Back End Boot Camp Final Project

Music Library

By Sabrina Backlin

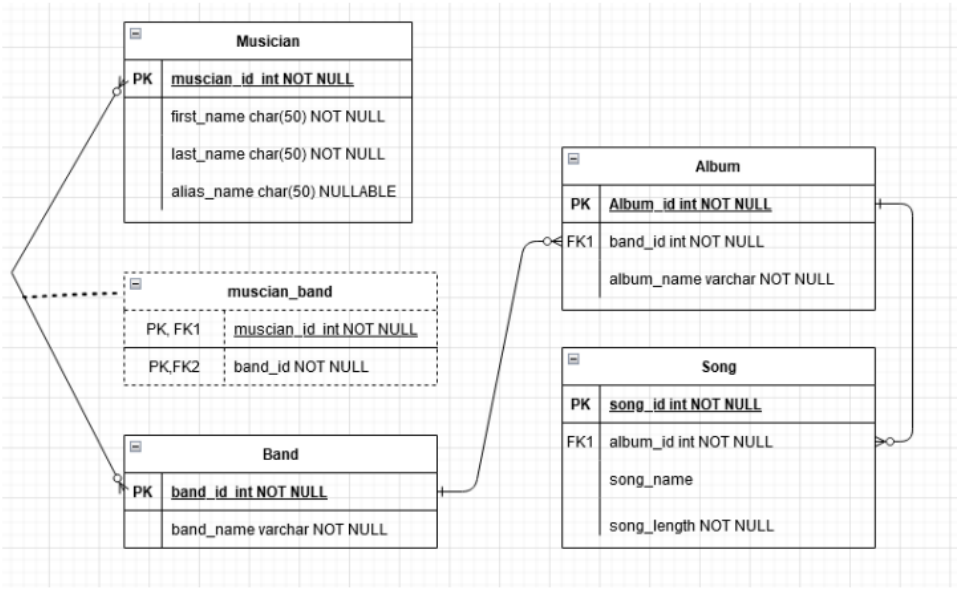
Overview:

I created a Music Library to hold musicians, bands, albums, and songs. My many to many relationship was Musician_Band. A musician might be in more than one band, and a band contains many musicians. My one to many relationships included Band to Album, and Album to Song.

You can find my code at: https://github.com/DGolf1313/FinalMusicLibrary

I started my project by setting out my tables in Draw.io

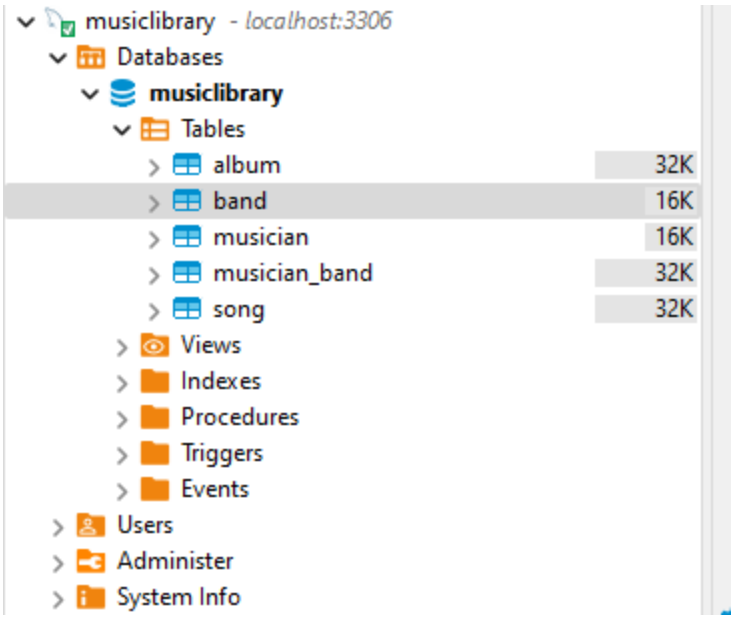
Here is my ERD:



