

بسمه تعالی

عنوان پروژه

Dockerize CRUD Project with Django

موضوع پروژه

پلی لیست موزیک

درس:

رایانش ابری

استاد درس:

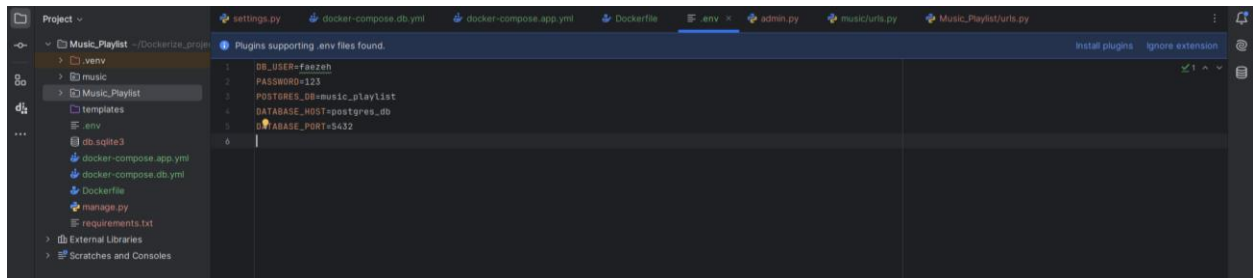
سرکار خانم دکتر هدی طاهری

دانشجو:

فائزه علی میرزایی – ۹۷۱۲۳۷۷۲۲۳

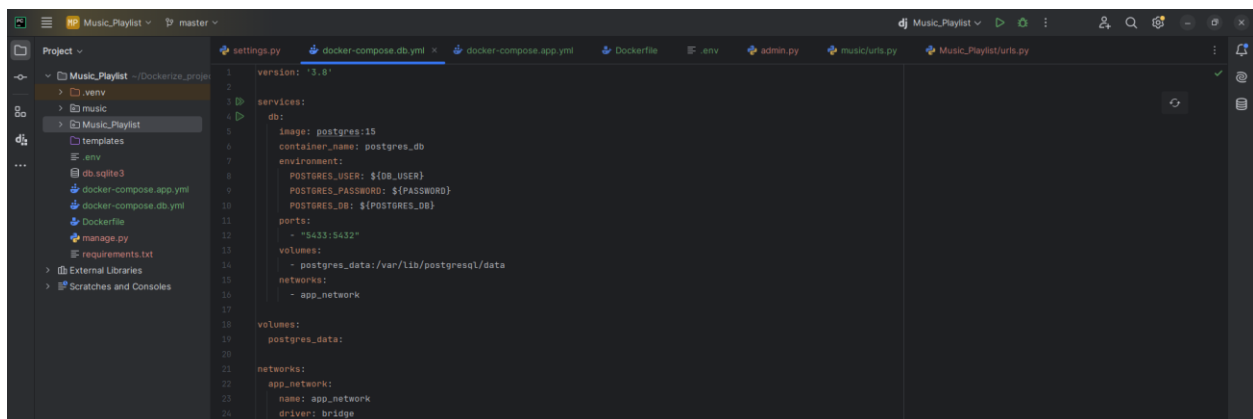
محتوای فایل های Docker-Compose

دو فایل docker-compose در پروژه داریم که یکی مربوط به دیتابیس و یکی مربوط به پروژه است. بعد از بالا آوردن کانتینر، این دو فایل با یکدیگر عملیات CRUD را انجام می‌دهند. فایل های docker-compose به شکل زیر هستند:



