Thai Pangsakulyanont

@dtinth · Bangkok, TH · dtinth@spacet.me · https://dt.in.th/

Work Experience

Taskworld — Bangkok, TH
Frontend Architect — Jun 2015 – Present

- Set up and evolved our end-to-end testing system, from Nightwatch (JavaScript) to RSpec and Capybara (Ruby), to ultimately creating taskworld/prescript, our own test runner written in TypeScript. It allows us to run tests in parallel, generate detailed test reports, and provides tools to help us debug end-to-end tests more easily.
- Implemented a browser automation abstraction layer in our test suite. 2 years later, this abstraction would allow us to migrate our tests from Selenium WebDriver to Playwright with very little change to the test scripts.
- Migrated the frontend application build pipeline from Browserify to webpack, giving us access to modern performance optimizations techniques such as tree-shaking and code splitting, helping us to make our application load faster.
- Replaced a jQuery-based kanban board component with one written from scratch on top of React to support touchscreens and give us more control over the UI and UX.
- Improved application observability in the frontend by integrating with Real User Monitoring (RUM) services to help engineers troubleshoot application errors more easily.
- Improved application observability in the backend by implementing diagnostics logging, structured logging, and distributed tracing. These added logs would become instrumental in providing us vital clues to resolve various issues that happen only on production.
- Evolved frontend application's continuous delivery pipeline based on GitHub Flow, giving engineers enough confidence to deploy frontend code to production environments on demand (i.e. multiple times a day).
- Set up systems to speed up the continuous delivery pipeline by caching build outputs, generating automatic changelogs, and allowing multiple small PRs to be batched up, tested and merged together.
- Created custom ESLint plugins to ensure consistent coding conventions are used across the codebase, saving us time and effort in reviewing PRs, and encouraged other engineers to do the same. taskworld/eslint-plugin-local
- Worked on building product features and fixing bugs while remaining focused on long-term maintainability and developer experience on the codebase. Also built some quick hacks (that are easy to delete!) in response to business needs, such as prototyping an integration for Google Meet to improve collaboration during COVID-19.
- Gave a few talks based on my working experience at Taskworld: "Smells in React Apps", "Race Conditions in JavaScript Apps", and "Embracing gradual typing".

OOZOU — Bangkok, TH
Intern — Aug 2014 – Dec 2014

- Worked on building products, fixing bugs, converting designs to code, delivering value. You know, the usual software developer stuff.
- Set up Jasmine test framework for a Ruby on Rails project that lacks frontend testing.
- Improved testing experience in JavaScript by building the oozou/stateful-context library.
- Investigated and fixed performance bottlenecks in Ruby using flamegraphs, creating the oozou/ruby-prof-flamegraph gem in the process.
- Wrote several articles on OOZOU blog: https://oozou.com/blog/authors/thai

Education

Kasetsart University
Software and Knowledge Engineering — Jun 2011 – May 2015

For more software projects, talks, and songs, please check out my website at https://dt.in.th/.

The following pages are about what I do outside of work.
They are rather long but in case you're interested…

Community projects

React Bangkok
A series of meetups and conferences focused on React.

- Gave a few talks: "HOCs and Recompose", "Live coding and mob programming session"
- Helped in organizing the events, mainly tasks that require coding.

JavaScript Bangkok
An international conference about JavaScript.

- Helped in architecting and remotely collaborating on the conference website using Figma and CodeSandbox. The process of building the website is livestreamed and I posted a video to document our synchronous remote collaboration on YouTube.
- Implemented lunchtime meal reservation system using Firebase, ensuring that the system can handle 800 people reserving their meal at the same time, keeping all data consistent.
- Live-translated Thai talks into English during the event.

The Stupid Hackathon Thailand
A series of hackathons to encourage people to just have fun building stuff. It's one of the funniest hackathons in Thailand. Helped in organizing the 1st to 4th hackathons. For the 5th hackathon, I was a consultant for the organizing team.

