Web Application Development

Front-End Development with Node.js

KIEU Quoc Viet HUYNH Vinh Nam

Information and Communication Technology Laboratory (ICTLab),
University of Science and Technology of Hanoi

Hanoi, Sept 2024

Table of Contents

Introduction

What is Front-End Development?

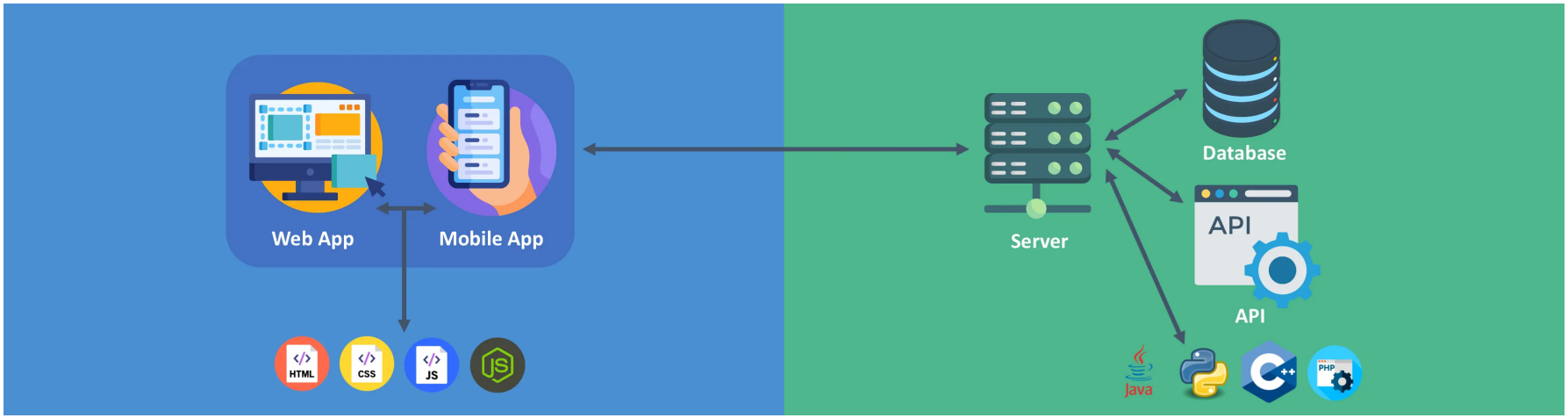
Front-End Development:

- Front-end refers to the part of a web application that users interact with directly.
- Typically built using HTML, CSS, and JavaScript.
- Focuses on user experience (UX) and user interface (UI) design.

Front-End Example

Front-End vs. Back-End Development

Front-End Development	Back-End Development
User-facing part of the app	Server-side logic and databases
Built with HTML, CSS, JS	Built with Node.js, Python, etc.
Handles UI/UX	Handles business logic



Node.js: Used for both back-end and front-end tooling.

FE Dev Tools

Recall: Role of Node.js in Front-End Development

Why Use Node.js for Front-End?

- **Build Tools:** Automate tasks like bundling and minification.
- **Package Management:** Install and manage front-end libraries using npm.
- **JavaScript Everywhere:** Unified language across front-end and back-end.

Recall: Node.js as a Build Tool

Common Build Tools:

- **Webpack:** Bundles JavaScript and assets.
- **Parcel:** Zero-config bundler for fast setup.
- **Gulp:** Automates repetitive tasks like minifying CSS/JS.

Why Bundle? Optimizes performance by reducing file size and load times.

Recall: npm (**N**ode **P**ackage **M**anager)

npm Overview:

- **npm** is the default package manager for Node.js.
- Used to install, manage, and version-control front-end and back-end libraries.

