# GIỚI THIỆU

## NUXT

Là một frameword của Vue, Nuxt giúp bạn xây dựng ứng dụng web hiện đại, mạnh mẽ, đặc biệt là **SSR (Server Side Rendering)**, **Static Site Generation (SSG)**, và cả **Hybrid Rendering**.

Có 2 bản là Nuxt 2 và Nuxt 3 hỗ trợ Vue 2 và Vue 3 tương ứng.

## Tự động hóa và quy ước

* Định tuyến: định tuyến tự động dựa trên cấu trúc nằm trong thư mục pages/
* Tự động tách mã nhỏ hơn
* Đi kèm khả năng tích hợp SSR (tự thiếp lập máy chủ)
* Tự động import component

## Công cụ máy chủ

Cung cấp khả năng lập trình fullstack, sử dụng **Rollup** và **Nodejs**. Tạo server API bởi đọc file **server/api/** và server middleware từ **server/middleware**

Khi build project Nitro sẽ build app và server thành 1 file phổ quát là **.output**

# CÀI ĐẶT

## Yêu cầu

* Node.js phiên bản 18. trở lên
* IDE lập trình như VSCode hoặc Webstorm

## Cài đặt

npx nuxi@latest init <project-name>

## Cấu hình

Các tệp cấu hình dự án thường nằm tại thư mục gốc của dự án

VD: myproject/các tệp cấu hình

### nuxt.config.ts

Ghi đè hoặc mở rộng các hành vi của ứng dụng

export default defineNuxtConfig({

// My Nuxt config

})

Hoặc có thể khai báo các biến môi trường tại đây thay vì các file .env như trước kia và được đặt trong biến runtimeConfig (private), runtimeConfig/public.

Các dữ liệu công khai cho phép cả server và client đọc được thì lưu ở **runtimeConfig/public** còn chỉ server đọc được thì lưu ở **runtimeConfig**

export default defineNuxtConfig({

runtimeConfig: {

// The private keys which are only available server-side

apiSecret: '123',

// Keys within public are also exposed client-side

public: {

apiBase: '/api'

}

}

})

Sử dụng bằng cách gọi useRuntimeConfig()

<script setup lang="ts">

const runtimeConfig = useRuntimeConfig()

</script>

Tuy nhiên nếu bạn có file .env nó sẽ ghi đè file config bởi .env (ưu tiên .env hơn)

### app.config.ts

Khá giống với nuxt.config.ts là đều khai báo các biến có thể dùng nhiều nơi trong ứng dụng, tuy nhiên với app.config.ts sẽ có vài điểm khác biệt.

Thường dùng cho cấu hình giao diện của ứng dụng: tên app, màu sắc, cấu hình mặc định cho UI components, và các cấu hình public khác. Khác với runtimeConfig, những thứ này sẽ không thể bị ghi đè bởi các cấu hình môi trường

export default defineAppConfig({

title: 'My Awesome Nuxt App',

theme: {

primaryColor: '#3490dc'

}

})

Cách sử dụng

<script setup lang="ts">

const appConfig = useAppConfig()

console.log(appConfig.title) // "My Awesome Nuxt App"

console.log(appConfig.theme.primaryColor) // "#3490dc"

</script>

### Các bản config khác

Thông thường có thể bạn sẽ có những file config khác như: postcss.config.js, vite.config.ts, tsconfig.json, tailwind.config.js, … Tuy nhiên trong nuxt sẽ gộp tất cả các file cấu hình đó vào 1 nut.config.ts, tuy nhiên chỉ những file liên quan trực tiếp với build system

VD: postcss.config.js thì được viết trong nuxt.config.ts như sau

export default defineNuxtConfig({

postcss: {

plugins: {

tailwindcss: {},

autoprefixer: {}

}

}

})

Tuy nhiên với những file không liên quan trực tiếp đến build system thì không cần cho vào mà vẫn giữ nguyên file: tsconfig.json, eslint.config.js, prettier.config.js, stylelint.config.js, tailwind.config.js, vitest.config.ts, …