- Gave talks about the art of rapid prototyping and provided technical mentorship to the participants (1st).
- Devised schemes to award hackathon prizes to participants in a non-competitive way (2nd-4th).
- Devised a series of trivia challenges that one has to solve in order to get a ticket to the hackathon due to high interest but limited venue capacity (2nd–3rd).
- Created an art direction and produced graphic assets for the hackathons (3rd–4th).
- Went virtual due to COVID-19. Set up virtual spaces and planned out activities during the hackathon (4th).
- Helped in organizing other events by the same team, such as Hacktoberfest Bangkok meetup 2020 and BKK.JS #14.
- Made a few projects myself: Bangkok Ipsum, MIDI Light Switch and Misheard.

Code in the Dark Thailand
An challenging frontend contest event by Tictail. For more info see codeinthedark.com. Being an open-sourced event, I helped bringing it to Thailand. We ran 3 events in Bangkok and Chiang Mai.

- Came up with web design implementation challenges for the participants (1st, 2nd).
- Devised simple challenges that one has to solve in order to get a ticket to the hackathon due to high interest but limited venue capacity (1st).
- Came up with a frontend quiz for the audience. The quiz result is used to select who would become a contestant. This replaced the first-come-first-served sign-up system which I believe prefers confidence over skills (3rd).

ELECT LIVE
An open-source website by the ELECT.in.th initiative that shows the live results of 2019 Thailand Election. I volunteered as a frontend architect.

- Helped in building the technical architecture of the website using React, Gatsby, Emotion and MobX. The architecture provided tools for developers to inspect the application state and raw data from within the app, allowing for easier debugging.

- Used the Puzzle-Driven Development methodology to break down programming tasks into smaller and easier tasks which can be independently worked on. This eventually allowed over 20 people to contribute to the website, closing over 100 issues in just 8 days.
- Documented our learnings in ELECT LIVE development notes (Thai) and on the electinth/election-live repository's README file (English).

wonderful.software Webring
A webring for Thai developers, designers, and artists. Each site in the ring links to the main webring site. The webring then takes the visitor to the next site in the ring, allowing traffic to flow between members' sites. See webring.wonderful.software.

- Built the webring using HTML, CSS and Vue.js using no build tooling. This lets people add their site to the project without having to install anything.
- Created an end-to-end test suite for the website using Cypress. As a website that I rarely maintain, I don't want to spend much time manually testing it when making changes.
- Created a semi-automated PR review workflow using GitHub Actions. It automatically visits the website and checks for a valid backlink, saving me time when reviewing PRs.
- Set up automation to continually update the website screenshots in the webring using Puppeteer.
- Set up system to collect and publish webring usage stats using Amplitude, Google Cloud Functions, and GitHub Actions.

Clubhouse Music Jamming Room
A community of people who like to jam together online using the open-source Jamulus software. We used Clubhouse to broadcast our jam sessions live and find new members.

- Set up and maintained the Jamulus servers for the group.
- Managed a fleet of VPS servers using Ansible as we make our servers public and adding more servers to our fleet. Created tools to centrally manage the servers using PHP (on the C&C server) and Python (on each Jamulus server).
- Set up server monitoring using Azure Monitor.
- Evaluated cloud service provider performance by signing up for a bunch of cloud services and putting them through a series of benchmarks (audio encoding test, packet delay variation (jitter) test). The results would then be analyzed to find out which cloud service provider provides the best service with reasonable price.
- Contributed a patch to introduce JSON-RPC to the Jamulus project as a way to programmatically control the Jamulus servers and clients.

Personal projects

Personal digital infrastructure project
A personal project to build tools that lets me be more lazy.

- Created a serverless rapid prototyping platform on Google Cloud Run dtinth/evalaas. It lets me prototype and make changes to the chat bot and see the results immediately, without having to go through a full deployment cycle.
- Set up a multi-channel chat bot (LINE, Slack, CLI) that automates my home, tracks my expenses, lets me take notes, and make quick calculations: dtinth/automatron.
- Built an Android app to send notifications from my phone to automatron, so that financial transactions can be automatically tracked: dtinth/dtinth.tools-android.
- Set up an automated script to periodically take screenshots of my web projects: dtinth/timelapse.
- Set up a system that lets me quickly take notes in Markdown from inside VS Code, and share them as a webpage: dtinth/notes.dt.in.th.
- Set up an Antora-based multi-repository documentation site that lets me document my projects without having to create a separate documentation site for each one: dtinth/docs.