I then created my database schema:
https://github.com/DGolf1313/FinalMusicLibrary/blob/main/src/test/resources/flyway/migrations/MusicLibrarySchema.sql

```sql
DROP TABLE IF EXISTS song;
DROP TABLE IF EXISTS album;
DROP TABLE IF EXISTS musician_band;
DROP TABLE IF EXISTS band;
DROP TABLE IF EXISTS musician;

CREATE TABLE musician (
    musician_id INT AUTO_INCREMENT NOT NULL,
    first_name CHAR(50) NOT NULL,
    last_name CHAR(50) NOT NULL,
    alias_name CHAR(50),
    PRIMARY KEY (musician_id)
);

CREATE TABLE band (
    band_id INT AUTO_INCREMENT NOT NULL,
    band_name VARCHAR(150) NOT NULL,
    PRIMARY KEY (band_id)
);

CREATE TABLE musician_band (
    musician_id INT NOT NULL,
    band_id INT NOT NULL,
    FOREIGN KEY (musician_id) REFERENCES musician (musician_id) ON DELETE CASCADE,
    FOREIGN KEY (band_id) REFERENCES band (band_id) ON DELETE CASCADE,
    UNIQUE KEY (musician_id, band_id)
);

CREATE TABLE album (
    album_id INT AUTO_INCREMENT NOT NULL,
    band_id INT NOT NULL,
    album_name VARCHAR(1000) NOT NULL,
    PRIMARY KEY (album_id),
    FOREIGN KEY (band_id) REFERENCES band (band_id) ON DELETE CASCADE
);

CREATE TABLE song (
    song_id INT AUTO_INCREMENT NOT NULL,
    album_id INT NOT NULL,
    song_name VARCHAR(1000) NOT NULL,
    song_length TIME NOT NULL,
    PRIMARY KEY (song_id),
    FOREIGN KEY (album_id) REFERENCES album (album_id) ON DELETE CASCADE
);
```

Here is my database showing the created tables:

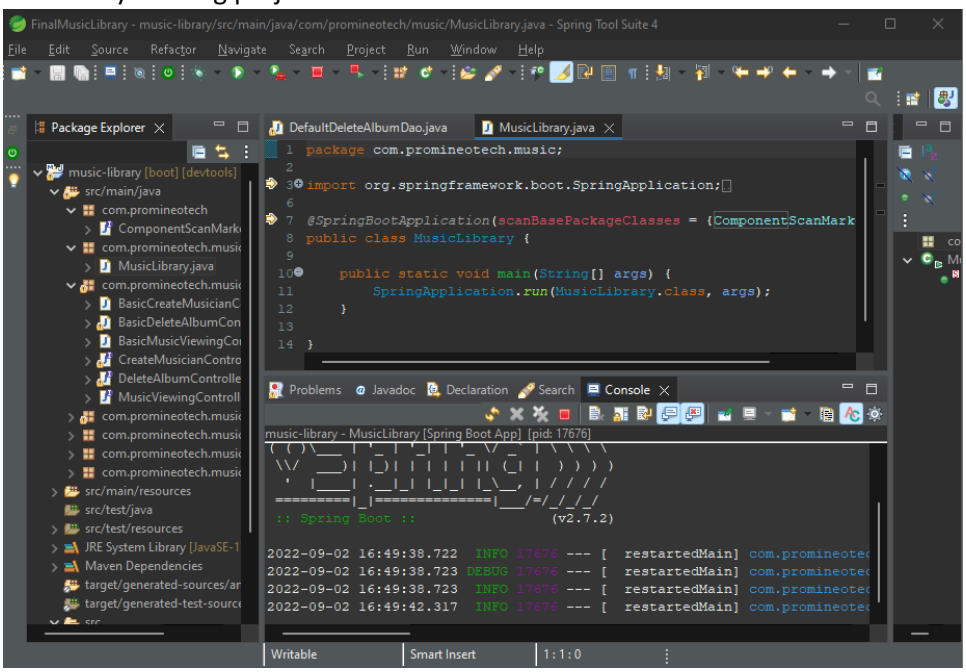


Next I wrote insert statements to fill my database. I choose three albums from three different bands. Two bands contained some of the same artist. Temple of the Dog, and Soundgarden. https://github.com/DGolf1313/FinalMusicLibrary/blob/main/src/test/resources/flyway/migrations/MusicLibraryData.sql

