



Lessons Learnt While Creating a New Framework on Top of React

React Advanced London, 2022

Shyam Swaroop

Co-Founder & CTO, Atri Labs



Shyam Swaroop



- **Creator of Atri engine, and
Co-Founder & CTO of Atri Labs**

Creating a full-stack web development framework

- **Prev: Graduate Research Assistant at
Columbia University & Software Engineer
at EXL Services**

JSX goes beyond web development

Used in backend for HTML generation

- Nothing different than SSG/SSR

Used as replacement for template engines

Some projects that take JSX beyond JS/TS

- gox - <https://github.com/8byt/gox>
- PyReact - <https://github.com/facebookarchive/react-python>

React goes beyond just frontend development

React for writing content in .mdx files

- This enables us to create frameworks specialized for documentation such as Docusaurus

React can render on any device

- Use react-reconciler to adapt to new devices

React's computational model can be adapted

- Synthesise music -
<https://github.com/FormidableLabs/react-music>

Frameworks

innovate and differentiate in
their offering for developers

Journey of developing a framework



Tools

- Typescript **Compiler** (tsc)
- **Linters** like eslint
- **Bundlers** for generating final assets (webpack, swc, esbuild)
- **Transpilers** for codemods (Babel)
- **Formatters** for prettifying the code (prettier)

Javascript Syntax

- Javascript has a very rapidly changing spec for syntax
- It has many module systems such as AMD, UMD, CommonJS, and recently the browser-supported module type

Decide your bundler

- Toughest decision
- Requirements of the framework, the skillset of the team, etc
 - For fast build - use esbuild, swc
 - For advanced codemods - use build tool that exposes AST. Babel does, esbuild does not.
 - esbuild ignores type in typescript
 - Might have to learn Rust or Go

Atri Framework uses Webpack with Babel

- Resources such as manifest schemas, manifest, and plugins refer to each other
- For example, a manifest has to refer to a manifest schema for runtime type checking
- Improved developer experience
 - Using webpack's resource query feature

```
const reactSchemaId = "@atrilabs/react-component-manifest-schema";
```

```
import reactSchemaId from "@atrilabs/react-component-manifest-schema?id";
```

Where do we start?

create-react-app repository

The best starting point

NextJS repository

Not necessary, but a good resource

Packages in create-react-app

Packages for building CRA

- react-scripts
- react-dev-utils

Packages for configuring tools

- babel-preset-react-app
- babel-plugin-named-asset-import
- eslint-config-react-app

Packages for generating template

- create-react-app
- cra-template
- cra-template-typescript

react-scripts/lib/react-app.d.ts

<https://github.com/facebook/create-react-app/blob/main/packages/react-scripts/lib/react-app.d.ts#L47-L56>

@atrilabs/layer-types

<https://github.com/Atri-Labs/atrilabs-engine/blob/main/packages/layer-types/src/index.d.ts#L76-L79>

react-scripts/configs

<https://github.com/facebook/create-react-app/blob/main/packages/react-scripts/config/webpack.config.js#L119-L178>

react-scripts/configs

<https://github.com/facebook/create-react-app/blob/main/packages/react-scripts/config/webpack.config.js#L179-L195>

react-scripts/configs

<https://github.com/facebook/create-react-app/blob/main/packages/react-scripts/config/webpack.config.js#L388-L413>

react-scripts/configs

<https://github.com/facebook/create-react-app/blob/main/packages/react-scripts/config/webpack.config.js#L528>

Dead Code Elimination & Tree Shaking

Available by default in production mode

- Otherwise, we have to add plugins such as `ModuleConcatenationPlugin`, `TerserPlugin` on our own.

Tree Shaking

- Wikipedia - Tree shaking eliminates unused functions from across the bundle by starting at the entry point and only including functions that may be executed.
- React's Higher Order Components (HOC) make it harder for terser to tree shake dead code.

Turn Off Tree Shaking for CSS

- Set `sideEffects` to `true` to prevent elimination.

Dead Code Elimination & Tree Shaking

Requirements (from Webpack docs)

- Use ES2015 module syntax (i.e. import and export).
- Ensure no compilers transform your ES2015 module syntax into CommonJS modules (this is the default behavior of the popular Babel preset `@babel/preset-env` - see the documentation for more details).

<https://github.com/facebook/create-react-app/blob/main/packages/react-scripts/config/webpack.config.js#L528>

index.jsx

```
import { WrappedButton } from "../WrappedButton";
```

WrappedButton.jsx

```
import hoistStatics from "hoist-non-react-statics"

function Button() {}

function ButtonWrapper(button) {
  return hoistStatics(button);
}

const WrappedButton = ButtonWrapper(Button());

export { WrappedButton };
```

WrappedButton.jsx

```
import hoistStatics from "hoist-non-react-statics";  
  
function Button() {}  
  
function ButtonWrapper(button) {  
  return hoistStatics(button);  
}  
  
const WrappedButton = /*#__PURE__*/ ButtonWrapper(Button());  
  
export { WrappedButton };
```

package.json

```
"sideEffects": false
```


Hot Module Replacement

Replace an old module with a new one during runtime

How do we maintain the state of the component after the module containing this component has been replaced?

React Fast Refresh @pmmmmwh/react-refresh-webpack-plugin

How to decide whether the changes made in the module should trigger a full page reload or hot module replacement?

- All the exports must be React Components

How do we do HMR for CSS and other similar files?

- During development, replace CSS files with JS modules.
- Plugins/loaders are available for different bundlers that take care of this — style-loader for Webpack.

Open source full stack framework

<https://github.com/Atri-Labs/atrilabs-engine>



Thanks for listening!