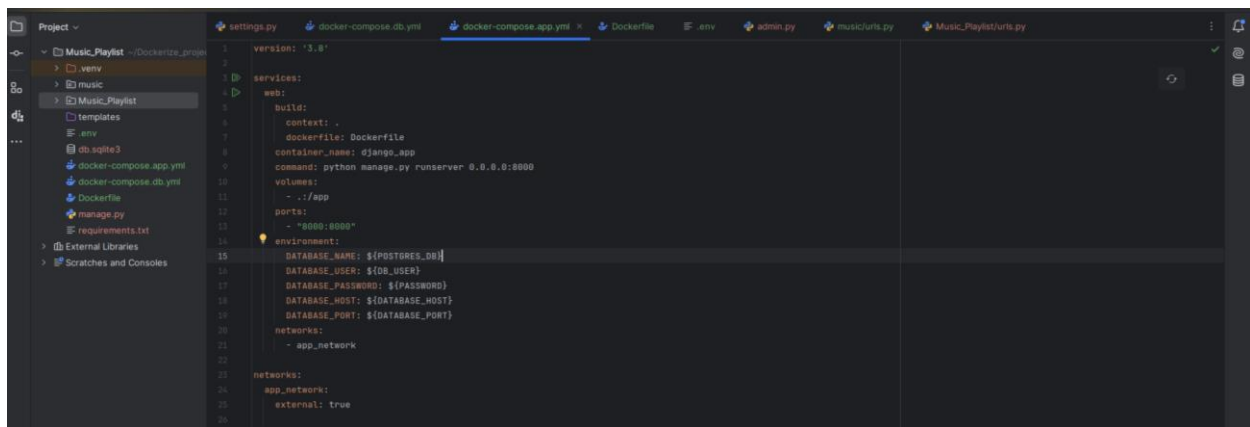
فایل .env

در این فایل مقادیر به صورت key,value نگهداری می‌شوند و از این مقادیر در docker-compose ها استفاده می‌شود، یعنی از کلید استفاده می‌کنیم و در زمان اجرا به جای کلید، مقدار آن استفاده می‌شود.



فایل docker-compose.db.yml

فایل بالا مربوط به دیتابیس است. کلید ها برای اتصال به دیتابیس در فایل env مشخص شده اند. اتصال دو کانتینر در شبکه app-network انجام می‌شد.



فایل docker-compose.app.yml

اجرای فایل های Docker-Compose

ابتدا باید کانینر های مربوط به docker-compose را build کرده و بالا بیاوریم. برای این کار دستور زیر را وارد می کنیم:

Docker-compose up –build

استفاده f– داریم، در این دستور باید با نام فایل صدا بزنیم و از docker-compose چون دو تا فایل استفاده می کنیم. کد دستوری زیر مربوط به d– از background کنیم. برای انتقال اجرای کانینرها به ایجاد و اجرای کانینر دیتابیس است:

Docker-compose -f docker-compose.db.yml up --build –d

```

faezeh@ubuntu: ~/Desktop/project$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
faezeh@ubuntu:~/Desktop/project$ sudo docker-compose -f docker-compose.db.yml up --build -d
Creating network "app_network" with driver "bridge"
Creating postgres_db ... done
faezeh@ubuntu:~/Desktop/project$ sudo docker-compose -f docker-compose.app.yml up --build -d
WARN[00] Found orphan containers (postgres_db) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
Building web
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/
Sending build context to Docker daemon  208.4kB
Step 1/8 : FROM python:3.10-slim
--> fed91319d6e7
Step 2/8 : RUN apt-get update && apt-get install -y gcc libpq-dev && rm -rf /var/lib/apt/lists/*
--> Using cache
--> d754b2be6e4d
Step 3/8 : WORKDIR /app
--> Using cache
--> 1c764c5b1b03
Step 4/8 : COPY requirements.txt .
--> Using cache
--> 19a08465d0a2
Step 5/8 : RUN pip install --no-cache-dir --upgrade pip && pip install --no-cache-dir -r requirements.txt
--> Using cache
--> 5a778f6ce405
Step 6/8 : COPY . .
--> 7175408b999
Step 7/8 : EXPOSE 8000
--> Running in d5daaa359c4
--> Removed intermediate container d5daaa359c4
--> fed9186fbd8
Step 8/8 : CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
--> Running in b765db28874
--> Removed intermediate container b765db28874
--> e574676b0b31
Successfully built e574676b0b31
Successfully tagged project_web:latest
Creating django_app ... done
faezeh@ubuntu:~/Desktop/project$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
638078cfd500   project_web   "python manage.py ru..."   11 seconds ago   Up 10 seconds   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp   django_app
fdbf7f9b732f   postgres:15   "docker-entrypoint.s..."   31 seconds ago   Up 31 seconds   0.0.0.0:5433->5432/tcp, [::]:5433->5432/tcp   postgres_db