MusicLibraryData.sql | new 2

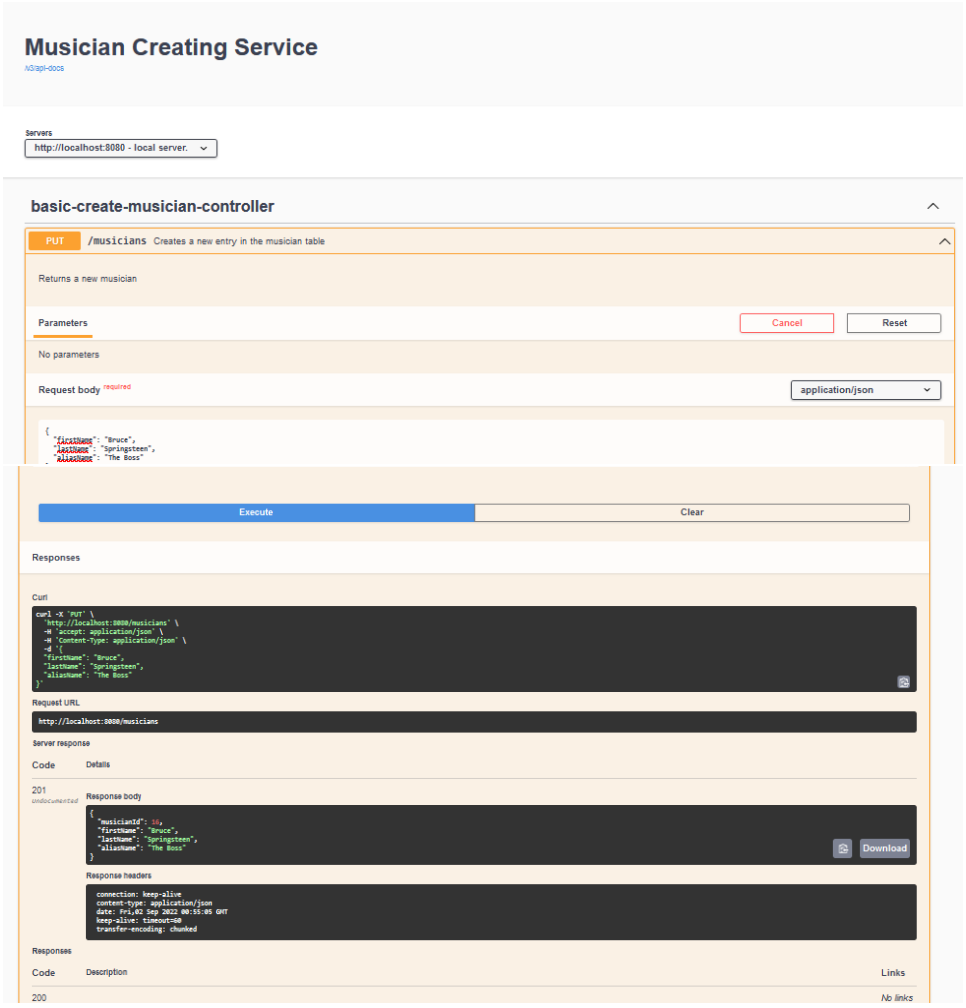
```sql
INSERT INTO musician  (first_name, last_name, alias_name) VALUES ('Dave', 'Hosking', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Killian', 'Gavin', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Tim', 'Hart', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Jon', 'Hart', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Dave', 'Symes', NULL);

INSERT INTO band (band_name) VALUES ('Boy & Bear');

INSERT INTO musician_band (musician_id, band_id) VALUES (1, 1);
INSERT INTO musician_band (musician_id, band_id) VALUES (2, 1);
INSERT INTO musician_band (musician_id, band_id) VALUES (3, 1);
INSERT INTO musician_band (musician_id, band_id) VALUES (4, 1);
INSERT INTO musician_band (musician_id, band_id) VALUES (5, 1);

INSERT INTO album (band_id, album_name) VALUES (1, 'Harlequin Dream');

INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Southern Sun', '00:04:41');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Old TownBlues', '00:03:24');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Harlequin Dream', '00:04:20');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Three Headed Woman', '00:04:03');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Bridges', '00:04:06');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'A Moments Grace', '00:04:53');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'End of the Line', '00:03:47');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Back Down the Black', '00:06:47');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Real Estate', '00:02:57');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Stranger', '00:04:57');
INSERT INTO song (album_id, song_name, song_length) VALUES(1, 'Arrow Flight', '00:04:22');



INSERT INTO musician  (first_name, last_name, alias_name) VALUES ('Chris', 'Cornell', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Jeff', 'Ament', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Matt', 'Cameron', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Stone', 'Gossard', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Mike', 'McCready', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Eddy', 'Vedder', NULL);

INSERT INTO band (band_name) VALUES ('Temple of the Dog');

INSERT INTO musician_band (musician_id, band_id) VALUES (6, 2);
INSERT INTO musician_band (musician_id, band_id) VALUES (7, 2);
INSERT INTO musician_band (musician_id, band_id) VALUES (8, 2);
INSERT INTO musician_band (musician_id, band_id) VALUES (9, 2);
INSERT INTO musician_band (musician_id, band_id) VALUES (10, 2);
INSERT INTO musician_band (musician_id, band_id) VALUES (11, 2);

INSERT INTO album (band_id, album_name) VALUES (2, 'Temple of the Dog');

INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Say Hello To Heaven', '00:06:24');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Reach Down', '00:11:12');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Hunger Strike', '00:04:06');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Pushin Forward Back', '00:03:45');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Call Me A Dog', '00:05:04');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Times Of Trouble', '00:05:42');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Wooden Jesus', '00:04:10');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Your Savior', '00:04:04');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'Four Walled World', '00:06:54');
INSERT INTO song (album_id, song_name, song_length) VALUES(2, 'All Night Thing', '00:03:51');



INSERT INTO musician  (first_name, last_name, alias_name) VALUES ('Kim', 'Thayil', NULL);
INSERT INTO musician (first_name, last_name, alias_name) VALUES ('Ben', 'Shepherd', NULL);


INSERT INTO band (band_name) VALUES ('Soundgarden');

INSERT INTO musician_band (musician_id, band_id) VALUES (6, 3);
INSERT INTO musician_band (musician_id, band_id) VALUES (12, 3);
INSERT INTO musician_band (musician_id, band_id) VALUES (8, 3);
INSERT INTO musician_band (musician_id, band_id) VALUES (13, 3);


INSERT INTO album (band_id, album_name) VALUES (3, 'Sumperunknown');

INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Let Me Drown', '00:03:51');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'My Wave', '00:05:12');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Fell on Black Days', '00:04:42');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Mailman', '00:04:25');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Superunknown', '00:05:06');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Head Down', '00:06:08');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Black Hole Sun', '00:05:18');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Spoonman', '00:04:06');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Limo Wreck', '00:05:47');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'The Day I Tried to Live', '00:05:19');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Kickstand', '00:01:34');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Fresh Tendrils', '00:04:16');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Fourth of July', '00:05:08');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Half', '00:02:14');
INSERT INTO song (album_id, song_name, song_length) VALUES(3, 'Like Suicide', '00:07:01');

