Tutorial mastering-nuxt-3

# Chapter 1 - Building Our MVP

## 1-0 Getting Set Up

In this lesson, you’ll get set up to start working on the Mastering Nuxt 3 codebase.

We’ll download the code, install pnpm and nuxi, and see how we can use git to checkout the code from a specific lesson.

Update, March 2023: Only the most recent commits are shown when running git log, so you may need to scroll down (using the mouse wheel, enter, or space bar) in order to see the first commits.

pnpm: <https://pnpm.io>

Fig: <https://fig.io>

## 1-1 File based routing

In this lesson, we’ll learn how to use Nuxt to create routes in our app. We’ll see how Nuxt automatically turns our files and directories into routes using Vue Router.

TailwindCSS: <https://tailwindcss.com>

Tailwind Prose plugin: <https://tailwindcss.com/docs/typography-plugin>

NuxtTailwind: <https://tailwindcss.nuxtjs.org>

Nuxt 3 tự động tạo route path theo cấu trúc trong thư mục pages.

Mặc định **pages** là thư mục gốc.

|  |  |
| --- | --- |
|  | A screenshot of a computer  Description automatically generated with medium confidence |

## 1-2 Displaying Course Lessons

In this lesson, we figure out how we want to set up our routes in order to show course lessons. We’ll discuss what makes a good route, and outline how we’ll implement it.

A screen shot of a computer

Description automatically generated with low confidence

* first-chapter, second-lesson: is dinamic route
* soucre/chapter/lesson: is not dinamic route

## 1-3 Nested Routes

In this lesson, we’ll learn how we can use nested routes to create powerful app hierarchies. We’ll separate out the logic for displaying a lesson into it’s own child page.

**Example**: course/lesson; **course** is parent, **lesson** is children (nested route).

**app.vue**

A screen shot of a computer program

Description automatically generated with low confidence

**pages/course.vue**

A screen shot of a computer program

Description automatically generated with low confidence

**course/lesson.vue**

A screenshot of a computer program

Description automatically generated with medium confidence

**Summary to create nested route**

* course: course.vue, folder course.
* lesson: course/lesson.vue.

## 1-4 Dynamic Routes

In this lesson, we’ll explore dynamic routes in Nuxt. This will let us navigate to any chapter and lesson in our course. We’ll also see all the different ways we can use dynamic routes to create complex routing.

### Example 1

A screen shot of a computer

Description automatically generated with low confidence

**Do example 1**

Từ file lesson.vue, thay đổi tên file bằng:

A screen shot of a computer

Description automatically generated with medium confidence

Sau đó ấn enter, được kết quả:

A screen shot of a computer

Description automatically generated with low confidence

Truy cập từ URL

A screenshot of a computer

Description automatically generated with medium confidence

We can access params url:

A screen shot of a computer program

Description automatically generated with low confidence

### Example 2



**Do example 2**

Từ file course/lesson.vue, thay đổi tên file bằng:

A screenshot of a computer

Description automatically generated with medium confidence

Sau đó ấn enter, được kết quả:

A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer program

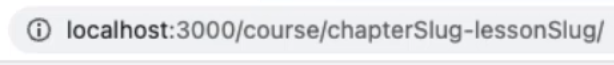
Description automatically generated with low confidence

Truy cập từ URL

A screenshot of a computer

Description automatically generated with medium confidence

### Example 3



A picture containing text, font, screenshot

Description automatically generated

### Example 4

Truy cập URL với tham số tùy chọn [[lessonSlug]]

A picture containing text, font, screenshot, line

Description automatically generated

A picture containing text, font, screenshot

Description automatically generated

Truy cập từ URL

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

**Vue devtool route**

A green and blue rectangle with black text

Description automatically generated with low confidence

## 1-5 Loading in Course Data

In this lesson, we’ll hook up some actual data so we have something more substantial to work with. We’ll also see how to use nuxi to generate files for us.

nuxi: <https://v3.nuxtjs.org/api/commands/add>

Configuring auto-imports: <https://v3.nuxtjs.org/api/configuration/nuxt.config#imports>

