Vite Configuration for React

When you create a React project with Vite, your `vite.config.js` (or `.ts`) typically looks like the following:

- `@vitejs/plugin-react` enables JSX support and fast refresh.
- `defineConfig` provides TypeScript support and better IDE suggestions.

```js
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig({
  plugins: [react()],
});
```

2. Configuring Aliases

You can define path aliases to simplify imports:

```js
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';
import path from 'path';

export default defineConfig({
  plugins: [react()],
  resolve: {
    alias: {
      '@': path.resolve(__dirname, './src'),
    },
  },
});
```

3. Customizing the Development Server

Vite allows you to configure the development server (`vite dev`) using the `server` option:

- `port`: Sets the port for the local server.
- `open`: Automatically opens the browser.

```
export default defineConfig({
  server: {
    port: 3000,  // Change the dev server port
    open: true,  // Opens browser in dev server
  },
});
```

4. Optimizing Build with `build` Options

To customize the production build:

- `outDir`: Specifies the output folder for `vite build`.
- `sourcemap`: Enables source maps for debugging production code.
- `chunkSizeWarningLimit`: Adjusts the warning threshold for large chunks.

```
export default defineConfig({
  build: {
    outDir: 'prod', // Change output directory
    sourcemap: true, // Generate source maps
    chunkSizeWarningLimit: 500, // Increase chunk size
limit warning
  },
});
```

5. CSS & PostCSS Configuration

Vite allows you to configure CSS pre-processors:

- This automatically imports a global SCSS file into all components.

```
export default defineConfig({
  css: {
    preprocessorOptions: {
      scss: {
        additionalData: `@import "@/styles/global.scss";`,
      },
    },
  },
});
```

6. Configuring Plugins

Vite supports plugins to extend its functionality. Some useful ones for React:

```
import react from '@vitejs/plugin-react';
import svgr from 'vite-plugin-svgr';

export default defineConfig({
  plugins: [
    react(),
    svgr(), // Enables importing SVGs as React components
  ],
});
```

Now, you can import SVGs like this:

```
import { ReactComponent as Logo } from '@/assets/logo.svg';
```

7. Using TypeScript in Vite

For TypeScript support, ensure you have `vite.config.ts`:

- `esbuild.jsxInject` automatically imports React in `.tsx` files, so you don't need to manually import it.

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig({
  plugins: [react()],
  esbuild: {
    jsxInject: `import React from 'react'`,
  },
});
```