

Day 11

Date-19/08/25

VIJAY M

Create Fast API, to store, retrieve all and retrieve specific song by Id. SQLite can be used as data store table can contain song name, playing time in seconds, album(or movie), artist along with Id.

Add an additional table called users. put api post request to insert user into it in fastapi. Additionally, when request to list music api is called , only a valid user is allowed to get the response else raise HTTPException with error code

Main.py

```
from sqlalchemy import create_engine, Column, Integer, String
from sqlalchemy.orm import sessionmaker, Session, declarative_base
from fastapi import FastAPI, Depends, Request, HTTPException
from pydantic import BaseModel
from typing import List
import uvicorn

url = "sqlite:///./song.db"

engine = create_engine(url, connect_args={"check_same_thread": False})
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)

Base = declarative_base()

app = FastAPI()

class User(Base):
    __tablename__ = "user"

    Id = Column(Integer, primary_key=True, index=True)
    Name = Column(String(100), nullable=False)
```

```
Base.metadata.create_all(bind=engine)
```

```
class Users(BaseModel):
```

```
    Id:int
```

```
    Name:str
```

```
class Song(Base):
```

```
    __tablename__ = "songs"
```

```
    id = Column(Integer, primary_key=True, index=True)
```

```
    title = Column(String(100), nullable=False)
```

```
    artist = Column(String(100), nullable=False)
```

```
    album = Column(String(100), nullable=False)
```

```
    playing_time = Column(Integer, nullable=False) # in seconds
```

```
Base.metadata.create_all(bind=engine)
```

```
class SongCreate(BaseModel):
```

```
    title: str
```

```
    artist: str
```

```
    album: str
```

```
    playing_time: int
```

```
class SongOut(SongCreate):
```

```
    id: int
```

```
    class Config:
```

```
        orm_mode = True
```

```
class SongDTO(BaseModel):
```

```
    title: str
```

```
class Config:
    orm_mode = True
```

```
def get_db():
    db = SessionLocal()
    try:
        yield db
    finally:
        db.close()
```

```
@app.post("/add_user",response_model=Users)
```

```
def add_user(user:Users,db: Session = Depends(get_db)):
    u=User(Id=user.Id,Name=user.Name)
    db.add(u)
    db.commit()
    db.refresh(u)
    return u
```

```
@app.post("/{name}/add_song", response_model=SongOut)
```

```
def add_song(name:str,song: SongCreate, db: Session = Depends(get_db)):
    user=db.query(User).filter(User.Name==name).first()
    if user==None:
        raise HTTPException(status_code=401, detail="No user found - Unauthorized access")
    else:
        s = Song(
            title=song.title,
            artist=song.artist,
            album=song.album,
```

```
    playing_time=song.playing_time
)

db.add(s)

db.commit()

db.refresh(s)

return s
```

```
@app.get("/{name}/get_songs", response_model=List[SongOut])
def get_songs(name:str,db: Session = Depends(get_db)):
    user=db.query(User).filter(User.Name==name).first()
    if user==None:
        raise HTTPException(status_code=401, detail="No user found - Unauthorized access")
    return db.query(Song).all()
```

```
@app.get("/{name}/get_song/{id}", response_model=SongDTO)
def get_song(name:str,id: int, db: Session = Depends(get_db)):
    user=db.query(User).filter(User.Name==name).first()
    if user==None:
        raise HTTPException(status_code=401, detail="No user found - Unauthorized access")
    s = db.query(Song).filter(Song.id == id).first()
    if not s:
        raise HTTPException(status_code=404, detail="Song not found")
    return SongDTO(title=s.title)
```

```
@app.middleware("http")
async def log_requests(request: Request, call_next):
    print("Middleware intercepted the request.")
    response = await call_next(request)
```

return response

if __name__ == "__main__":

uvicorn.run("main:app", host="127.0.0.1", port=8000, reload=True)

Output:

`/ {name} / add_song`

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/vimal/add_song' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "unna1 unna1",
    "artist": "ARR",
    "album": "Abmigabadhi",
    "playing_time": 200
  }'
```

Request URL

`http://127.0.0.1:8000/vimal/add_song`

Server response

Code

Details

200

Response body

```
{
  "title": "unna1 unna1",
  "artist": "ARR",
  "album": "Abmigabadhi",
  "playing_time": 200,
  "id": 5
}
```

Response headers

```
content-length: 86
content-type: application/json
date: Tue, 19 Aug 2025 11:51:58 GMT
server: uvicorn
```

`/ {name} / add_song`

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/vimaldfs/add_song' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "unnal unnal",
    "artist": "ARR",
    "album": "Abmigabadhi",
    "playing_time": 200
  }'
```

