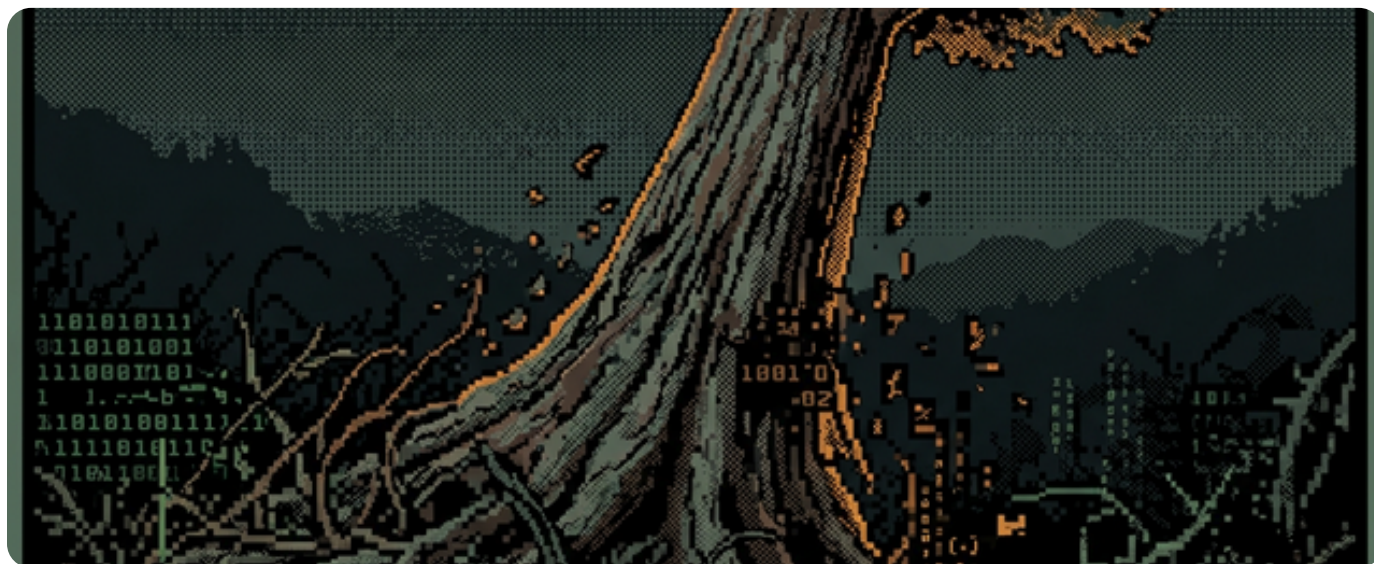


RedwoodSDK 1.0: Getting Out of the Weeds



PETER PISTORIUS • MARCH 11, 2026



Today is the launch of **RedwoodSDK 1.0**.

It has been a five-year journey that started in 2020 with Tom ([@mojombo](#)), Rob ([@cannikin](#)), and David Price ([@thedavidprice](#)). We built a framework in the open, but hidden behind that was my personal path of struggle and existential dread. For every framework out there, there are real people behind it, and for me, it wasn't a straight line. It took leaving the framework to start my own company, failing, and coming back with a completely different perspective on what a framework should actually do.

The Redwood Journey: A Brief Timeline

- **March 10, 2020:** RedwoodJS v0.1 is announced to the world.
- **April 4, 2022:** RedwoodJS v1.0.0 is officially released.
- **March 11, 2026:** Today, the launch of RedwoodSDK 1.0.

The Spark

It all started with a tweet. I was lying in bed and commented "that's cool" on Tom's post about Netlify functions. He slid into my DMs and asked if I wanted to build a framework. At the time, I was already working for him.

As a guy from South Africa who wanted to be part of Silicon Valley his entire life, this felt like total validation. I literally jumped up and down on my bed like a child. I was so excited.

The Lesson: Just Ask

Early on, my focus was split and I wasn't moving fast enough. Tom told me he really wanted the framework to exist and he was thinking of paying someone to work on it full-time. I felt terrible, like I had let him down, and I blurted out: "**Just pay me.**"

The Reality Check

We built RedwoodJS during the pandemic, and it was an incredibly safe space to be productive. But I knew I always wanted to lead. I wanted to push my own ideas out into the world, almost to prove to myself that I could.

I left to become a founder and built Snaplet, my own startup, using Redwood. That's when I realized something embarrassing: I wasn't the right kind of developer for an early-stage startup.

I was so focused on the code, not the business. I spent an enormous amount of time on "nonsense":

- Hosting the database.
- Fighting restrictive Lambda functions.
- Wrestling with Fargate.
- Gluing together infrastructure.

To me, the business was just adjacent to the work. It was the place where I did things, not the reason I was doing them. I had to learn to care about things that weren't code, and it was harder than I thought.

The Developer's Cage

missed the point. He didn't mean coding is worthless; he meant that as developers, our expertise is often our cage.

We get so obsessed with the "how", the elegant plumbing, that we forget the "why." We solve the technical problem the tool was meant for, instead of the human problem the business exists for. I've been that developer. I've been so good at my craft that I coded my way right into failure.

A New Perspective: RedwoodSDK

When I got the opportunity to take over Redwood, I had to be honest about our constraints. With a smaller team and a tighter budget, we couldn't just keep iterating on the original, broad scale. I wanted to build something that reflected success, not just technical elegance.

That is why Justin and I built **RedwoodSDK** around three design principles:

Without Magic

RedwoodSDK avoids all hidden behavior. No code generation. No transpilation side effects. No special treatment of file names or exports. Only explicit import and export statements. Everything respects JavaScript's core contracts.

If the runtime relies on convention instead of clarity, it breaks the language contract. With RedwoodSDK, what you write is what runs.

Composability Over Configuration

RedwoodSDK gives you primitives, not policy. You build from functions,

infrastructure.

You are in control. RedwoodSDK helps you build the software you want without getting in your way.

Web-First Architecture

RedwoodSDK is built for the web as it exists today. It uses native Web APIs — fetch, Request, Response, URL — without wrapping them. If the platform already gives you a tool, we don't abstract it. We help you use it directly and idiomatically.

This philosophy drives every architectural decision in RedwoodSDK. By staying close to the platform, we reduce complexity, remove hidden behavior, and make code easier to understand and maintain. It's not just about writing software — it's about understanding the software you're writing.

RedwoodSDK is designed to stay out of your way so you can focus on the business, not the weeds.

I want to make a special mention of **Justin van der Merwe**, who has tirelessly worked on RedwoodSDK and helped me define the principles that have resulted in something that feels so good. His dedication and craftsmanship are woven into the foundation of what RedwoodSDK is today.

This is my journey. But there are hundreds more people that have [contributed to](#)

ABOUT

[Why Cloudflare?](#)

[Personal Software](#)

[Blog](#)

[Media](#)

[Contributors](#)

[SyntaxFM Interview](#)

RESOURCES

[Quick Start](#)

[Docs](#)

[Cloudflare RedwoodSDK Docs](#)

[Realtime](#)

SOCIAL

 [GitHub](#)

 [Discord](#)

 [X](#)

features. its standard-based router, with support for middleware and interceptors, gives you fine-grained control over every request and response. With built-in access to Cloudflare Workers, D1 (Database), R2 (Storage), Queues, AI, and full local emulation via Miniflare, development feels just like production.

Copyright © 2026 RedwoodJS Inc. All rights reserved.