```

After creating a working database I created my Mavnen project, added the dependencies, and connected to the database.
Here is my running project:



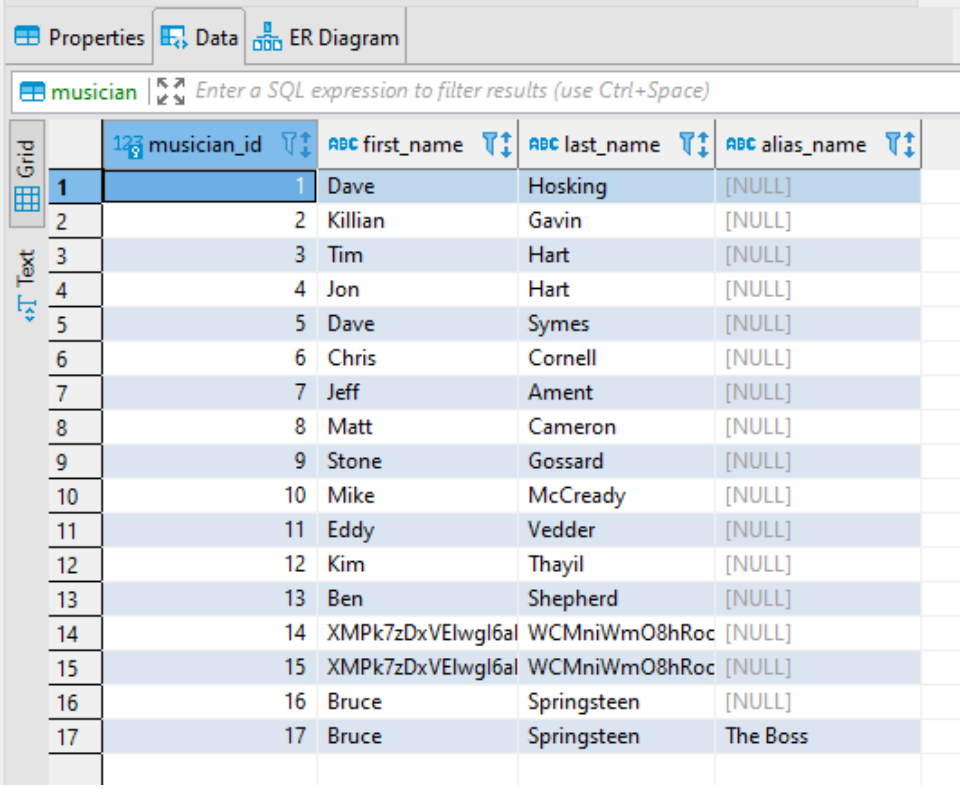
My Operations:

CREATE or POST: Add a new musician
I added the musician Bruce Springsteen, with the alias “The Boss”
Swagger:



