

# ブラウザ環境

今も昔も、ウェブフロントエンドが今のJavaScript/TypeScriptの主戦場です。本章ではそのウェブフロントエンドの環境について紹介します。

現在、メジャーなウェブフロントエンドのフレームワークというと、React、Vue.js、Angularです。本書で取り上げるのはReactとVue.jsです。React以外にReactをベースにした統合フロントエンドフレームワークとなっているNext.jsも取り上げます。Angularは2以降からTypeScriptに書き直されて、TypeScript以外では書けなくなり、最初からTypeScriptが有効な状態でプロジェクトが作成されるため、説明は割愛します。

本章では、それぞれの環境の共通部分について紹介します。

## ウェブアプリケーションにおけるビルドツールのゴール

JavaScriptでビルドといっても、いろいろなステップがあります。

1. TypeScriptやBabelを使って、ターゲットとなるバージョンのJavaScriptに変換
2. SCSSとかPostCSSを使ってブラウザにない機能を使って書かれたCSSを素のCSSに変換
3. webpackなどを使って、1つのJavaScriptファイル、もしくは遅延ロードをするJSファイル群を生成

実際には綺麗にステップが分かれることはなくて、バンドラーがimport文を追跡しつつファイルを探し、.tsを見つけてはTypeScriptで処理して（コンパイル）、コード中にSCSSを見つけてはSCSSの処理系に投げて、一つのファイルにまとめる（バンドル）・・・みたいな工程を行ったりきたりしながらビルドします。以前は、これにJake、Gulp、Gruntなどのタスクランナーも組み合わせやっていたんですが、今はwebpack単体にts-loaderなどを組み合わせる感じで一通りできます。webpackがシェア80%で一強です。

しかし、ファイルを集めてきてコンパイルしつつ1ファイルにバンドルし、必要に応じてminify化してサイズを小さくするといったタスクは、比較的重い処理です。そのため、ウェブのフロントエンドのビルド環境は、開発者のストレスを軽減するために、さまざまな機能を備えてきました。

- バンドルツールは毎回ファイルを読み込むのではなく、メモリに常駐し、変更されたファイルだけすばやく読み込んでバンドルを完成させる
- ブラウザに対して、変更検知を伝えるコードを差し込んでおくことで、すばやくブラウザをリロードして最新のファイルを読み込む仕組み

これらを実現するのが開発サーバです。開発サーバーはHTTPサーバーとして起動し、JavaScriptやHTML、CSSを配信するサーバーとしてブラウザからは見えます。その裏では、ファイルシステムのソースコードを監視し、変更があったら即座にビルドをして動作確認までのリードタイムを

短くします。それだけではなく、その開発中のウェブサイトを見ているブラウザに対して強制リセットをしかけたり（ホットリロード）、起動中にJavaScriptのコードを差し替えたり（ホットモジュールリプレースメント）といったことを実現します。また、TypeScriptだけではなく、CSSでも、事前コンパイルでコーディングの効率をあげる方法が一般化しているため、この設定も必要でしょう。

この分野では数多くのツールがあります。スキルのある人は自分の好みや要件に合わせてツールを選択すると良いでしょう。TypeScriptの対応についても、最初から対応していたり、後からプラグインで拡張など、いろいろなものがあります。

webpackは細かくカスタマイズできますし、豊富な開発リソースで頻繁にリリースされています。ツリーシェイキングといった不要なコードを除外してサイズを小さくする機能にいち早く取り組んだり、業界をリードしています。困った時に検索すると情報も多く出てきます。一方で、TypeScript対応で開発サーバーやCSS対応など、機能を足していこうとすると設定やプラグインが増えていきます。特に、ReactのJSX構文を利用する場合は、バンドラーの処理の前段でTypeScriptをJavaScriptに変換したあとにBabelを使い、最後にバンドラーで1ファイルや複数ファイルにまとめるなど、ビルドのパイプラインが多段になりがちです。ReactやVueの環境構築ツールやNext.js、Nuxt.jsなどはwebpackを内包して、少ない設定の量で一定の機能を備えたビルド環境を整えてくれます。本書でも、webpackそのものを紹介することはしませんが、これらのツールの紹介をします。

他にも数多くのバンドラーがあります。webpack以降に出てきたものの多くは設定が少ない、あるいは設定ファイルが不要（ゼロコンフィギュレーション）を売りにしたものが数多くあります。Rollupは人気のあるバンドラーで、TypeScriptを使うにはプラグインが必要ですが、そうでないのであれば設定がほとんど必要ありません。RollupをベースにTypeScriptサポートを最初から組み込んだmicrobundleもあります。HTMLやCSSのビルドもできて開発サーバーも全てついてくるオールインワンでビルド速度を重視したParcelや、Go製でビルド速度に特化したesbuildもあります。一方で、カスタマイズが必要なので最初からカスタマイズ前提でCLIを提供しないFuseboxなどもあります。