```

```
faezeh@ubuntu: ~/Desktop/project
--> Using cache
--> 5a778f6ce405
Step 6/8 : COPY . .
--> 717540886590
Step 7/8 : EXPOSE 8000
--> Running in d5daae359c4
--> Removed intermediate container d5daae359c4
--> fe6391a6fbdb
Step 8/8 : CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
--> Running in b765dbb28874
--> Removed intermediate container b765dbb28874
--> e574676b0b31
Successfully built e574676b0b31
Successfully tagged project_web:latest
Creating django_app ... done
faezeh@ubuntu: ~/Desktop/project$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
638078cfd500   project_web   "python manage.py ru..." 11 seconds ago Up 10 seconds 0.0.0.0:8000->8000/tcp, :::8000->8000/tcp   django_app
fd6b7f9b732f   postgres:15   "docker-entrypoint.s..." 31 seconds ago Up 31 seconds 0.0.0.0:5433->5432/tcp, [::]:5433->5432/tcp   postgres_db
faezeh@ubuntu: ~/Desktop/project$ sudo docker exec -it django_app bash
root@638078cfd500:/app# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, music, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying music.0001_initial... OK
  Applying music.0002_music_delete_book... OK
  Applying sessions.0001_initial... OK
root@638078cfd500:/app#
```

پس از ایجاد و up کردن کانتینر ها، با اجرای دستور `docker ps` کانتینرهای در حال اجرا را میبینیم.

در عکس دوم مشاهده می‌شود که به کانتینر پروژه وارد شده ایم و با اجرای دستور زیر جداول مورد نیاز در پایگاه داده را ایجاد می‌کنیم:

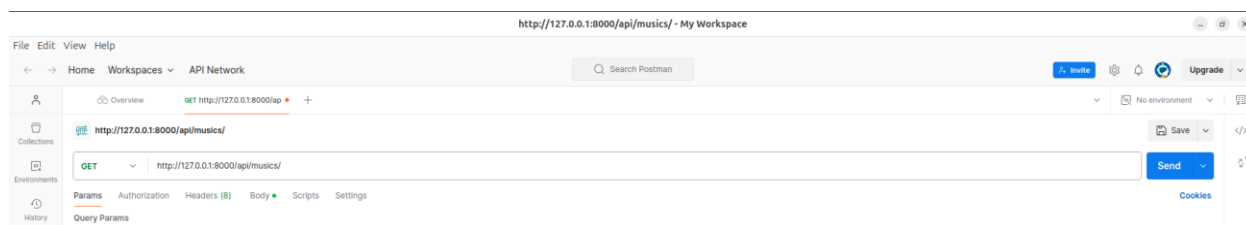
`Python manage.py migrate`

برای ورود به کانتینر پروژه از دستور زیر استفاده می‌کنیم:

`Docker exec -it django_app bash`

در این مرحله کانتینر ها ساخته شده، شبکه شده و آماده استفاده هستند. در نهایت تست و اجرای دستورات CRUD روی پروژه انجام می‌شود.

