Nuxt Vue web framework

START HERE

`npm create nuxt <dir>`
Scaffold a project

`npx nuxt dev`
Start the dev server

`npx nuxt add-template page <name>`
Create a page

`npx nuxt module add <module>`
Install a module

`npx nuxt typecheck`
Check types

`npx nuxt build`
Production build

`npx nuxt generate`
Static site output

`npx nuxt preview`
Try the build locally

nuxt dev HMR server → `nuxt build` → **.output** Nitro server bundle → `node` → **Run** SSR on Node or edge

CONCEPTS

Files in app/pages, app/components and app/composables are picked up and auto-imported: no import lines needed.

nuxt generate prerenders every route to plain HTML in .output/public for any static host.

PROJECT

`npm create nuxt <dir>`
Scaffold a new project

`npx nuxt dev`
Start the dev server with HMR
Port also comes from PORT or NUXT_PORT.

`npx nuxt build`
Production build into .output
Output is a Nitro server; run it with node .output/server/index.mjs.

`npx nuxt generate`
Prerender to static HTML
Same as build --prerender.

`npx nuxt preview`
Serve the production build locally
Run build first.

`npx nuxt prepare`
Generate .nuxt and types
Handy as a postinstall script or in CI before linting.

`npx nuxt typecheck`
Type-check the whole app
Needs vue-tsc or Golar; offers to install one.

`npx nuxt info`
Print versions and modules
Paste the output into bug reports; --json for scripts.

`npx nuxt cleanup`
Delete .nuxt, .output and caches
First thing to try when types or HMR act strange.

`npx nuxt upgrade`
Upgrade Nuxt and dependencies

`npx nuxt <cmd> --logLevel verbose`
More build output

DEV SERVER

`npx nuxt dev -p <port>`
Listen on another port
Add --strictPort to fail instead of picking a free one.

`npx nuxt dev -o`
Open the browser on start

`npx nuxt dev -h <host>`
Bind to a host or IP

DEV SERVER (cont.)

`npx nuxt dev --public`
Listen on all interfaces
Lets phones and other machines on the LAN reach it.

`npx nuxt dev --qr`
Show a QR code for the URL

`npx nuxt dev --https`
Serve over local HTTPS
Uses a generated, locally trusted certificate.

`npx nuxt dev --tunnel`
Share via a Cloudflare tunnel

`npx nuxt dev --dotenv <file>`
Load an extra .env file
Repeatable; later files win.

`npx nuxt dev --no-tui`
Plain log output

`npx nuxt dev --inspect`
Attach the Node inspector

BUILD & DEPLOY

`npx nuxt build --prerender`
Build and prerender static routes

`npx nuxt build --preset <name>`
Choose a deploy target
E.g. node-server, vercel, netlify, cloudflare-pages; NITRO_PRESET also works.

`npx nuxt analyze`
Inspect the bundle size
Experimental; serves a report unless --no-serve or CI is set.

`node .output/server/index.mjs`
Run the built server
PORT and HOST env vars control where it listens.

SCAFFOLDING

`npx nuxt add-template page <name>`
New page in app/pages

`npx nuxt add-template component <Name>`
New component in app/components

`npx nuxt add-template composable <name>`
New composable in app/composables

`npx nuxt add-template layout <name>`
New layout in app/layouts

`npx nuxt add-template middleware <name>`
New route middleware
Add --global to run it on every route.

SCAFFOLDING (cont.)

`npx nuxt add-template plugin <name>`
New plugin in app/plugins
--client or --server adds a mode suffix.

`npx nuxt add-template api <name>`
New API handler in server/api
Add --get or --post to restrict the method.

`npx nuxt add-template server-route <name>`
New route in server/routes

MODULES

`npx nuxt module add <module>`
Install and register a module
Edits package.json and the modules list in nuxt.config.

`npx nuxt module search <query>`
Find modules by keyword

`npx nuxt module remove <module>`
Uninstall and unregister a module

DIRECTORIES

Where things go

Path	Holds
`app/pages/`	file-based routes (about.vue, [id].vue)
`app/components/`	auto-imported components
`app/composables/`	auto-imported use* functions
`app/layouts/`	page wrappers, used with <NuxtLayout>
`app/middleware/`	route guards
`app/plugins/`	code that runs when the app starts
`app/utils/`	auto-imported helpers
`server/api/`	handlers served at /api/*
`server/routes/`	handlers served at their path
`server/middleware/`	run on every server request
`public/`	static files served from /
`shared/`	code usable by app and server

app/app.vue is the root component; add <NuxtPage /> there once you have a pages/ folder.

A file in server/api/hello.get.ts answers GET /api/hello only; the .get/.post suffix picks the method.

CONFIG

`nuxt.config.ts: modules`
List of extensions, loaded in order

`nuxt.config.ts: runtimeConfig`
Env-backed settings
Keys outside public stay server-only; override with NUXT_<KEY> env vars.

`nuxt.config.ts: css`
Global stylesheets

`nuxt.config.ts: app`
baseURL, head tags, transitions

`nuxt.config.ts: ssr`
Turn server rendering on or off
ssr: false gives a client-only SPA.

`nuxt.config.ts: routeRules`
Per-route caching and rendering
Mix prerender, swr, redirect and ssr: false per path.

`nuxt.config.ts: compatibilityDate`
Pin behaviour of presets and modules

`nuxt.config.ts: devtools`
Enable Nuxt DevTools

`nuxt.config.ts: nitro`
Server engine options

`nuxt.config.ts: vite`
Pass options to Vite

Env var to config

Env var	Sets
`NUXT_PORT / PORT`	dev and preview port
`NUXT_HOST / HOST`	listen host
`NUXT_<KEY>`	runtimeConfig.<key>
`NUXT_PUBLIC_<KEY>`	runtimeConfig.public
`NITRO_PRESET`	build deploy target

RECIPES

New project

```
npm create nuxt my-app
cd my-app
npx nuxt dev
```

Add a module and a page

```
npx nuxt module add @nuxt/image
npx nuxt add-template page
about
# edit app/pages/about.vue
```

Static site

```
npx nuxt generate
npx nuxt preview
# upload .output/public
```

Minimal nuxt.config.ts

```
export default
defineNuxtConfig({
  compatibilityDate:
'2025-07-15',
  modules: ['@nuxt/image'],
  runtimeConfig: { public:
{ apiBase: '' } },
  routeRules: { '/blog/**':
{ prerender: true } },
})
```