Enterprise Marketplace

Cheatsheets for experienced React developers getting started with TypeScript

Basic | Advanced | Migrating | HOC | 中文翻译 | Contribute! | Ask!

Migrating (to TypeScript) Cheatsheet

This Cheatsheet collates advice and utilities from real case studies of teams moving significant codebases from plain JS or Flow over to TypeScript. It makes no attempt to *convince* people to do so, but we do collect what few statistics companies offer up after their conversion experience.

> ⚠️ This Cheatsheet is extremely new and could use all the help we can get. Solid advice, results, and up to date content all welcome.

Prerequsite

Read TypeScript's official Guide for migrating from JS and you should already be familiar with their React conversion guide.

General Conversion approaches

- Level 0: Don't use TypeScript, use JSDoc
  - See our JSDoc section
- Level 1A: Majority JavaScript, increasingly strict TypeScript
  - as recommended by official TS guide
  - use `allowJS` (Experiences: clayallsop, pleo)
- Level 1B: Total rename to TypeScript from the start
  - "Just rename all .js files to .ts"?
  - use the loosest, bare minimum settings to start with
- Level 2: Strict TypeScript
  - use Microsoft's `dts-gen` to generate `.d.ts` files for your untyped files. This SO answer has more on the topic.
  - use `declare` keyword for ambient declarations - see declaration merging to patch library declarations inline

Misc tips/approaches successful companies have taken

- `@ts-ignore` on compiler errors for libraries with no typedefs
- pick ESLint over TSLint (source: ESLint and TS Roadmap)
- New code must always be written in TypeScript. No exceptions. For existing code: If your task requires you to change JavaScript code, you need to rewrite it. (Source: Hootsuite)

▶ Webpack tips

JSDoc

- https://github.com/Microsoft/TypeScript/wiki/JsDoc-support-in-JavaScript
- webpack's codebase uses JSDoc with linting by TS https://twitter.com/TheLarkInn/status/984479953927327744 (some crazy hack: https://twitter.com/thelarkinn/status/996475530944823296)

Problems to be aware of:

- `object` is converted to `any` for some reason.
- If you have an error in the jsdoc, you get no warning/error. TS just silently doesn't type annotate the function.
- casting can be verbose

(*thanks Gil Tayar and Gleb Bahmutov for sharing above commentary*)

From JS

the "Just Renaming" strategy

- OSX/Linux: `find src -name "*.js" -exec sh -c 'mv"$0" "${0%.js}.tsx"' {} \;`

More resources

- Migrating a `create-react-app` / `react-scripts` app to TypeScript - don't use `react-scripts-ts`
- Migrating an EJECTED CRA app to TS
- Lyft's JS to TS migration tool (includes PropTypes migration)
- Hootsuite
- Storybook's migration (PR)

- How we migrated a 200K+ LOC project to TypeScript and survived to tell the story - Coherent Labs - using `grunt-ts`, jQuery and Kendo UI

Old content that is possibly out of date

- Incrementally Migrating JS to TS (old)
- Microsoft's TypeScript React Conversion Guide (old)

From Flow

- Try flow2ts: `npx flow2ts` - doesn't work 100% but saves some time (see this and other tips from @braposo at TravelRepublic)
- Incremental Migration to TypeScript on a Flowtype codebase at Entria
- Jest's migration (PR)
- Expo's migration (issue)
- Atlassian's migration (PR)
- Yarn's migration (issue)
- MemSQL's Studio's migration - blogpost with many useful tips

Results

- Number of production deploys doubled for Hootsuite
- Found accidental globals for Tiny
- Found incorrect function calls for Tiny
- Found rarely used, buggy code that was untested for Tiny

Misc writeups by notable companies

- Lyft
- Google
- Tiny - Talk from ForwardJS here
- Slack

Links