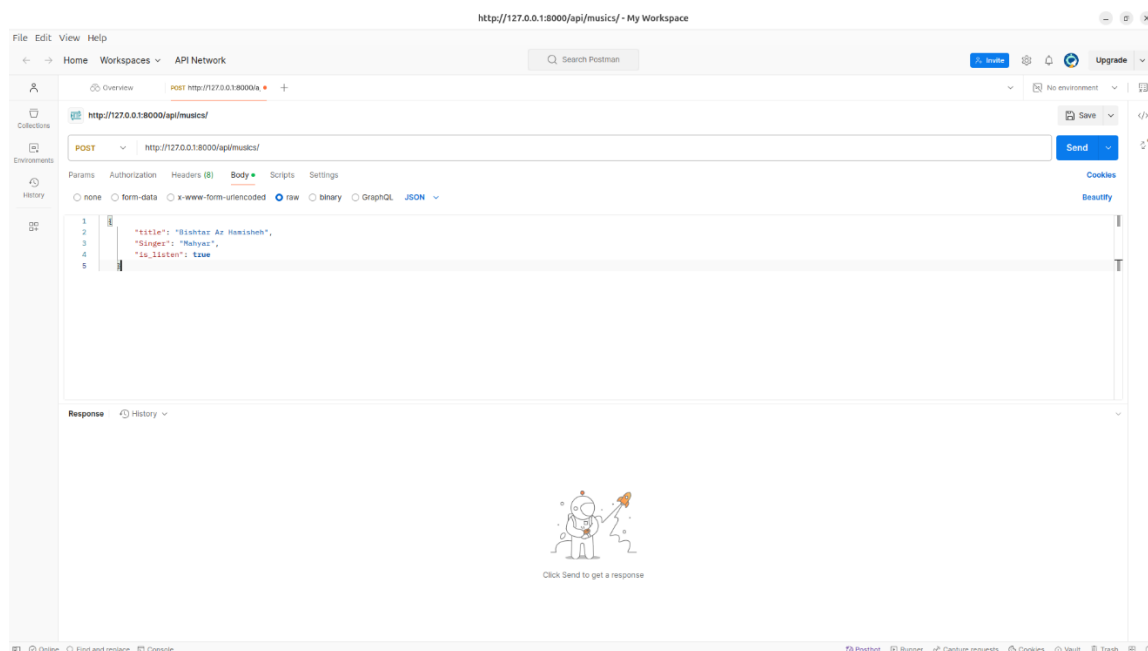
عملیات CRUD با استفاده از Postman

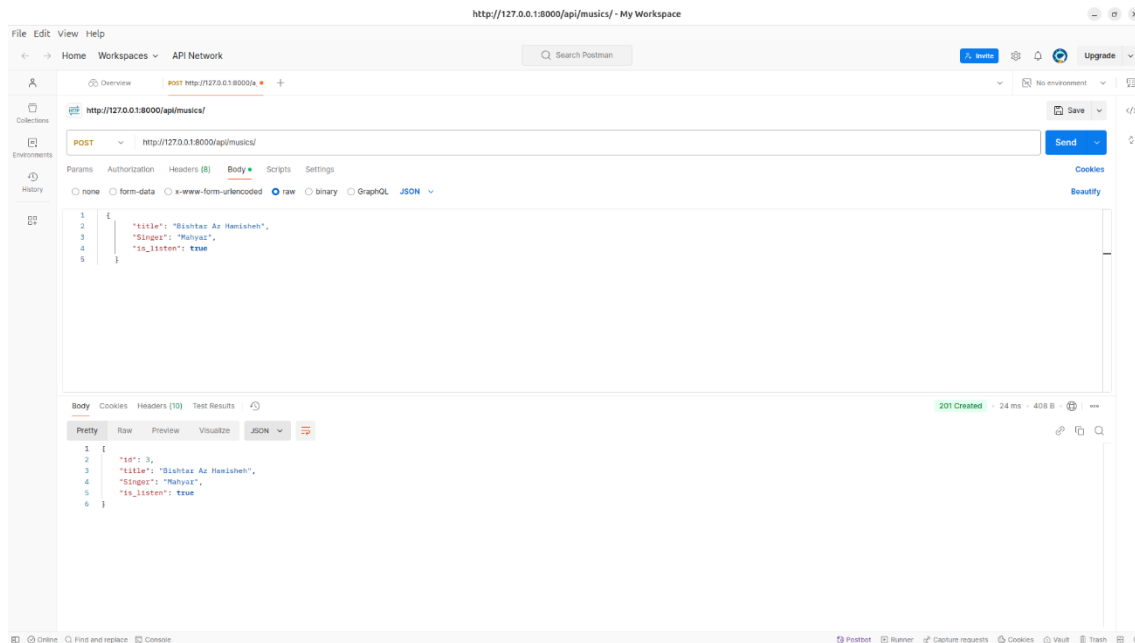


اتصال به **postman**: لینک اجرایی پروژه را مطابق بالا به postman می‌دهیم.

عملیات **Create**: در postman همان عملیات post می‌باشد.

