Introducing libSquoosh

We are happy to introduce libSquoosh, an experimental Node library on top of which the Squoosh CLI is built, giving you all the capabilities for the Squoosh CLI with a JavaScript-idiomatic interface.

Squoosh.app is a PWA that compresses images for you in the browser. It supports many old and new image formats and processes them client-side in the browser through WebAssembly. This means your pictures stay safely on your own computer rather than being sent to a server somewhere, and that Squoosh works even when offline.

At Chrome DevSummit 2020 we announced Squoosh v2, together with the Squoosh CLI to bring all the codecs of Squoosh to the command-line using Node and WebAssembly. This allows you to compress entire folders with one command and make use of the CLI's to let it choose the codec parameters for you.

The CLI enables a lot of automation and so it's only natural that developers began asking for a more idiomatic interface than programmatically invoking the Squoosh CLI via the shell. Anton (@atjn on GitHub) stepped up to the task and separated the Squoosh CLI code into two parts: The command line interface code and the underlying core functionality.

```
import { ImagePool } from "@squoosh/lib";

// libSquoosh uses a worker-pool under the hood
// to parallelize all image processing.
const imagePool = new ImagePool();

// Accepts both file paths and Buffers/TypedArrays.
const image = imagePool.ingestImage("./squoosh.jpeg");

// Optional.
// await image.preprocess({
//   resize: {
//     enabled: true,
//     width: 128,
//   },
// });
```

```
await image.encode({
  // All codecs are initialized with default values
  // that can be individually overwritten.
  mozjpeg: {
    quality: 10,
  },
  avif: {
    cqLevel: 10,
  },
  jxl: {},
});

const { extension, binary } = await image.encodedWith.mozjpeg;
await fs.writeFile(`output.${extension}`, binary);
// ... same for other encoders ...

await imagePool.close();
```

Our goal is to make image compression more accessible to tooling authors. We hope to see integration into Webpack, Rollup and other build tools to make sure your images are appropriately optimized for the web.

I'd like to express a huge "thank you" to Anton for the time he has committed to Squoosh!

It's still early for the Squoosh CLI and libSquoosh and we have many more ideas and plans that we'd like to implement. In the meanwhile, try libSquoosh! However, be mindful that this is an early, experimental release and that there is a good chance you will run into some bugs. If you find some or have questions, please open an issue.

If you are interested in contributing to Squoosh—for example contributing to the extremely sparse documentation around libSquoosh or help with any of the other parts of the app—we are starting a mentorship program to help you get started. If you want to know more, check out our tracking issue.

□□□ #2
□ □□□ □□□ 2026-10-31 21:59:02 UTC
□ □□□ □□□ 2026-10-31 21:59:03 UTC