Technologies

This is an alphabetical list of technologies, tools, and software that I have some experience with, sourced from my past projects:

Airtable API, Allure Framework, Amazon Lightsail, Amazon S3, Amplitude, AngularJS, Antora, API Documenter, API Extractor, AsciiDoc, Auth0, AWS Cloud9, Axios, Azure Blob Storage, Azure Monitor, Babel, Browserify, Canvas API, Chai, Chrome Extension, CircleCI, Cloudflare CDN, CodeSandbox, CoffeeScript, CSS, Cucumber, curl, Cypress, Datadog, Debian, DeviceOrientation API, DigitalOcean App Platform, Discord.js, Docker, Docker Compose, Elasticsearch, Electron, Emotion, ESLint, execa, Express, Facebook Graph API, FFmpeg, Figma, Firebase Authentication, Firebase Cloud Firestore, Firebase Hosting, Firebase Realtime Database, Fish Shell, getDisplayMedia API, getUserMedia API, Git, GitHub Actions, GitHub API, GitHub Codespaces, GitHub Packages, GitHub Pages, Glitch.com, Google APIs, Google Artifact Registry, Google Cloud IoT Core, Google Cloud Run, Google Cloud Speech API, Google Cloud Storage, Google Container Registry, Google Drive API, Google Kubernetes Engine, Google Sheets API, Google Sign-In, Gulp, Handlebars, Heroku, Hot Module Replacement, HTML Custom Elements, Huawei Cloud ECS, Icecast2, Immutable.js, JavaScript, Jekyll, Jest, jQuery, JSON-RPC, JWT, JXA, Karma, Keynote, Less CSS, LINE Bot, Linode, Linux, Lodash, Markdown, MobX, Mocha, MongoDB, MongoDB Atlas, n-gram Language Modeling, ncc, Netlify, Netlify Functions, Next.js, Node.js, Notion, npm, Nuxt.js, OAuth2, Octokit, PHP, Pino, pnpm, postMessage API, PouchDB, Prescript, Prettier, Puppeteer, PWA, Python, React, Redux, Reselect, RobotJS, Ruby, RxJS, SCSS, Selenium WebDriver, Service Workers, Shell Script, Slack API, Socket.io, Styled Components, Stylus, Svelte, SVG, Tailwind CSS, Test Automation, Three.js, tmux, TweetNaCl, Twind, Twitter API, TypeScript, Ubuntu, UMD, Userscript, Vercel, Vim, Visual Studio Code, Vite, VS Code Extension API, Vue.js, VuePress, Web Audio API, Web Gamepad API, Web MIDI API, Web Share Target API, Web Speech API, webpack, WebRTC (simple-peer), WebSocket, YAML, Yargs, Yarn, Yjs.

The following technologies are also mentioned in my GitHub repositories, but I haven't used them enough (or haven't used them for too long — many technologies on this list are already dead):

Adobe AIR, Android SDK, Apollo Server, AppleScript, Applitools, Atom, AudioWorklet, AutoIt, AutoPy, Batch Script, Blurhash, Bower, C, C++, Caddy Server, ClojureScript, Cloudflare Workers, Cocoa, CodeMirror, Docsify, Docusaurus, Fastify, Firefox Extension API, Flutter, FMOD, Gauge, GLSL, Google App Engine, GraphQL, Hammerspoon, Haxe, Heft, IPFS, Jasmine, Java, Jupyter Notebook, Kotlin, Kubernetes, Lerna, LevelDB, libsndfile, Lua, Metalsmith, Music Theory, OAuth 1.0a, Objective-C, OpenAI API, opentype.js, Opus, Pandoc, Philips Hue API, PostCSS, Preact, PulseAudio, Qt, Redux Toolkit, RefluxJS, remark, RequireJS, Rust, Serde, SonarCloud, SpiderMonkey, Stencil.js, StratifiedJS, Surge.sh, Swift, Tokio, Tornado Web Server, Travis CI, unist, Visual Basic .NET, Visual Basic 6, Vuetify, Web Share API, WebAssembly, WebGL, WebRTC, Wintersmith, x86 Assembly, XMPP, Xterm.js, XULRunner, Xvfb, YouTube Player API, YQL, ZeroMQ.