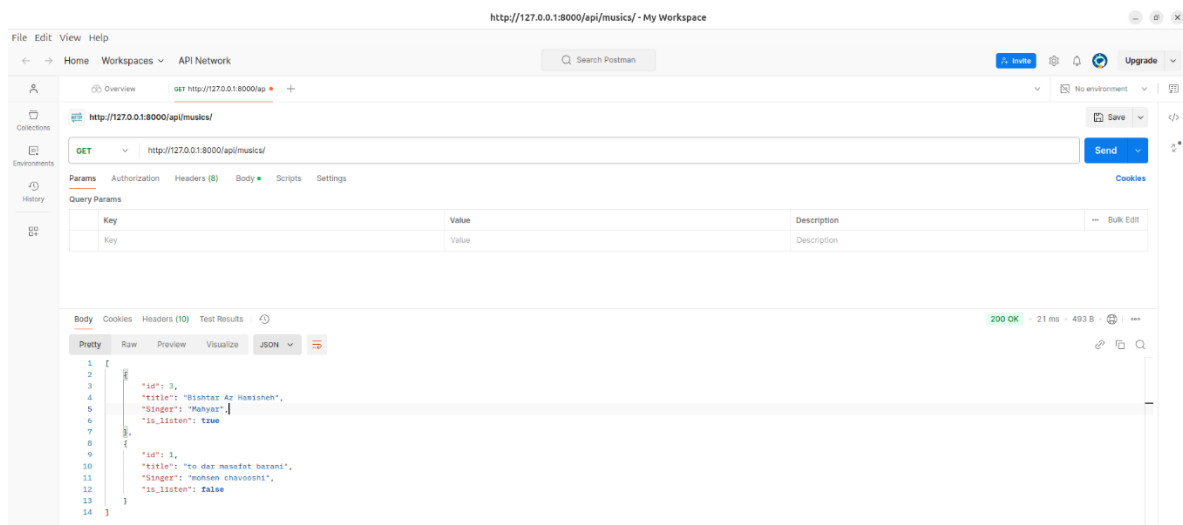
برای این کار از منوی کشویی که کنار لینک قرار دارد باید گزینه post را انتخاب کنیم. سپس در سر برگ Body گزینه raw را انتخاب کرده و داده ورودی برای ذخیره در پایگاه داده را به صورت فایل json نوشته و دکمه send را می‌زنیم.





مطابق تصویر بالا، بعد از ذخیره داده در دیتابیس پیام 201 created نمایش داده می شود یعنی عملیات با موفقیت انجام شده است.

