

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 <https://api.lyrics.ovh>

```
get_random_row(user=Depends(login_manager),
db=Depends(get_db))
```

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`

```
15 @router.get("/random-lyrics", tags=["lyrics"])
16 def get_random_row(user=Depends(login_manager), db: Session =
17 Depends(get_db)):
18     """
19     Fetches a random song from the music library along with its lyrics from
20     the lyrics.ovh API.
21
22     - **user**: User - The authenticated user making the request.
23     - **db**: Session - The database session for querying the database.
24     - **return**: Returns a JSON object containing the song's ID, details,
25     and lyrics.
26     """
27     with db:
28         row = db.query(MusicLibrary).order_by(func.random()).first()
29         if row is None:
30             raise HTTPException(status_code=404, detail="No songs found in
the library.")
31         lyrics = fetch_lyrics(row.artist, row.title)
32         return {"id": row.id, "row": row, "lyrics": lyrics}
```

```
get_random_row_and_lyrics_and_metadata(user=Depends(login_manager), db=Depends(get_db))
```

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`

```

32 @router.get("/random-lyrics-metadata", tags=["lyrics"])
33 def get_random_row_and_lyrics_and_metadata(user=Depends(login_manager), db:
34 Session = Depends(get_db)):
35     """
36     Fetches a random song from the music library along with its lyrics and
37     metadata including artwork.
38
39     - **user**: User - The authenticated user making the request.
40     - **db**: Session - The database session for querying the database.
41     - **return**: Returns a JSON object containing the song's ID, details,
42     lyrics from the lyrics.ovh API, and artwork from the metadata.
43     """
44     with db:
45         row = db.query(MusicLibrary).order_by(func.random()).first()
46         if row is None:
47             raise HTTPException(status_code=404, detail="No songs found in
the library.")
48         lyrics = fetch_lyrics(row.artist, row.title)
49         artwork = get_artwork("megasetbucket", row.filepath)
50         return {"id": row.id, "row": row, "lyrics": lyrics, "artwork":
artwork}

```