React Tutorials - Quotes App -100- Getting Started

Introduction

This tutorial series will guide you through building a Quotes Application using React with TypeScript. We'll use modern development tools and best practices to create a responsive, user-friendly application.

Table of Contents

Prerequisites

Before starting, ensure you have Node.js installed on your system. This tutorial requires:

- **Node.js**: Version 18.0.0 or higher
  - To check your Node.js version, run: `node --version`
  - Download the latest version from Node.js official website

Installation Verification

Open a terminal and run these commands to verify your setup:

```
node --version    # Should show v18.0.0 or higher
npm --version     # Should show 9.0.0 or higher
```

If you need to update Node.js:

- **Windows/macOS**: Download the latest LTS version from Node.js website
- **Linux**: Use Node Version Manager (nvm)

  ```
  # Install nvm
  curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh |
  bash

  # Install latest LTS version
  nvm install --lts

  # Use the installed version
  nvm use --lts
  ```

Development Tools

Ensure you have the following tools installed:

- A modern code editor (VS Code recommended)
- Git for version control
- A modern web browser (Chrome or Firefox recommended)

Creating the Project Structure

Create react app using `vite`

Vite is a modern build tool that provides a faster and leaner development experience for modern web projects. It leverages native ES modules and provides a lightning-fast development server with Hot Module Replacement (HMR).

Why Vite?

- **Speed**: Vite is significantly faster than Create React App (CRA)
- **Modern Architecture**: Uses native ES modules for better performance
- **Optimized Build**: Produces highly optimized production builds
- **TypeScript Support**: Built-in TypeScript support without additional configuration
- **Hot Module Replacement**: Instant feedback during development

Using npm, pnpm, or yarn for building the project

You can use different package managers to create and manage your project. Choose the one that best fits your needs:

Package Manager Options

Choose your preferred package manager based on your needs:

npm

```
npm create vite@latest quotes-app --template react-ts
cd quotes-app
npm install
```

Key Advantages:

- Default Node.js package manager
- Largest package ecosystem
- Most community resources
- Best documentation
- Widest compatibility

pnpm

```
pnpm create vite@latest quotes-app --template react-ts
cd quotes-app
pnpm install
```

Key Advantages:

- Disk space efficient (uses symlinks)
- Faster installation speeds
- Strict dependency management
- Built-in monorepo support
- Lower disk usage

yarn

```
yarn create vite@latest quotes-app --template react-ts
cd quotes-app
yarn
```

Key Advantages:

- Parallel package downloads
- Deterministic installations
- Built-in security features
- Offline mode support

- Workspaces for monorepos

Project Dependencies Explained

- **bootstrap**: UI framework for responsive design
- **@types/bootstrap**: TypeScript type definitions for Bootstrap
- **react-ts template**: Includes TypeScript configuration and React setup

Run the project

Choose the command based on your package manager:

run using npm

```
npm run dev
```

run using pnpm

```
pnpm dev
```

run using yarn

```
yarn dev
```

After running these commands, your application will be available at http://localhost:5173

Project Structure

```
quotes-app/
├── src/
│   ├── App.tsx
│   ├── main.tsx
│   ├── App.css
│   └── index.css
├── public/
│   └── index.html
├── vite.config.ts
└── package.json
```

Install Bootstrap

```
pnpm install bootstrap @types/bootstrap
```

`main.tsx`: add the import statement to include bootstrap at the end of existing imports

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

Clean up the project

To start with a clean slate, we'll remove the default boilerplate code:

Clear the contents of the following files:

- `App.css`: Remove default styles. Empty the contents of the file.
- `index.css`: Remove default styles. Empty the contents of the file.
- `App.tsx`: Remove demo content. Empty the contents of the file

Modify code and test bootstrap installation

Open the `App.tsx` file

```
function App() {
  return (
    <div className="container mt-5">
      <button className="btn btn-primary">Hello World</button>
    </div>
  );
}

export default App;
```

This simple component tests that both React and Bootstrap are working correctly.

The `main.tsx` file

The contents of `main.tsx` file should like the one shown below.

```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import './index.css'
import App from './App.tsx'
import 'bootstrap/dist/css/bootstrap.min.css';
import 'bootstrap/dist/js/bootstrap.bundle.min.js';

createRoot(document.getElementById('root')!).render(
  <StrictMode>
    <App />
  </StrictMode>,
);
```

The main file imports Bootstrap CSS and sets up the React application root.

Open `public\index.html` file

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/quotes.png" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Quotes App</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.tsx"></script>
  </body>
</html>
```

This HTML file serves as the application shell and includes necessary meta tags for responsive design.

Configure Vite - `vite.config.ts`

Change the port to run on 3000 instead of the default port.

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

export default defineConfig({
  plugins: [react()],
  server: {
    port: 3000,
    host: '0.0.0.0'  // Allows access from other devices on the network
  }
});
```

