU",
    "SUGA"
  ],
  "duration_s": 167,
  "name": "eight",
  "playlists": [],
  "self_url": "https://jordmark-music-app-
```

```
api.wl.r.appspot.com/tracks/6232102555090944"
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

406 – NOT ACCEPTABLE

```
{
  "Error": "The server only produces responses that conform to an
'application/json' value provided in the request 'Accept' header."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

PATCH All Tracks

This endpoint isn't supported.

Request

```
PATCH /tracks
```

Response

405 – NOT ALLOWED

```
{
  "Error": "The request method is not allowed for the endpoint."
}
```

UPDATE Track by ID

Update a single track in the database. This endpoint doesn't allow you to update the track's 'playlists' property.

Request

```
PUT /tracks/:track_id
```

Authorization: none

Parameters:

- track_id:
 - type: string
 - required: yes
 - description: the id of the track to update

Request Body: required

Request Body Format: JSON

Request Body Attributes:

- album:
 - type: string
 - required: yes
 - description: the name of album containing the track
- artists:
 - type: array of strings
 - required: yes
 - description: the track's artists
- duration_s:
 - type: integer
 - required: yes
 - description: the track duration in seconds
- name:
 - type: string
 - required: yes
 - description: the track name

Sample Request

```
{
  "album": "IU 5th Album 'Lilac'",
  "artists": [ "IU" ],
  "duration_s": 214,
  "name": "Lilac"
}
```

Responses

200 – OK

```
{
  "id": "5735810494103552",
```

```
"album": "IU 5th Album 'Lilac'",
"artists": [
  "IU"
],
"duration_s": 214,
"name": "Lilac",
"playlists": [],
"self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/tracks/5735810494103552"
}
```

400 – BAD REQUEST

```
{
  "Error": "The request object is missing one or more of the required
attributes."
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

406 – NOT ACCEPTABLE

```
{
  "Error": "The server only produces responses that conform to an
'application/json' value provided in the request 'Accept' header."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

UPDATE All Tracks

This endpoint isn't supported.

Request

```
PUT /tracks
```

Response

405 – NOT ALLOWED

```
{
  "Error": "The request method is not allowed for the endpoint."
}
```

DELETE Track by ID

Delete a single track from the database.

Request

```
DELETE /tracks/:track_id
```

Authorization: none

Parameters:

- track_id:
 - type: string
 - required: yes
 - description: the id of the track to delete

Request Body: none

Responses

204 – NO CONTENT

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

DELETE All Tracks

This endpoint isn't supported.

Request

```
DELETE /tracks
```

Response

```
405 - NOT ALLOWED
```

```
{
  "Error": "The request method is not allowed for the endpoint."
}
```

Playlists

Playlists are a user's protected resources, and as such, a user must be authorized to create, read, update, or delete a playlist.

CREATE Playlist

Add a playlist to the database.

Request

```
POST /playlists
```

```
Authorization: bearer token
```

```
Parameters: none
```

```
Request Body: required
```

```
Request Body Format: JSON
```

```
Request Body Attributes:
```

- name:
 - type: string
 - required: yes
 - description: the name of the playlist
- public:
 - type: boolean
 - required: yes
 - description: the public/private status of the playlist

Sample Request

```
{
  "name": "user 1 playlist 1",
  "public": true
}
```

Responses

201 – CREATED

```
{
  "id": "5643280054222848",
  "name": "user 1 playlist 1",
  "public": true,
  "owner_id": "108620139713685040945",
  "tracks": [],
  "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/playlists/5643280054222848"
}
```

400 – BAD REQUEST

```
{
  "Error": "The request object is missing one or more of the required
attributes."
}
```

401 – UNAUTHORIZED

```
{
  "Error": "The request is missing a valid auth token."
}
```

406 – NOT ACCEPTABLE

```
{
  "Error": "The server only produces responses that conform to an
'application/json' value provided in the request 'Accept' header."
}
```

500 – INTERNAL SERVER ERROR


```
{
  "Error": "An unexpected server error occurred."
}
```

GET Playlists

Retrieves a collection of a user's playlists from the database.

Request

GET /playlists

Authorization: bearer token

Parameters: none

Request Body: none

Responses

200 - OK