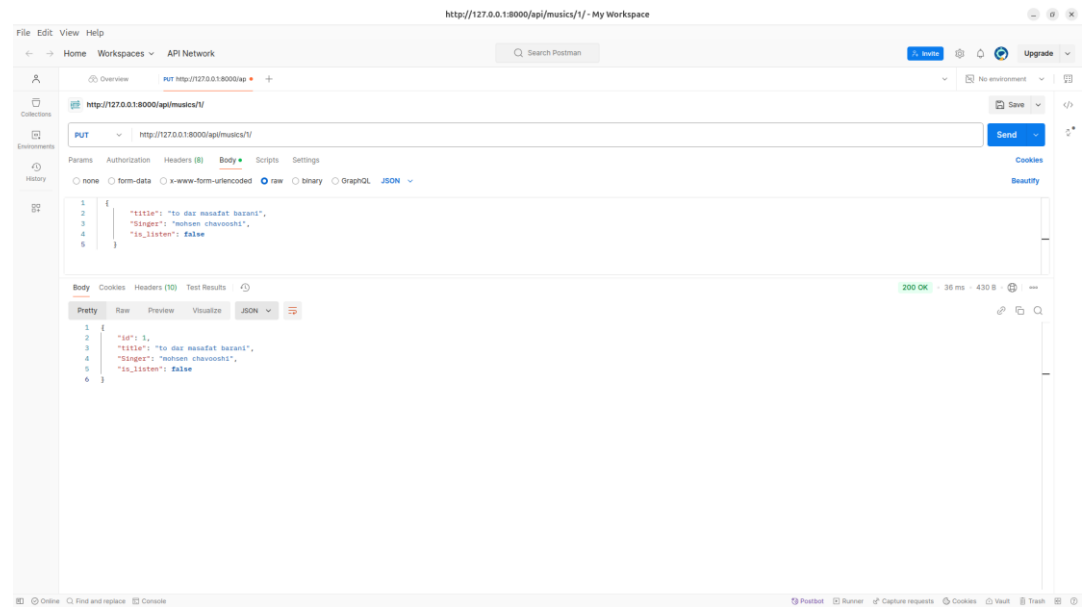
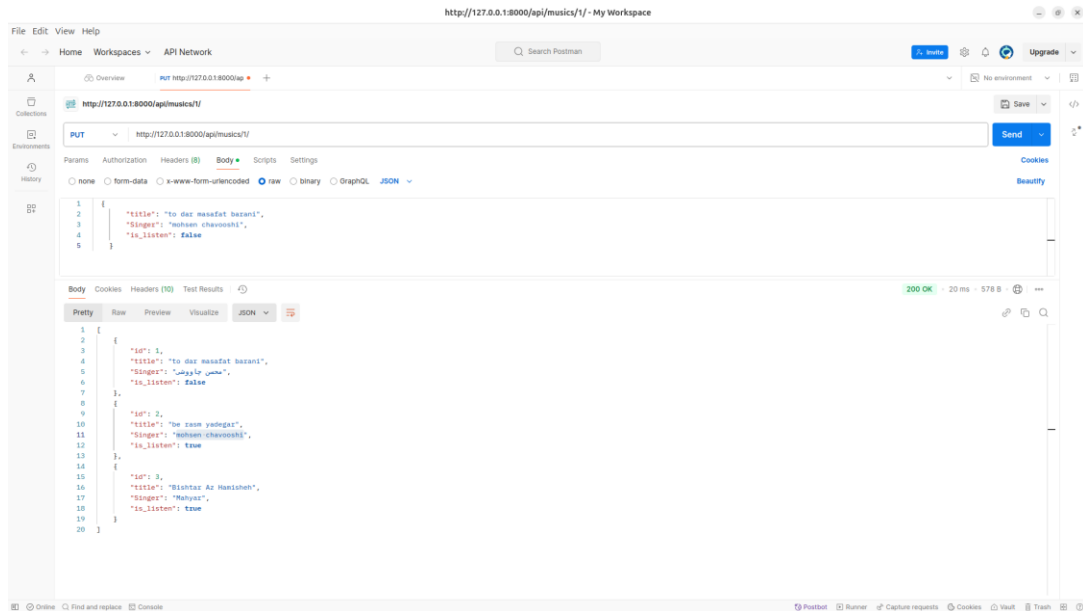
عملیات Read: در postman از گزینه get استفاده می کنیم.



پیام 200 OK نشان دهنده موفقیت آمیز بودن عملیات است و postman مقادیر داخل پایگاه داده را به ما برمی گرداند.

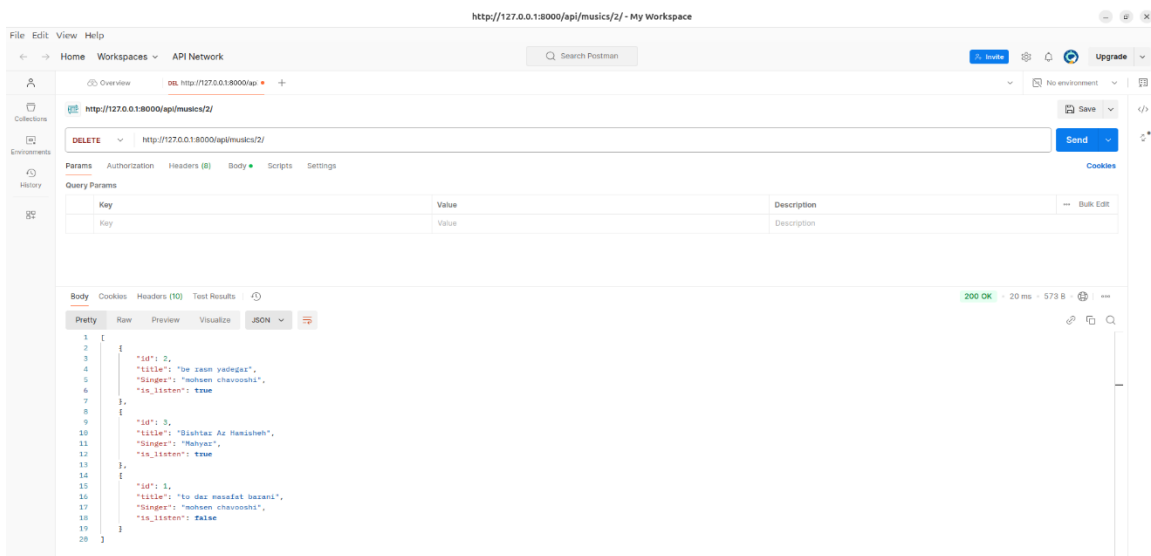
عملیات **update**: برای اینکار از put در postman استفاده می کنیم.