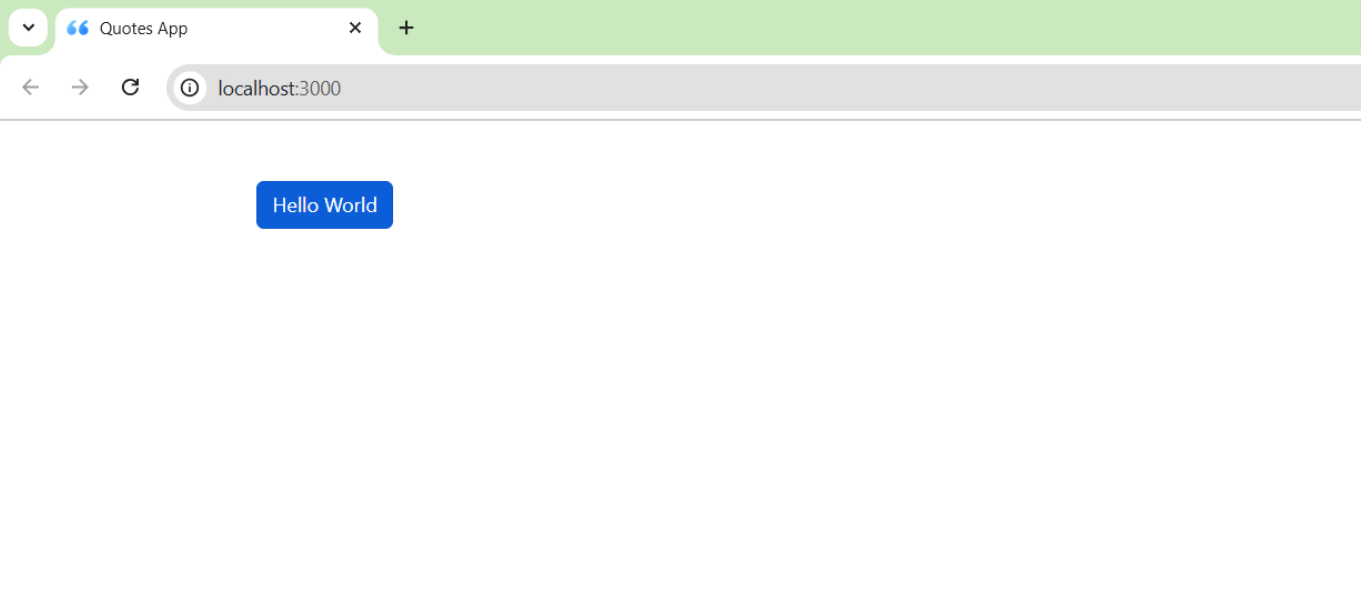
This configuration sets up Vite with React plugin and configures the development server.

Verify Installation

Start the application and check for:

1. The application runs without errors
2. You see a blue "Hello World" button (confirms Bootstrap is working)
3. The button is centered on the page (confirms container class is working)

```
pnpm dev
```



Setting Dev Environment Right

To enhance your development experience, install these recommended VS Code extensions:

- Auto Import
  - Automatically finds, parses and provides code actions for all available imports
  - Speeds up development by reducing manual import typing
- ES7+ React/Redux/React-Native snippets
  - Provides shortcuts for common React patterns
  - Includes TypeScript snippets
- HTML Format
  - Formats HTML code for better readability
  - Supports JSX/TSX formatting
- Prettier - Code formatter
  - Ensures consistent code formatting
  - Can be configured to format on save

Recommended VS Code Settings

Create or update `.vscode/settings.json` in your project:

```
{
  "editor.formatOnSave": true,
  "editor.defaultFormatter": "esbenp.prettier-vscode",
  "editor.tabSize": 2,
  "files.eol": "\n"
}
```

Troubleshooting Common Issues

Node.js Installation Issues

1. **Error: Node.js command not found**

   - Ensure Node.js is properly installed: `node --version`
   - Try restarting your terminal/command prompt
   - Check if Node.js is added to your system's PATH

2. **Permission Errors (Linux/Mac)**

   ```
   sudo chown -R $USER /usr/local/lib/node_modules
   sudo chown -R $USER ~/.npm
   ```

Vite Project Creation Issues

1. **'create-vite' command not found**

   - Try using npx: `npx create-vite@latest`
   - Or install globally: `npm install -g create-vite`

2. **TypeScript Template Errors**

   ```
   # Remove node_modules and reinstall
   rm -rf node_modules
   pnpm install
   ```

Bootstrap Integration Issues

1. **Bootstrap Styles Not Loading**

   - Verify import order in `main.tsx`
   - Check for console errors
   - Try clearing browser cache

2. **Bootstrap JavaScript Features Not Working**

   - Ensure both CSS and JS are imported:

   ```
   import 'bootstrap/dist/css/bootstrap.min.css';
   import 'bootstrap/dist/js/bootstrap.bundle.min.js';
   ```

Port Conflicts

1. **Port 3000 Already in Use**

   - Kill the process using the port:

     ```
     # Windows
     netstat -ano | findstr :3000
     taskkill /PID <PID> /F

     # Mac/Linux
     lsof -i :3000
     kill -9 <PID>
     ```

   - Or change the port in `vite.config.ts`

Development Environment

1. **VS Code Extensions Not Working**

   - Reload VS Code: `Ctrl/Cmd + Shift + P` → "Reload Window"
   - Verify extension installation
   - Check extension settings

2. **Prettier Not Formatting**

   - Verify `.vscode/settings.json` configuration
   - Try formatting manually: `Alt/Option + Shift + F`
   - Check if Prettier is set as default formatter

For additional help, consult:

- Vite Documentation
- React Documentation
- Bootstrap Documentation

Index | Next >>