```
[
  {
    "name": "user 1 playlist 3",
    "tracks": [],
    "self_url": "https://jordmark-music-app-api.wl.r.appspot.com/playlists/5070428590571520",
    "owner_id": "108620139713685040945",
    "public": true,
    "id": "5070428590571520"
  },
  {
    "public": true,
    "tracks": [],
    "name": "user 1 playlist 2",
    "owner_id": "108620139713685040945",
    "self_url": "https://jordmark-music-app-api.wl.r.appspot.com/playlists/5080330100801536",
    "id": "5080330100801536"
  },
  {
    "tracks": [],
    "self_url": "https://jordmark-music-app-api.wl.r.appspot.com/playlists/5633378543992832",
    "public": true,
    "name": "user 1 playlist 1",
    "owner_id": "108620139713685040945",
    "id": "5633378543992832"
  },
  {
    "name": "user 1 playlist 6",
```

```
    "tracks": [],
    "public": true,
    "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/playlists/5640825748848640",
    "owner_id": "108620139713685040945",
    "id": "5640825748848640"
  },
  {
    "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/playlists/5671636300726272",
    "owner_id": "108620139713685040945",
    "tracks": [],
    "public": true,
    "name": "user 1 playlist 5",
    "id": "5671636300726272"
  }
]
```

401 – UNAUTHORIZED

```
{
  "Error": "The request is missing a valid auth token."
}
```

406 – NOT ACCEPTABLE

```
{
  "Error": "The server only produces responses that conform to an
'application/json' value provided in the request 'Accept' header."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

GET Playlist by ID

Retrieve a user's playlist from the database.

Request

```
GET /playlists/:playlist_id
```

Authorization: bearer token

Parameters:

- playlist_id:
 - type: string
 - required: yes
 - description: the id of the playlist to retrieve

Request Body: none

Responses

200 – OK

```
{
  "name": "user 1 playlist 1",
  "tracks": [],
  "public": true,
  "owner_id": "108620139713685040945",
  "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/playlists/5658646574792704",
  "id": "5658646574792704"
}
```

401 – UNAUTHORIZED

```
{
  "Error": "The request is missing a valid auth token."
}
```

403 – FORBIDDEN

```
{
  "Error": "The user is not authorized to access the requested
resource."
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

406 – NOT ACCEPTABLE

```
{
  "Error": "The server only produces responses that conform to an
'application/json' value provided in the request 'Accept' header."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

PATCH Playlist by ID

Update a single playlist in the database. This endpoint doesn't allow you to update the playlist's 'tracks' property.

Request

PATCH /playlists/:playlist_id

Authorization: bearer token

Parameters:

- playlist_id:
 - type: string
 - required: yes
 - description: the id of the playlist to patch

Request Body: required

Request Body Format: JSON

Request Body Attributes:

- name:
 - type: string
 - required: no
 - description: the name of the playlist
- public:
 - type: boolean
 - required: no
 - description: the public/private status of the playlist

Sample Request

```
{
  "public": false
}
```

Responses

200 – OK

```
{
  "id": "5658646574792704",
  "name": "patched user 1 playlist 1",
  "public": true,
  "owner_id": "108620139713685040945",
  "tracks": [],
  "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/playlists/5658646574792704"
}
```

401 – UNAUTHORIZED

```
{
  "Error": "The request is missing a valid auth token."
}
```

403 – FORBIDDEN

```
{
  "Error": "The user is not authorized to access the requested
resource."
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

406 – NOT ACCEPTABLE

```
{
  "Error": "The server only produces responses that conform to an
'application/json' value provided in the request 'Accept' header."
}
```

```
500 - INTERNAL SERVER ERROR
```

```
{
  "Error": "An unexpected server error occurred."
}
```

PATCH All Playlists

This endpoint isn't supported.

Request

```
PATCH /playlists
```

Response

```
405 - NOT ALLOWED
```

```
{
  "Error": "The request method is not allowed for the endpoint."
}
```

UPDATE Playlist by ID

Update a single playlist in the database. This endpoint doesn't allow you to update the playlist's 'tracks' property.

Request

```
PUT /playlists/:playlist_id
```

Authorization: bearer token

Parameters:

- playlist_id
 - type: string
 - required: yes
 - description: the id of the playlist to update

Request Body: required

Request Body Format: JSON

Request Body Attributes:

- name:
 - type: string
 - required: yes
 - description: the name of the playlist
- public:
 - type: boolean

- required: yes
- description: the public/private status of the playlist

Sample Request

```
{
  "name": "updated user 1 playlist 1",
  "public": false
}
```

Responses

200 – OK

```
{
  "id": "5681414901071872",
  "name": "updated user 1 playlist 1",
  "public": false,
  "owner_id": "108620139713685040945",
  "tracks": [],
  "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/playlists/5681414901071872"
}
```

400 – BAD REQUEST

```
{
  "Error": "The request object is missing one or more of the required
attributes."
}
```

401 – UNAUTHORIZED