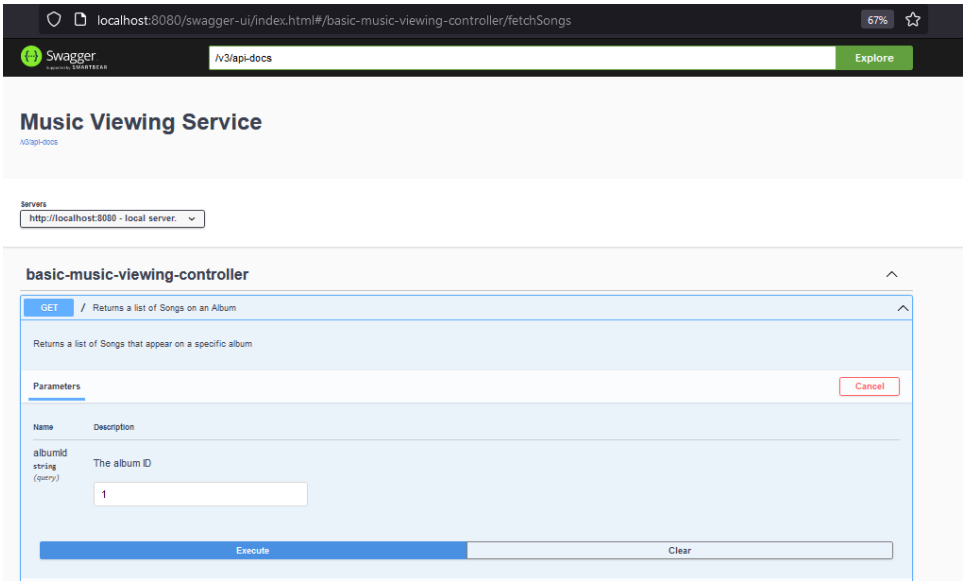
The added musician in the database: (*note, you can see that there are several Bruce Springsteen entries. The first time the alias didn't register, I found a syntax error that was causing the alias_name to

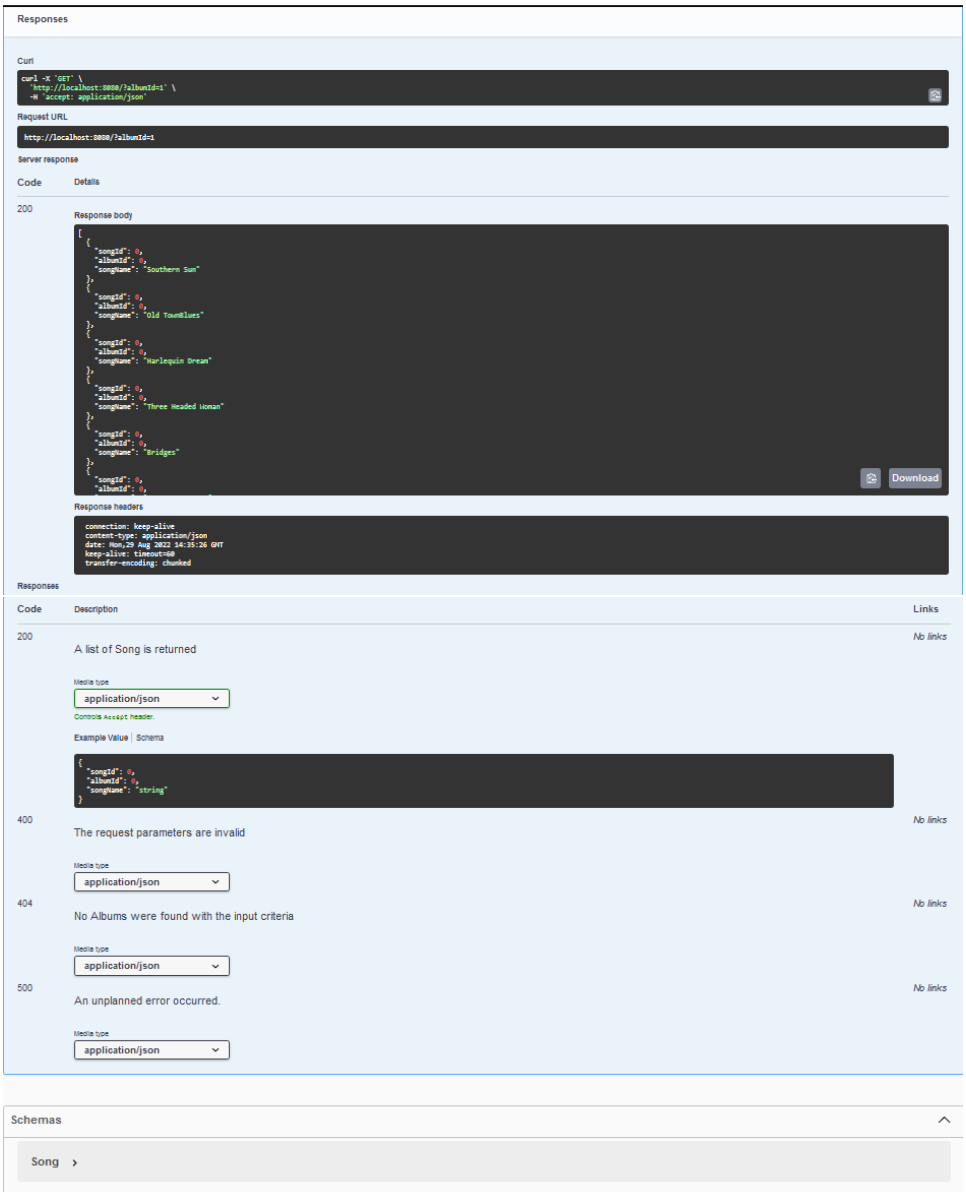
not enter. Once fixed you can see that it is now loading properly.



