Loopy Cool - Complete Technical Documentation

Loopy Cool is a Progressive Web Application (PWA) designed for real-time audio looping, layered track creation, waveform visualization, master recording, and cloud-authenticated usage via Google Identity Services. It leverages the Web Audio API, MediaRecorder API, and modern browser capabilities to provide a studio-like loop creation environment directly in the browser without external plugins.

Core Features

- Multi-track loop creation with independent playback control.
- Real-time microphone recording using MediaRecorder API.
- Waveform visualization powered by AnalyserNode.
- Loop playback using AudioBufferSourceNode with seamless looping.
- Master gain control and mute functionality.
- Master mix recording exported as high-quality WAV file.
- Google Sign-In authentication using Google Identity Services.
- LocalStorage persistence for saved tracks.
- PWA installation support with Service Worker caching.
- Responsive design optimized for desktop and mobile.

Technical Architecture

Loopy Cool is structured around the Web Audio API audio graph system: 1. AudioContext acts as the central engine. 2. Individual tracks create AudioBufferSourceNode instances. 3. Each source connects to an AnalyserNode for waveform animation. 4. All tracks route into a master GainNode. 5. The master GainNode connects to the destination output. Master recording is implemented using a ScriptProcessorNode that captures the mixed stereo signal from the master GainNode and reconstructs a final AudioBuffer which is encoded to WAV format manually.

Best Practices for Users

- Record loops at consistent tempo for seamless layering.
- Keep loops short (2–8 seconds) for optimal memory performance.
- Adjust master volume to avoid clipping during layered playback.
- Stop tracks before deleting to prevent audio artifacts.
- Install as PWA for best performance and offline access.

Deep Technical Implementation Details

Audio Recording: MediaRecorder captures raw microphone input as audio blobs. These blobs are decoded into AudioBuffer using decodeAudioData and stored within track objects. Waveform Rendering: Waveforms are generated either statically (analyzing channel data samples) or dynamically using AnalyserNode frequency data for real-time animation. WAV Encoding: Custom WAV encoding writes RIFF headers manually including: - Chunk ID - File size - Format - Subchunk1 (fmt) - Subchunk2 (data) Each Float32 sample is converted to 16-bit PCM format before being written into the DataView buffer. Persistence: Tracks are serialized into Base64 WAV strings and saved into localStorage. Upon load, Base64 is decoded and reconstructed into AudioBuffer objects. Security: - Google Identity token decoding uses Base64URL parsing. - Service Worker enables offline caching. - DevTools shortcut blocking implemented (client-side deterrent only).

Performance Optimization Notes

- Keep FFT size small (32) for lightweight waveform animation. - Use requestAnimationFrame for smooth visual updates. - Disconnect unused nodes to prevent memory leaks. - Avoid large AudioBuffers to reduce RAM usage. - Limit track count for mobile devices.

Conclusion

Loopy Cool combines modern web technologies to create a fully functional browser-based loop station. By leveraging Web Audio API architecture, PWA capabilities, and optimized UI rendering, it delivers a seamless music creation experience without requiring external software. This documentation serves as a full technical and user guide reference for developers and advanced users seeking to understand or extend the Loopy Cool platform.