```
{
  "Error": "The request is missing a valid auth token."
}
```

403 – FORBIDDEN

```
{
  "Error": "The user is not authorized to access the requested
resource."
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

406 – NOT ACCEPTABLE

```
{
  "Error": "The server only produces responses that conform to an
'application/json' value provided in the request 'Accept' header."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

ADD a Track to a Playlist

Add's a track to a user's playlist. The playlist will contain the track's id in its 'tracks' property array. This endpoint simultaneously updates the 'playlists' property of the track that was added to the playlist.

Request

PUT /playlists/:playlist_id/tracks/:track_id

Authorization: bearer token

Parameters:

- playlist_id
 - type: string
 - required: yes
 - description: the id of the playlist to which the track will be added
- track_id
 - type: string
 - required: yes
 - description: the id of the track to be added to the user's playlist

Request Body: None

Responses

200 – OK

```
{
  "id": "5669152601669632",
  "name": "track added to user 1 playlist 1",
  "public": true,
  "owner_id": "108620139713685040945",
  "tracks": [
    "5712837116690432"
  ],
  "self_url": "https://jordmark-music-app-
api.wl.r.appspot.com/playlists/5669152601669632"
}
```

400 – BAD REQUEST

```
{
  "Error": "Duplicate tracks are not allowed."
}
```

401 – UNAUTHORIZED

```
{
  "Error": "The request is missing a valid auth token."
}
```

403 – FORBIDDEN

```
{
  "Error": "The user is not authorized to access the requested
resource."
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

406 – NOT ACCEPTABLE

```
{
```

```
"Error": "The server only produces responses that conform to an  
'application/json' value provided in the request 'Accept' header."  
}
```

UPDATE All Playlists

This endpoint isn't supported.

Request

```
PUT /playlists
```

Response

405 – NOT ALLOWED

```
{  
  "Error": "The request method is not allowed for the endpoint."  
}
```

DELETE Playlist by ID

Deletes a user's playlist from the database. This endpoint simultaneously updates the 'playlists' property of each track that was in the playlist.

Request

```
DELETE /playlists/:playlist_id
```

Authorization: bearer token

Parameters:

- playlist_id
 - type: string
 - required: yes
 - description: the id of the playlist to delete

Request Body: None

Responses

204 – NO CONTENT

401 – UNAUTHORIZED

```
{
  "Error": "The request is missing a valid auth token."
}
```

403 – FORBIDDEN

```
{
  "Error": "The user is not authorized to access the requested resource."
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

DELETE a Track from a Playlist

Removes a track from a user's playlist. This endpoint simultaneously updates the 'playlists' property of the track that was in the playlist.

Request

```
DELETE /playlists/:playlist_id/tracks/:track_id
```

Authorization: bearer token

Parameters:

- playlist_id:
 - type: string
 - required: yes
 - description: the id of the playlist from which to remove the given track
- track_id
 - type: string

```
- required: yes
- description: the id of the track to remove from the playlist
Request Body: None
```

Responses

204 – NO CONTENT

401 – UNAUTHORIZED

```
{
  "Error": "The request is missing a valid auth token."
}
```

403 – FORBIDDEN

```
{
  "Error": "The user is not authorized to access the requested
resource."
}
```

404 – NOT FOUND

```
{
  "Error": "A resource with the requested id could not be found."
}
```

500 – INTERNAL SERVER ERROR

```
{
  "Error": "An unexpected server error occurred."
}
```

DELETE All Playlists

This endpoint is not supported.

Request

DELETE /playlists

Response

```
405 - NOT ALLOWED
```

```
{  
  "Error": "The request method is not allowed for the endpoint."  
}
```

Partial Credit Explanations

Pagination

I intended to meet this requirement by adding paging to the 'playlists' and 'tracks' collection entities.

I tried looking at the cursor paging tutorial from the google cloud docs but I found the code to be too incomplete for my experience level. Here's the code:

<https://cloud.google.com/datastore/docs/samples/datastore-cursor-paging>. I also tried constructing a custom middleware to handle a query using an offset and limit to retrieve a maximum of five entities but I couldn't get that to work either. I commented out the code I tried implementing in the 'tracksModel' file.

I kept getting an internal server error so I suspect something about the way I was locating and setting the cursor was incorrect, but I couldn't diagnose the problem using console logging.

Wrong Error Codes

Sometimes, I experienced issues with receiving the wrong error codes for requests that made use of one of the track or playlists environment variables that were dynamically updated by Postman. I troubleshooted these issues by clearing the datastore and sending the corresponding POST or PUT request to reupdate the track or playlist environment variables and the issue was resolved.

[Return to the Top](#)