

Day 10

VIJAY M

date-18/08/25

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

creating virtual environment

`python -m venv myapp`

activate

`myapp\Scripts\activate`

Install packages

`pip install fastapi sqlalchemy uvicorn`

song.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 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_song", response_model=SongOut)
def add_song(song: SongCreate, db: Session = Depends(get_db)):

    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("/get_songs", response_model=List[SongOut])
def get_songs(db: Session = Depends(get_db)):

    return db.query(Song).all()

@app.get("/get_song/{id}", response_model=SongDTO)
def get_song(id: int, db: Session = Depends(get_db)):

    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("song:app", host="127.0.0.1", port=8000, reload=True)
```

Output:

/add_song

Request URL	
http://127.0.0.1:8000/add_song	
Server response	
Code	Details
200	<div><div>Response body</div><pre>{ "title": "Once Upon a time", "artist": "aniruth", "album": "vikram", "playing_time": 253, "id": 3 }</pre></div> <div><div>Response headers</div><pre>content-length: 90 content-type: application/json date: Mon, 18 Aug 2025 11:50:21 GMT server: uvicorn</pre></div>
Response	

/get_songs

Request URL	
http://127.0.0.1:8000/get_songs	
Server response	
Code	Details
200	<p>Response body</p> <pre>[{ "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 }]</pre>

/get_song/1

Request URL	
http://127.0.0.1:8000/get_song/1	
Server response	
Code	Details
200	<p>Response body</p> <pre>{ "title": "vathi comming" }</pre> <p>Response headers</p> <pre>content-length: 25 content-type: application/json date: Mon, 18 Aug 2025 11:50:39 GMT server: uvicorn</pre>

middleware

```
warnings.warn(message, UserWarning)
INFO:      Started server process [2832]
INFO:      Waiting for application startup.
INFO:      Application startup complete.
Middleware intercepted the request.
INFO:      127.0.0.1:53475 - "GET /docs HTTP/1.1" 200 OK
Middleware intercepted the request.
INFO:      127.0.0.1:53475 - "GET /openapi.json HTTP/1.1" 200 OK
Middleware intercepted the request.
INFO:      127.0.0.1:53476 - "POST /add_song HTTP/1.1" 200 OK
Middleware intercepted the request.
INFO:      127.0.0.1:53500 - "POST /add_song HTTP/1.1" 200 OK
Middleware intercepted the request.
INFO:      127.0.0.1:53510 - "POST /add_song HTTP/1.1" 200 OK
Middleware intercepted the request.
INFO:      127.0.0.1:53512 - "GET /get_songs HTTP/1.1" 200 OK
Middleware intercepted the request.
INFO:      127.0.0.1:53517 - "GET /get_song/1 HTTP/1.1" 200 OK
█
```