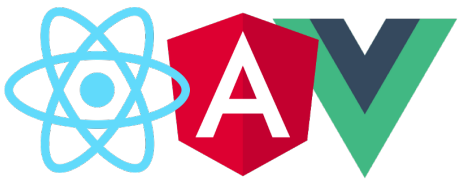
npm Command Examples:

```
npm init  // Initialize a new project
npm install <package>  // Install a package
npm start  // Start the project
```

Common FE Libraries

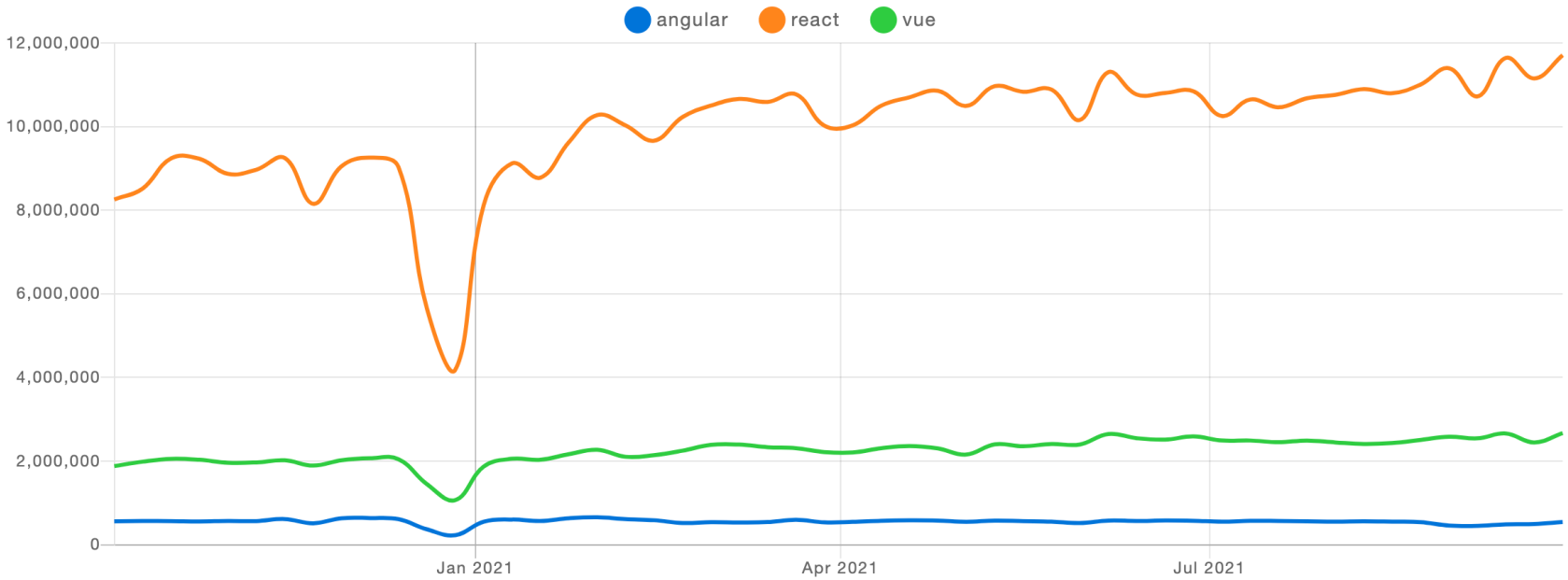
Some Common Front-End Libraries:

- **React**: A powerful and flexible JavaScript library developed by Facebook.
- **Vue.js**: A lightweight, progressive JavaScript framework designed to be easy to integrate into existing projects.
- **Angular**: A comprehensive JavaScript framework developed by Google, designed for building complex and large-scale web applications.



Common FE Libraries (cont.)

Statistics:



Installing Front-End Libraries with npm

How to Install Front-End Libraries:

- npm allows easy installation of JavaScript libraries like React, Vue.js, or Angular.

Example: Installing React:

```
npm install react react-dom
```

Note: The `package.json` file keeps track of installed packages.

