

Documentation for routes/lyrics.py

This module contains the endpoints for the lyrics service. It provides routes for searching for lyrics by song title and artist, and for retrieving lyrics using a public api <https://api.lyrics.ovh>.

Endpoints

Endpoint: /lyrics/random-lyrics

Fetches a random song from the music library along with its lyrics from the lyrics.ovh API.

- **user:** User - The authenticated user making the request.
- **db:** Session - The database session for querying the database.
- **return:** Returns a JSON object containing the song's ID, details, and lyrics.

Source code in routes/lyrics.py :

```
@router.get("/random-lyrics", tags=["lyrics"])
def get_random_row(user=Depends(login_manager), db: Session = Depends(get_db)):
    with db:
        row = db.query(MusicLibrary).order_by(func.random()).first()
        if row is None:
            raise HTTPException(status_code=404, detail="No songs found in the library")
        lyrics = fetch_lyrics(row.artist, row.title)
        return {"id": row.id, "row": row, "lyrics": lyrics}
```

Endpoint: /lyrics/random-lyrics-metadata

Fetches a random song from the music library along with its lyrics and metadata including artwork.

- **user:** User - The authenticated user making the request.
- **db:** Session - The database session for querying the database.
- **return:** Returns a JSON object containing the song's ID, details, lyrics from the lyrics.ovh API, and artwork from the metadata.

Source code in routes/lyrics.py :

```
@router.get("/random-lyrics-metadata", tags=["lyrics"])
def get_random_row(user=Depends(login_manager), db: Session = Depends(get_db)):
    with db:
        row = db.query(MusicLibrary).order_by(func.random()).first()
        if row is None:
            raise HTTPException(status_code=404, detail="No songs found in the library")
        lyrics = fetch_lyrics(row.artist, row.title)
        return {"id": row.id, "row": row, "lyrics": lyrics, "metadata": row.metadata}
```

```
lyrics = fetch_lyrics(row.artist, row.title)
artwork = get_artwork("megasetbucket", row.filepath)
return {"id": row.id, "row": row, "lyrics": lyrics, "artwork": artwork}
```

For more details, visit the [documentation](#).