

Modern Frontend Toolchain ✨

By Lee Zong Xun

Who am I?

I am a senior [at] NUS, studying computer science and statistics. I have interned at Tiktok, Jupyterlab, Expensify and CVWO. I am also the Vice President of NUS GDSC and the Program Lead of SoC TIPS.



Let's dive into the toolchain

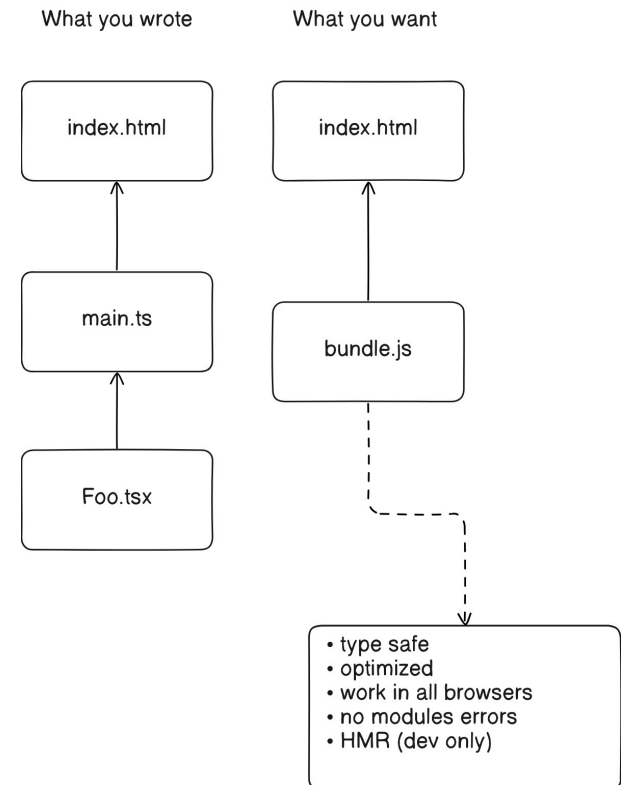
How Vite and other development environments work

Two "Major" Parts

- `npm run dev` - development
- `npm run build` - build for production

Part 1: We need JavaScript (that works in the browser)

- TSX
 - Remove types (JSX left)
 - Change JSX in JS
 - Change JS to other JS
 - Transform `node_modules` imports to relative paths with extensions



Part 2: We need a good DX

- HMR
- Source maps
- It should be fast

Part 3: We need a good prod PX

- bundle should be small
- remove unused code
- type check, be confident in your app

Brief Overview

[Link to diagram](#)

Hot Module Reload

Goal: a good frontend toolchain will make HMR "free"

User doesn't need to do anything special - just save their file, and see the updates!

[Link to diagram](#)

Server

- Express static server
- HMR middleware
 - This is the "dev server"
 - Intercept *.js , inject HMR "client"

Client

- `HotModule` class implementation
- `hmrClient` function to "initialize" the HMR and connect to the web socket
- "register" the accept callback
 - this is what to do when the module is saved/changed. Eg how to HMR for this specific module type
- websocket to respond to Dev Server events (connectWs)

Dev Server

- WebSocket server
- file watcher
- notify HMR client of file changes
- trigger reload of module

Credits to Lachlan Miller 🦾