Hand-on Some Sample Projects

Project Structure Overview

Typical Front-End Project Structure:

- `src/`: Contains your source files (HTML, CSS, JS).
- `dist/`: Contains bundled files for production.
- `node_modules/`: Installed npm packages.
- `package.json`: Project metadata and dependencies.

Command to Create Project Structure:

```
npm init  // Creates package.json
mkdir src dist  // Create directories
```

Creating a Basic Front-End with Node.js Tools

HTML and JavaScript Setup:

- Create an HTML file (`index.html`) inside `src/`.
- Create a JavaScript file (`app.js`) inside `src/`.

Example HTML File:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4      <title>Node.js Front-End</title>
5      </head>
6      <body>
7      <h1>Hello, Front-End with Node.js!</h1>
8      <script src="app.js"></script>
9      </body>
10 </html>
```

Using Webpack to Bundle Front-End Code

Setting Up Webpack:

- Install Webpack using npm:

```
npm install webpack webpack-cli --save-dev
```

- Create a `webpack.config.js` file to define entry points and output files.

Example Webpack Configuration:

```
module.exports = {
    entry: './src/app.js',
    output: {
        filename: 'bundle.js',
        path: __dirname + '/dist'
    }
};
```

Running and Testing the Front-End

Steps to Run the Front-End:

1. Bundle the project using Webpack:

```
npm run build
```

2. Open the `index.html` file in a browser to see the output.

Testing and Debugging Tips:

- Use the browser's developer tools (Console, Network) for debugging.

React Overview

What is React?

- A JavaScript library for building user interfaces.
- Developed by Facebook, widely used for single-page applications (SPAs).
- Component-based architecture: Reusable, independent UI components.

Setting Up React with Node.js

Installing React via npm:

```
npm install react react-dom
```

Basic React Component:

```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3
4  function App() {
5      return <h1>Hello, React!</h1>;
6  }
7
8  ReactDOM.render(<App />, document.getElementById('root'));
```

Integrating React with Webpack

Webpack Configuration for React:

```
module.exports = {
    entry: './src/index.js',
    output: {
        filename: 'bundle.js',
        path: __dirname + '/dist'
    },
    module: {
        rules: [
        { test: /\.js$/, use: 'babel-loader' }
        ]
    }
};
```

Babel: Used to transpile modern JavaScript (ES6+) and JSX into browser-compatible JS.

Deploying the Front-End Application

Deployment Strategies:

- Host on platforms like Netlify, Vercel, or GitHub Pages.
- Serve as static files from a back-end server (e.g., using Express.js).

Production Build:

```
npm run build  // Creates optimized production build
```

Deploy the contents of the `dist/` directory.

Recap of Node.js for Front-End

Key Points:

- Node.js is crucial for managing and building front-end assets.
- Tools like npm, Webpack, and Gulp automate front-end tasks.
- Front-end libraries like React integrate easily with Node.js.

Question and Next Steps

Discussion

Any Questions?

Next Steps: Front-End Development

- Review the code examples we covered in class.
- Explore more advanced front-end frameworks (Vue.js, Angular).
- Learn state management with React (Redux, Context API).
- Experiment with modern JavaScript features and CSS preprocessors (Sass).

Midterm Announcement

- Choose your group (6 people maximum) in the same class page [Deadline: October 4th, 2024]
- The topics will be assigned randomly, visit this google spreedsheet `https://docs.google.com/spreadsheets/d/1X3gZ2CB0LDxNv_hprwek1V9XVuRaa9V5gdHrVC8n3iM/edit?usp=sharing` to see your topic
  - Visit `https://github.com/VietKQ-usth/WebDev2024` and folk this repo.
  - Every team member has to commit your work.

Midterm Announcement (cont.)

- Requirements:
  - Friendly UI design, understandable, not to complex
  - Fancy UI effects/animations (bonus)
  - Should contain all the pages to serve all use cases from the assigned topic
- Prepare for a presentation (5 minutes per group maximum).

Thank you for listening!