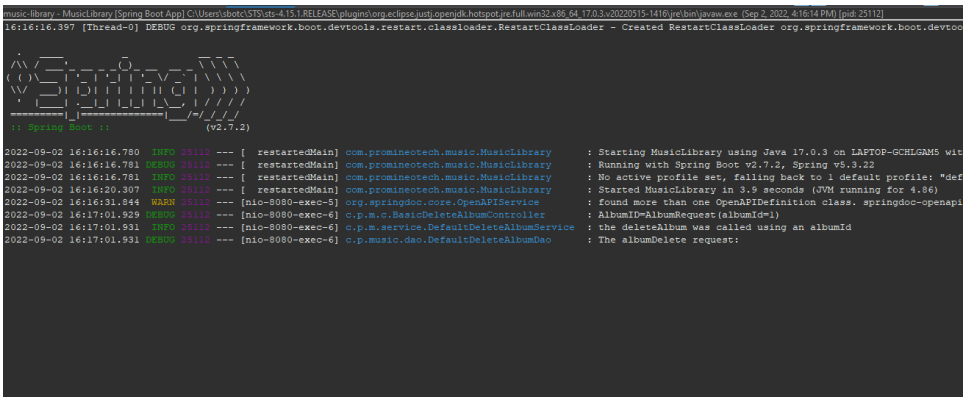
	musician_id	first_name	last_name	alias_name
1	1	Dave	Hosking	[NULL]
2	2	Killian	Gavin	[NULL]
3	3	Tim	Hart	[NULL]
4	4	Jon	Hart	[NULL]
5	5	Dave	Symes	[NULL]
6	6	Chris	Cornell	[NULL]
7	7	Jeff	Ament	[NULL]
8	8	Matt	Cameron	[NULL]
9	9	Stone	Gossard	[NULL]
10	10	Mike	McCready	[NULL]
11	11	Eddy	Vedder	[NULL]
12	12	Kim	Thayil	[NULL]
13	13	Ben	Shepherd	[NULL]
14	14	XMPk7zDxVElwgl6al	WCMniWmO8hRoc	[NULL]
15	15	XMPk7zDxVElwgl6al	WCMniWmO8hRoc	[NULL]
16	16	Bruce	Springsteen	[NULL]
17	17	Bruce	Springsteen	The Boss

