

Docs

Components

Blocks

Charts

Themes

Colors

Search...

🔍

🌙

Getting Started

Introduction

Installation

components.json

Theming

Dark mode

CLI

Monorepo 

New

Next.js 15 + React 19

Typography

Open in v0

Figma

Changelog

Installation

Next.js

Vite

Remix

Astro

Laravel

Gatsbyby

Manual

Components

Accordion

Alert

Alert Dialog

Aspect Ratio

Avatar

Badge

Breadcrumb

Button

Calendar

Card

Carousel

Chart

Docs > Vite

Vite

Install and configure Vite.

1

Create project

Start by creating a new React project using vite :

pnpm

npm

yarn

bun

pnpm create vite@latest

2

Add Tailwind and its configuration

Install tailwindcss and its peer dependencies, then generate your tailwind.config.js and postcss.config.js files:

pnpm

npm

yarn

bun

pnpm add -D tailwindcss postcss autoprefixer

pnpm

npm

yarn

bun

pnpm dlx tailwindcss init -p

Add this import header in your main css file, src/index.css in our case:

1

@tailwind base;

@tailwind components;

@tailwind utilities;

/\* ... \*/

Configure the tailwind template paths in tailwind.config.js :

/\*\* @type {import('tailwindcss').Config} \*/

module.exports = {

content: [". /index.html", ". /src/\*\*/\*.

{ts,tsx,js,jsx}"],

theme: {

extend: {},

},

plugins: [],

}

3

Edit tsconfig.json file

The current version of Vite splits TypeScript configuration into three files, two of which need to be edited. Add the baseUrl and paths properties to the compilerOptions section of the tsconfig.json and tsconfig.app.json files:

1

{

"files": [],

"references": [

{

"path": ". /tsconfig.app.json"

},

{

"path": ". /tsconfig.node.json"

}

],

"compilerOptions": {

"baseUrl": ".",

"paths": {

"@/\*": [". /src/\*"]

}

}

}

4

Edit tsconfig.app.json file

Add the following code to the tsconfig.app.json file to resolve paths, for your IDE:

1

{

"compilerOptions": {

// ...

"baseUrl": ".",

"paths": {

"@/\*": [

". /src/\*"

]

}

// ...

}

}

5

Update vite.config.ts

Add the following code to the vite.config.ts so your app can resolve paths without error:

pnpm

npm

yarn

bun

pnpm add -D @types/node

import path from "path"

import react from "@vitejs/plugin-react"

import { defineConfig } from "vite"

export default defineConfig({

plugins: [react()],

resolve: {

alias: {

"@": path.resolve(\_\_dirname, ". /src"),

},

},

})

6

Run the CLI

Run the shadcn-ui init command to setup your project:

pnpm

npm

yarn

bun

pnpm dlx shadcn@latest init

7

Configure components.json

You will be asked a few questions to configure components.json :

1

Which style would you like to use? > New York

2

Which color would you like to use as base color?

> Zinc

3

Do you want to use CSS variables for colors? >

no / yes

8

That's it

You can now start adding components to your project.

pnpm

npm

yarn

bun

pnpm dlx shadcn@latest add button

The command above will add the Button component to your project. You can then import it like this:

1

import { Button } from "@ /components/ui/button"

2

3

export default function Home() {

4

return (

5

<div>

6

<Button>Click me</Button>

7

</div>

8

)

9

}

< Next.js

Remix >

Built by shadcn. The source code is available on GitHub.

PDF

The Docs are generated using the MDX plugin (React Compiler, Styled Components)

Created by FULLSTACK REACT PDF