

# Simple HLS server in Python using the Flask web framework

In this example, the Flask application has two routes: `/live/<stream_id>.m3u8` and `/live/<stream_id>/<int:sequence>.ts`. The first route uses the `ffmpeg` command-line tool to convert an RTMP stream into an HLS stream, and returns the contents of the `.m3u8` playlist file. The second route returns the contents of individual `.ts` segments.

To run this server, you'll need to have the Flask web framework and the `ffmpeg` command-line tool installed. You can start the server by running the Python script and access the HLS stream by opening [http://localhost:5000/live/<stream\\_id>.m3u8](http://localhost:5000/live/<stream_id>.m3u8) in a media player that supports HLS.

```
from flask import Flask, Response
import subprocess

app = Flask(__name__)

@app.route('/live/<stream_id>.m3u8')
def m3u8(stream_id):
    ffmpeg = subprocess.Popen(['ffmpeg', '-i', 'rtmp://localhost/live/' + stream_id, '-c:v', 'h264', '-c:a', 'aac', '-f', 'hls', '-hls_time', '10', '-hls_list_size', '0', '-hls_flags', 'delete_segments', 'static/' + stream_id + '.m3u8'], stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    return Response(ffmpeg.stdout, mimetype='application/x-mpegURL')

@app.route('/live/<stream_id>/<int:sequence>.ts')
def ts(stream_id, sequence):
    with open('static/' + stream_id + '/' + str(sequence) + '.ts', 'rb') as f:
        return f.read()

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Below is based on the multiple sources :

```
from flask import Flask, Response

app = Flask(__name__)

@app.route('/live/<stream_id>.m3u8')
def m3u8(stream_id):
    with open(f"static/{stream_id}.m3u8", "r") as f:
        return Response(f.read(), mimetype="application/x-mpegURL")

@app.route('/live/<stream_id>/<int:sequence>.ts')
def ts(stream_id, sequence):
    with open(f"static/{stream_id}/{sequence}.ts", "rb") as f:
        return f.read()

if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

In above example, the Flask application has two routes: `/live/<stream_id>.m3u8` and `/live/<stream_id>/<int:sequence>.ts`. The first route returns the contents of the `.m3u8` playlist file for the specified stream. The second route returns the contents of individual `.ts` segments for the specified stream and sequence number.

To run this server, you'll need to have the Flask web framework installed. The two HLS files should be stored in a `static` directory and the paths to the `.m3u8` playlist files and the `.ts` segments should match the paths specified in the routes. You can start the server by running the Python script and access the HLS streams by opening [http://localhost:5000/live/<stream\\_id>.m3u8](http://localhost:5000/live/<stream_id>.m3u8) in a media player that supports HLS.