READ or GET: View a list of songs on an album: Running on Swagger

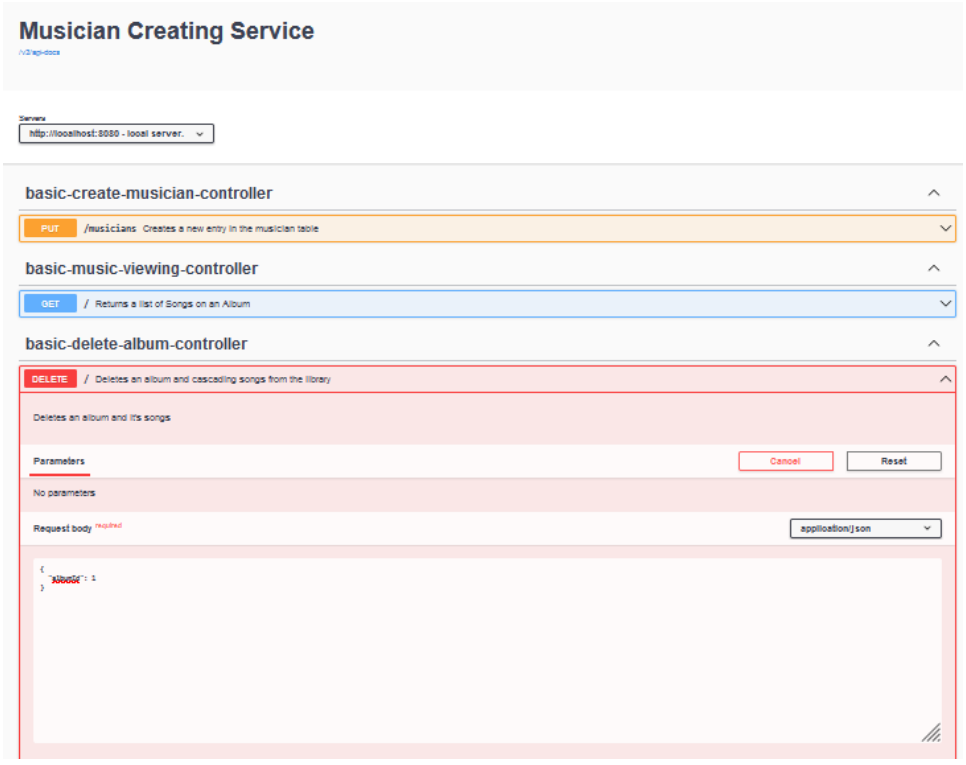


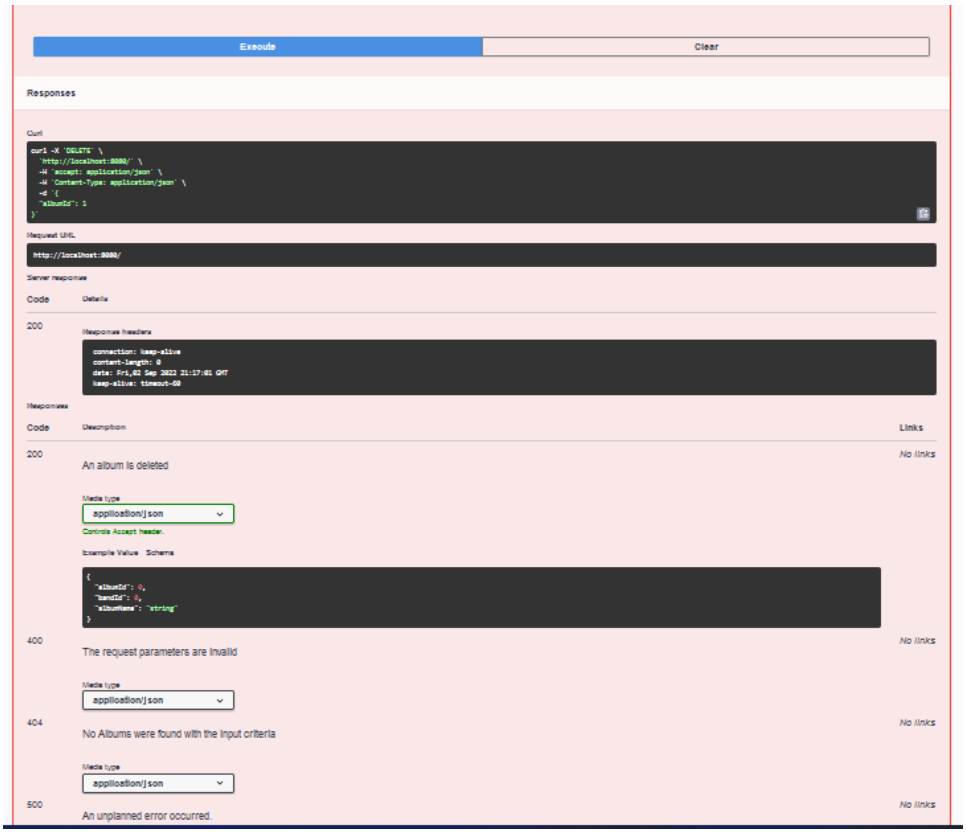


DELETE or DELETE: Delete an album from the Album Table by it's Album_Id:
The project running after the request:

