LESS MORE

JIM KANG: RÉSUMÉ

To view or more or less information, use the slider above.

I'm a software developer in Cambridge, Massachusetts. I've been a software developer since the year 2000. You can reach me at jimkang@fastmail.com.

My current strengths are making full stack web apps and finding ways to iterate quickly. I am working on a future strength: designing interactive explanations.

This is an overview of my work. If you just want to go straight to my publicly available code, here are my Github repos and my NPM packages.

WHAT I'M LOOKING FOR

I WANT TO BUILD PRODUCTS THAT

- Help people understand ideas and situations.
- Get people to imagine possibilities.
- Put power in the hands of users instead of taking it from them. (e.g. Pull instead of push, aid humans instead of replacing them.)
- Get users to think instead of just consume. This can be done without putting onerous burdens on them.

I AM INTERESTED IN

- Art
- Visual explanations
- Procedural generation
- Product design

PROJECTS

Personal projects are important to me. They are a great way to get a sense of my interests and abilities, and unlike some of my commercial work, I can freely talk about all of it.

A few highlights:

- An interactive explanation of quadtrees.

- annoy-node is Node bindings for Annoy, a popular Approximate Nearest Neighbors implementation in C++.

If you want a **complete** view of my projects, look at Observatory. It organizes 300+ projects from a variety of perspectives.

JOBS

WEB AND MOBILE ERA

This part of my career started around 2008 and continues today.

SPOTIFY

I currently work at Spotify.

In 2016-2018, I worked on a team that built music recommenders for users (e.g. the This Is playlists — greatest hits playlists for artists that stay fresh by updating daily based on listening patterns) — and Time Capsule). Now, I work on a team that is building voice experiences.

Some things I've worked on at Spotify:

- An image generator that composes and renders thousands of playlist covers each day based on playlist contents by taking advantage of headless Chrome
- Internal apps (web apps, Chrome extensions, Electron apps) for evaluating and adjusting algorithmically-generated content
- Interaction and content prototypes
- Voice experience prototypes built in Alexa Skills and the browser (using the Web Speech API and DialogFlow)
- External promotional web sites
- Gathering data from users via surveys

My work involves full-stack web engineering, navigating a unique internal infrastructure. It also involves investigating both user needs and technical possibilities.

PAYPAL

I worked at PayPal on the Shop, a system built in Node.

LESS — MORE

I worked at NPR on a station management app and API.