**Frontend Project Package Documentation**

**Package.json**

In this documentation, we will provide an overview of the packages used in the frontend project. The package.json file contains information about the project's dependencies and scripts used for development, building, and testing.

**Dependencies:**

1. **@reduxjs/toolkit** (^1.9.5): Redux Toolkit is used for state management in the application. It provides utilities to simplify the usage of Redux and manage the application's state.
2. **@types/node** (^20.3.1): These are TypeScript type definitions for Node.js. They allow TypeScript to understand and provide type information for Node.js modules and features.
3. **@types/react** (^18.2.12): TypeScript type definitions for React. They enable TypeScript to provide type checking and better developer tooling when working with React components.
4. **@types/react-dom** (^18.2.5): TypeScript type definitions for ReactDOM. These definitions enhance TypeScript support when working with the ReactDOM package.
5. **axios** (^1.4.0): Axios is used for making HTTP requests to APIs. It simplifies the process of handling async data fetching and error handling.
6. **bootstrap** (^5.3.0): Bootstrap is a popular CSS framework that provides ready-to-use styles and components for building responsive web applications.
7. **chart.js** (^4.3.0): Chart.js is used for creating interactive and visually appealing charts and graphs on the website.
8. **devextreme** (23.1.3) and **devextreme-react** (23.1.3): DevExtreme is a set of responsive UI components for building data-intensive web applications. The React bindings (devextreme-react) allow seamless integration with the React ecosystem.
9. **env-cmd** (^10.1.0): This package is used for setting environment-specific variables when running scripts. It allows us to define different environment configurations for development, testing, and production.
10. **formik** (^2.4.2): Formik is a form library for React that simplifies form handling and validation.
11. **i18next** (^23.2.11): i18next is used for internationalization (i18n) support, allowing the application to support multiple languages.
12. **i18next-browser-languagedetector** (^7.1.0): This package enables automatic language detection based on the user's browser settings.
13. **i18next-http-backend** (^2.2.1): i18next HTTP backend is used to load translation resources asynchronously over HTTP.
14. **jwt-decode** (^3.1.2): jwt-decode is used to decode JSON Web Tokens (JWT) on the client-side.
15. **primeflex** (^3.3.1), **primeicons** (^6.0.1), and **primereact** (^9.5.0): PrimeReact is a collection of rich UI components for React applications. Primeflex provides a responsive grid system, and primeicons includes icon sets.
16. **react** (^18.2.0) and **react-dom** (^18.2.0): React is the core library for building user interfaces, and react-dom is used for rendering React components into the DOM.
17. **react-hook-form** (^7.44.3): react-hook-form is an alternative form library for React, providing form handling and validation capabilities.
18. **react-i18next** (^13.0.2): react-i18next is a React wrapper for i18next, facilitating the usage of internationalization in React components.
19. **react-redux** (^8.1.0): react-redux is used to connect React components with the Redux store, enabling components to access and dispatch actions.
20. **react-router-dom** (^6.13.0): react-router-dom is used for client-side routing in the application, enabling navigation between different views.
21. **react-scripts** (^5.0.1): react-scripts is a set of pre-configured scripts for Create React App, providing out-of-the-box development and build workflows.
22. **react-secure-storage** (^1.2.2): react-secure-storage provides a secure way to store data in the client-side local storage.
23. **reactstrap** (^9.2.0): Reactstrap is a set of React components that are built on top of Bootstrap, offering reusable UI elements.
24. **redux-persist** (^6.0.0): redux-persist is used to persist and rehydrate the Redux store across page reloads.
25. **sass** (^1.64.1): Sass is a CSS preprocessor used to write more organized and maintainable CSS code.
26. **typescript** (^5.1.3): TypeScript is a typed superset of JavaScript that provides static typing and other developer tools for large-scale projects.
27. **web-vitals** (^3.3.2): web-vitals is used to measure key performance metrics of the website.

**Scripts:**

The package.json file also includes various scripts that facilitate different tasks during development, building, and testing:

* **start**: Runs the development server for the application.
* **build**: Creates a production build of the application.
* **test**: Executes tests for the application.
* **eject**: Ejects the application from Create React App, allowing more customization but irreversible.
* **build:dev**: Builds the application using the development environment configuration.
* **build:test**: Builds the application using the test environment configuration.
* **build:prod**: Builds the application using the production environment configuration.
* **build:scss**: Compiles and builds the SCSS files for the light and dark themes.
* **build:Watchscss**: Watches the SCSS files and automatically compiles them when changes are detected.

**Eslint Configuration:**

The **eslintConfig** section extends the "react-app" and "react-app/jest" configurations, which provide linting rules and settings for React applications.

**Browserslist:**

The **browserslist** section defines the targeted browsers for the production and development environments. It sets compatibility targets to ensure the application works on a wide range of browsers.

**DevDependencies:**

The **devDependencies** section contains **env-cmd** (^10.1.0), which is used for setting environment-specific variables during development.

This documentation covers the packages used in the frontend project and their purposes in the development and build processes.