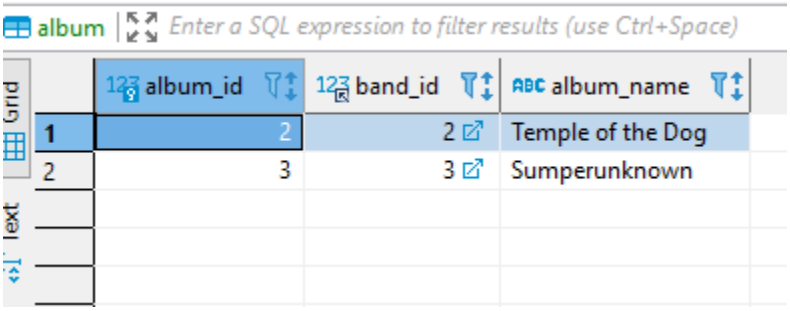

music-library - MusicLibrary [Spring Boot App] C:\Users\sbotc\STS\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Sep 2, 2022, 4:16:14 PM) [pid: 25112]
16:16:16.397 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader org.springframework.boot.devtoo
:: Spring Boot ::
(v2.7.2)
2022-09-02 16:16:16.780 INFO 25112 --- [restartedMain] com.promineotech.music.MusicLibrary : Starting MusicLibrary using Java 17.0.3 on LAPTOP-GCHLGAM5 wit
2022-09-02 16:16:16.781 DEBUG 25112 --- [restartedMain] com.promineotech.music.MusicLibrary : Running with Spring Boot v2.7.2, Spring v5.3.22
2022-09-02 16:16:16.781 INFO 25112 --- [restartedMain] com.promineotech.music.MusicLibrary : No active profile set, falling back to 1 default profile: "def
2022-09-02 16:16:20.307 INFO 25112 --- [restartedMain] com.promineotech.music.MusicLibrary : Started MusicLibrary in 3.9 seconds (JVM running for 4.86)
2022-09-02 16:16:31.844 WARN 25112 --- [nio-8080-exec-5] org.springdoc.core.OpenAPIService : found more than one OpenAPIDefinition class. springdoc-openapi
2022-09-02 16:17:01.929 DEBUG 25112 --- [nio-8080-exec-6] c.p.m.c.BasicDeleteAlbumController : AlbumID=AlbumRequest(albumId=1)
2022-09-02 16:17:01.931 INFO 25112 --- [nio-8080-exec-6] c.p.m.service.DefaultDeleteAlbumService : the deleteAlbum was called using an albumId
2022-09-02 16:17:01.931 DEBUG 25112 --- [nio-8080-exec-6] c.p.music.dao.DefaultDeleteAlbumDao : The albumDelete request:

Swagger:


Musician Creating Service
/v3/api-docs
Servers
http://localhost:8080 - local server.
basic-create-musician-controller
PUT /musicians Creates a new entry in the musician table
basic-music-viewing-controller
GET / Returns a list of Songs on an Album
basic-delete-album-controller
DELETE / Deletes an album and cascading songs from the library
Deletes an album and it's songs
Parameters
Cancel
Reset
No parameters
Request body required
application/json
{
"albumId": 1
}



The Database showing the album_id = 1 no longer there:



Database showing the cascade delete working properly, deleting the songs from the album_id = 1:

song | Enter a SQL expression to filter results (use Ctrl+Space)

Grid | Text | Record

	song_id	album_id	song_name	song_length
1	12	2	Say Hello To Heaven	00:06:24
2	13	2	Reach Down	00:11:12
3	14	2	Hunger Strike	00:04:06
4	15	2	Pushin Forward Back	00:03:45
5	16	2	Call Me A Dog	00:05:04
6	17	2	Times Of Trouble	00:05:42
7	18	2	Wooden Jesus	00:04:10
8	19	2	Your Savior	00:04:04
9	20	2	Four Walled World	00:06:54
10	21	2	All Night Thing	00:03:51
11	22	3	Let Me Drown	00:03:51
12	23	3	My Wave	00:05:12
13	24	3	Fell on Black Days	00:04:42
14	25	3	Mailman	00:04:25
15	26	3	Superunknown	00:05:06
16	27	3	Head Down	00:06:08
17	28	3	Black Hole Sun	00:05:18
18	29	3	Spoonman	00:04:06
19	30	3	Limo Wreck	00:05:47
20	31	3	The Day I Tried to Live	00:05:19
21	32	3	Kickstand	00:01:34
22	33	3	Fresh Tendrils	00:04:16
23	34	3	Fourth of July	00:05:08
24	35	3	Half	00:02:14

What I've learned and Looking Back:

SCRUM RETRO after the 1st week of the final project:

Looking at the big project ahead of me, I started out by making a list of tasks. This included: Creating an ERD, Creating the Database, Inserting Data into the Database and ensuring it runs properly, Creating a new Maven Project, then writing out the guts of the project beginning with a READ process. I was able to quickly knock out the ERD and all the database set up. I struggled more with building the guts, and getting the correct annotations where they belonged.

ROOT CAUSE ANALYSIS:

I learned a great deal about what many of the annotations actually do. It was very helpful to see the changes in the Swagger UI when I corrected the annotation @PostMapping to @PutMapping.

Building my first operation was slow and rough. I struggled to get the first function to run correctly. The second operation was a bit faster, but I still struggled through many errors. By the final one, I wrote it out quickly and in about half the time as my other ones!

PROJECT RETRO:

My goal was to get at least one of each type of function completed. Looking at other classmates projects. I see that I could have simplified my processes so that they could have been more easily copy and pasted to use for multiple entities instead of needing more detailed changes. Looking over other classmates or former students design is very helpful for thinking about the approach I will take when creating my next project. I hope to continue to grow in my organization and mapping skills.