KAMIL OGÓREK

Software Engineer

About

I am a fervent enthusiast of software development, with a particular interest in web technologies. My journey includes a wealth of experience as a remote employee, contributing to open source projects, speaking at events, and leading workshops. I've devoted numerous hours to experimenting with a variety of languages, frameworks, and tools. This diverse background has molded me into the professional I am today. In my work, I never settle for less, constantly raising the bar and striving for the most optimal solutions.

As a firm advocate of Software Craftsmanship, I am always on a quest for knowledge, continually enhancing my skills, and generously giving back to the community by frequently contributing to open-source projects.

Currently, I hold the position of a TC39 Invited Expert under the Tooling Group, serve as a core member of tRPC, and regularly contribute to Deno.

Off the clock, I'm a fitness and nutrition enthusiast, climber, cyclist, and retired Olympic weightlifter. My passions extend to drumming, music, cooking, and appreciating great food.

Psst... Scroll down to read about my work!

Contact

kamil.ogorek@gmail.com

kamilogorek.com
github.com/kamilogorek
twitter.com/kamilogorek

Skills

Whether it's JavaScript, TypeScript, Node.js, Go, Rust, Workflow Automation, Developer Tooling, or any other technology in your project's stack, I'm up for the challenge. I don't believe in merely listing a plethora of random technologies. Throughout my career, I've dabbled in everything from Python, Ruby, Clojure, to Elixir. If I haven't worked with it, played with it, or heard about it, you can bet that I can learn it.

Experience

Senior Software Engineer @ Sentry
August 2017 – Present

Lead Code Whisperer @ Corgibytes, LLC
August 2016 – August 2017

Senior Client-side Engineer @ X-Team
June 2014 – August 2016

Client-side Engineer @ X-Team
April 2013 – June 2014

Front-end Developer @ Chilid
March 2012 – May 2013

Front-end Developer @ XSolve
March 2012 – May 2013

Front-end Developer @ XHTMLized
March 2012 – May 2013

Public Speaking

"JavaScript Source Maps, Can We Do Better?"
JSNation, Amsterdam 2023
Meet.js, Kraków 2023

"Become a Front-end Developer – React.js Workshops"
Coder's Lab, Warsaw 2015-2017

Become a Front-end Developer – JavaScript Workshops"
Coder's Lab, Warsaw 2015-2017

"Ampersand.js – Minimalistic approach to not so min. problems"
Reject.js, Berlin, 2014
Meet.js Summit, Poznań, 2014

"Keep Your Projects Tight"
Meet.js, Kraków, 2013

Open Source

https://github.com/getsentry/sentry-javascript

JavaScript/TypeScript/Node.js - creator

Over the four years, since I took over the old Sentry SDK, I've adopted a fresh approach and built an entirely new, fully isomorphic SDK for our JavaScript needs from scratch. After nearly 1000 commits, it has become the most widely used SDK by Sentry, handling millions of exceptions daily. Its core is reusable with an extendable architecture that serves integrations for almost the entire JS ecosystem.

My responsibilities spanned beyond just writing the SDK code. I also handled its architecture, tooling, deployment setup, and integrations with platforms like React, Angular, Ember, Vue, Next.js, Node.js, Serverless runtimes, and even WASM.

Two years after transitioning to a different team, the Web SDK team, which initially consisted of just me, has expanded into a group of individuals. This team has successfully built upon the foundation I left, delivering an increasing number of features each week with great success.

https://github.com/getsentry/symbolicator

Rust - maintainer

Upon joining the "Processing Team", which is primarily responsible for working on the "Symbolicator," our Rust-based native symbols processor, I had two main tasks. The first was to "make source maps awesome," and the second was to port our age-old Python code from deep within an ancient monolith into a new component of the Symbolicator.

The initial task involved integrating our homemade rust-sourcemap crate into our Rust processing pipeline, introducing Python bindings, and offloading all the heavy lifting from the Python realm. This was a resounding success, which we detailed briefly in our blog post titled "How we made JavaScript stack traces awesome". This success also formed the basis for my recent conference talks on Source Maps.

The second task involved merging this Rust processing pipeline directly with symbolicator, which was used for communication with the main Sentry service. This involved processing incoming events, fetching necessary sources and source maps, and ensuring smooth operation through caching and error tolerance. During this migration, we also began our experiment around "self-identifying" Source Maps files, which you can read about in our blog post "The Case for Debug IDs".

https://github.com/getsentry/sentry-cli

Rust - maintainer

Sentry-cli serves as the primary gateway for all debug file-related tasks, whether it's JS Source Maps, Native Debug Symbols, or Memory Dumps. It's integral across all our products, simplifying the user experience by enabling local file validation, bundling, Source Maps rewriting and resolution, and uploading to our servers. It also handles any API-based tasks that could benefit our customers.

When I took over, the public API needed enhancements, so my main goal was to overhaul the entire user-facing CLI experience, making it more user-friendly while maintaining full backward compatibility. This tool is deeply ingrained not only in our systems but also in our users' systems. Despite the challenge of learning Rust while doing this, it proved worthwhile. Over the next three years, when I joined the "Processing Team" which primarily uses Rust, the benefits of my earlier efforts became evident.

https://github.com/getsentry/sentry-go

Go - creator

Taking on the challenge of learning a new language while creating a brand-new SDK for a language we aimed to branch out to? Count me in! As you might infer from my previous experience, this wasn't my first rodeo. Understanding the journey I was about to embark on, I agreed to develop a Go SDK for Sentry entirely from scratch, despite having no prior experience with the ecosystem and no colleagues familiar with the language. It was certainly an exciting journey. Now, seeing nearly 1k stars on GitHub and the significant traffic we receive from Go environments, I can confidently say it was a rewarding endeavor!

Other, less important, but still fun contributions:

https://github.com/denoland/deno/{deno,deno_std,fresh}

I've also made contributions to a few other Deno repositories. In total, I've made over 50 contributions that span various areas, including tooling, the standard library, the Fresh framework, the JS-runtime API, as well as the Rust core itself.

https://github.com/getsentry/sentry-webpack-plugin/

I developed a Webpack integration for the sentry-cli, which currently boasts nearly 1.3M downloads/week.

https://github.com/getsentry/craft

I've overseen and enhanced our CLI tool that streamlines and automates package releases.

https://github.com/trpc/trpc

As part of the core team, I assist in maintaining the repository and tooling. I also contribute to decision-making processes and help drive the development forward.