Request URL

http://127.0.0.1:8000/vimaldfs/add_song

Server response

Code

Details

401

Undocumented

Error: Unauthorized

Response body

```
{
  "detail": "No user found - Unauthorized access"
}
```

Response headers

```
content-length: 48
content-type: application/json
date: Tue, 19 Aug 2025 11:52:45 GMT
server: uvicorn
```

Responses

`/`**`{name}/get_songs`**

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/vimalds/get_songs' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/vimalds/get_songs
```

Server response

Code

Details

401

Undocumented

Error: Unauthorized

Response body

```
{
  "detail": "No user found - Unauthorized access"
}
```

Response headers

```
content-length: 48
content-type: application/json
date: Tue, 19 Aug 2025 11:50:42 GMT
server: uvicorn
```

← → ↻ ⓘ 127.0.0.1:8000/vimaldkg/get_songs

Pretty-print ☐

```
{"detail": "No user found - Unauthorized access"}
```

`/{name}/get_songs`

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/vimal/get_songs' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/vimal/get_songs
```

Server response

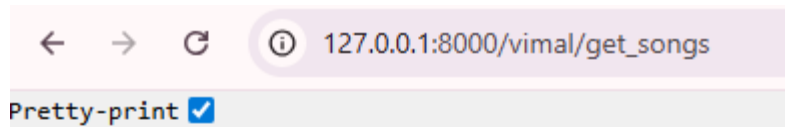
Code

Details

200

Response body

```
[
  {
    "title": "vathi comming",
    "artist": "aniruth",
    "album": "master",
    "playing_time": 180,
    "id": 1
  },
  {
    "title": "nan ready",
    "artist": "aniruth",
    "album": "leo",
    "playing_time": 240,
    "id": 2
  },
  {
    "title": "Once Upon a time",
    "artist": "aniruth",
    "album": "vikram",
    "playing_time": 253,
    "id": 3
  },
  {
    "title": "im the danger",
    "artist": "aniruth",
    "album": "coolie",
    "playing_time": 130,
  }
```

```
[
  {
    "title": "vathi coming",
    "artist": "aniruth",
    "album": "master",
    "playing_time": 180,
    "id": 1
  },
  {
    "title": "nan ready",
    "artist": "aniruth",
    "album": "leo",
    "playing_time": 240,
    "id": 2
  },
  {
    "title": "Once Upon a time",
    "artist": "aniruth",
    "album": "vikram",
    "playing_time": 253,
    "id": 3
  },
  {
    "title": "im the danger",
    "artist": "aniruth",
    "album": "coolie",
    "playing_time": 130,
    "id": 4
  }
]
```

`/vimal/get_song/{id}`

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/vimal/get_song/2' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/vimal/get_song/2
```

Server response

Code

Details

200

Response body

```
{
  "title": "nan ready"
}
```

Response headers

```
content-length: 21
content-type: application/json
date: Tue, 19 Aug 2025 11:48:12 GMT
server: uvicorn
```

Responses

← → ↻ ⓘ 127.0.0.1:8000/vimal/get_song/2

pretty-print ☒

```
{
  "title": "nan ready"
}
```

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/vimalsd/get_song/2' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/vimalsd/get_song/2
```

Server response

Code

Details

401

Undocumented

Error: Unauthorized

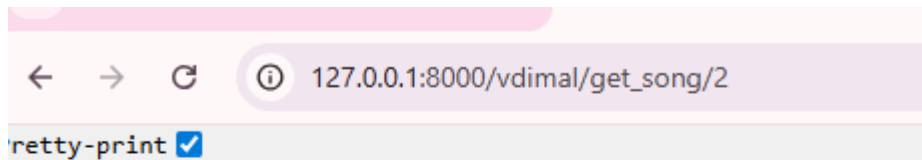
Response body

```
{
  "detail": "No user found - Unauthorized access"
}
```

Response headers

```
content-length: 48
content-type: application/json
date: Tue, 19 Aug 2025 11:49:44 GMT
server: uvicorn
```

Responses



```
"detail": "No user found - Unauthorized access"
```

/add_user

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/add_user' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Id": 5,
    "Name": "rajini"
  }'
```

Request URL

`http://127.0.0.1:8000/add_user`

Server response

Code

Details

200

Response body

```
{
  "Id": 5,
  "Name": "rajini"
}
```

Response headers

```
content-length: 24
content-type: application/json
date: Tue, 19 Aug 2025 12:08:02 GMT
server: uvicorn
```

Responses

user

Table: user



» Filter in

	<u>Id</u>	Name
	Filter	Filter
1	0	suriya
2	1	kamal
3	2	vimal
4	3	danush
5	4	rajan
6	5	rajini