برای put کردن کافیت id آن مقداری که میخوایم تغییری در آن ایجاد کنیم، کنار لینک اجرایی پروژه مطابق تصویر وارد کرده و تغییرات مدنظر را اعمال کرده و سپس دکمه send را بزنیم.

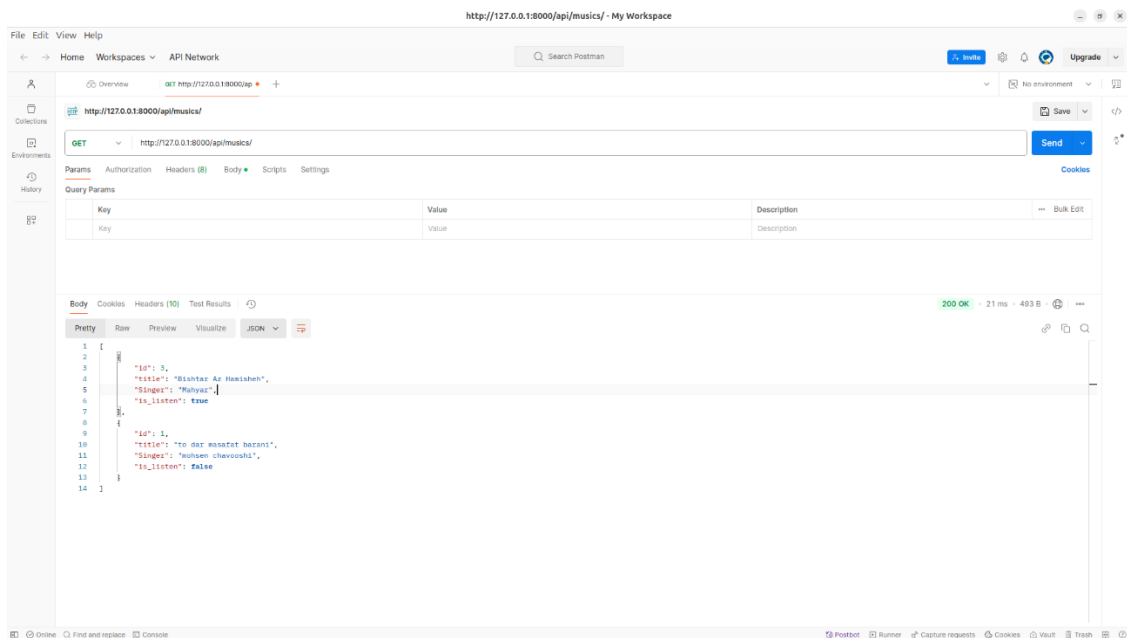


در عکس دوم حاصل اعمال تغییرات را در داده دیتابیس نشان می دهد.

Delete: مانند عملیات قبلی باید آیدی مقدار مورد نظر در کنار لینک وارد شود.



اگر پس از حذف مقدار بالا از دستور get استفاده کنیم، خروجی مطابق تصویر پایین خواهد بود:



در نهایت فقط این دو مقدار در پایگاه داده باقی مانده‌اند.

جهت توقف اجرای کانٹینر ها دستورات زیر را در ترمینال وارد می کنیم:

```
faezeh@ubuntu: ~/Desktop/project
faezeh@ubuntu:~/Desktop/project$ sudo docker-compose -f docker-compose.app.yml down
Stopping django_app ... done
WARNING: Found orphan containers (postgres_db) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to
clean it up.
Removing django_app ... done
Network app_network is external, skipping
faezeh@ubuntu:~/Desktop/project$ sudo docker-compose -f docker-compose.db.yml down
Stopping postgres_db ... done
Removing postgres_db ... done
Removing network app_network
faezeh@ubuntu:~/Desktop/project$
```