### Create compatable

nuxi add composable useCourse

A screen shot of a computer

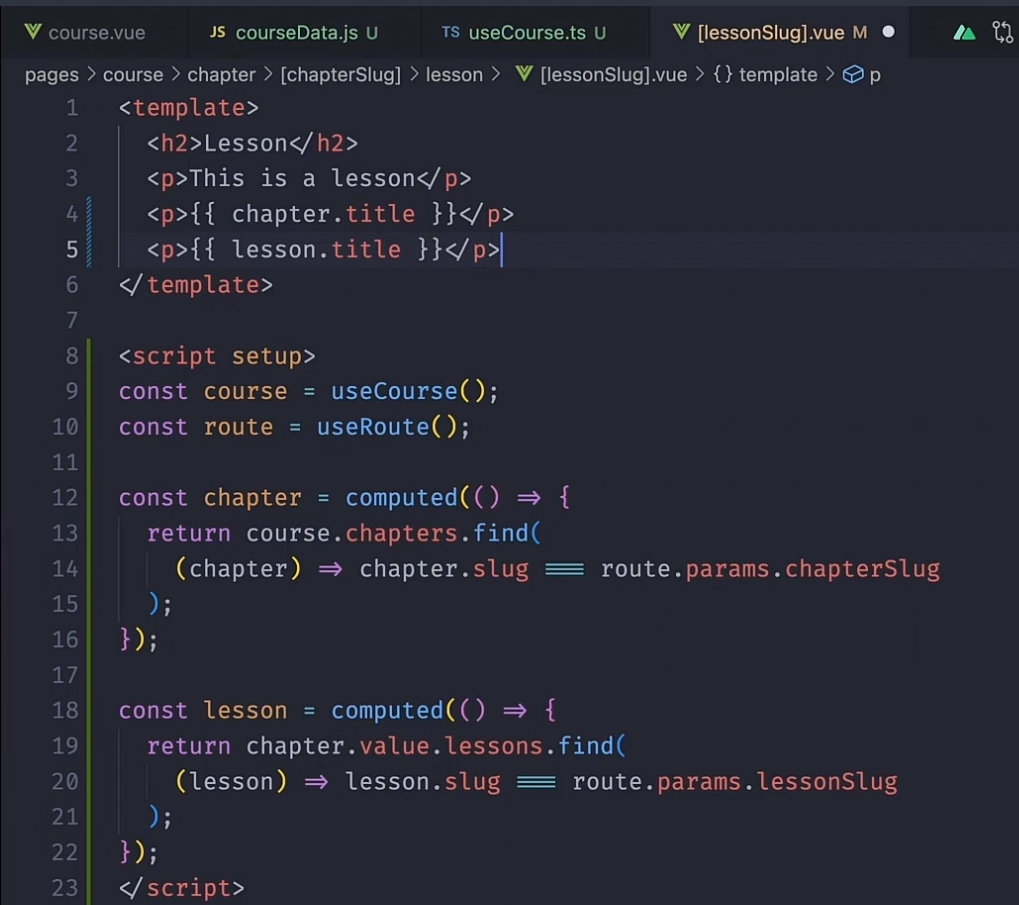
Description automatically generated with low confidence

### Create data

A screenshot of a computer screen

Description automatically generated with low confidence

### Import data to component



## 1-7 Add VideoPlayer Component

### Create component

nuxi add component VideoPlayer

A screen shot of a computer

Description automatically generated with low confidence



## 1-8 What is Universal Rendering?

In this lesson we do a deep dive into how Nuxt renders our apps. We’ll take some time to understand how Universal rendering works, and why it gives us the best of SPA and SSR.

Nuxt 3 rendering modes: <https://v3.nuxtjs.org/guide/concepts/rendering>

Ref: <https://viblo.asia/p/so-sanh-server-side-rendering-vs-client-side-rendering-vs-pre-rendering-vs-dynamic-rendering-LzD5dWoOljY>

### Server side rendering (SSR)

A picture containing text, screenshot, jack, design

Description automatically generated

* Cơ chế: Server xử lý logic, render ra html rồi gửi về client hiển thị kết quả.
* Ưu: load lần đầu nhanh, SEO.
* Nhược:
  + Tốn băng thông, nhiều request server se quá tải.
  + Mỗi lần request, sang trang mới, server lại phải render lại toàn bộ trang 🡪 lâu.

### Single page app

A picture containing text, screenshot, design

Description automatically generated

Cơ chế: lần đầu client request đến server, server xử lý, trả về app bundle. Từ lần 2 trở đi, client tự request và render ra html.

Ưu: chia tải với server, việc xử lý logic ở server, client render và hiển thị kết quả. Server chỉ trả về dữ liệu json để client hiển thị ra.

Nhược: khởi động lần đầu chậm, không SEO.

### Universal rendering

A picture containing text, screenshot, design

Description automatically generated

Cơ chế: Request lần đầu, server xử lý logic và render ra html rồi gửi về client. Từ lần 2 trở đi, client client tự request, nhận kết quả json từ server trả về và render ra html.

## 1-9 Navigating with NuxtLink

In this lesson, we’ll see how and why NuxtLink is so useful. We’ll update the navigation of our app to be more efficient, and dig into what NuxtLink is actually doing for us.

NuxtLink: <https://v3.nuxtjs.org/api/components/nuxt-link>

### Example

|  |  |
| --- | --- |
|  |  |

### Active link

Cách 1

A picture containing text, font, screenshot, line

Description automatically generated

Cách 2

A screen shot of a computer program

Description automatically generated with low confidence

## 1-11 The useHead composable

In this lesson, we’ll investigate the useHead to reactively set the title of our page.

useHead: <https://v3.nuxtjs.org/api/composables/use-head>

## 1-12 Tracking progress with useState

In this lesson we’ll dip our toes into state management with useState. We’ll see how we can use it to track state across our app, and why we’d use it instead of ref.

useState: <https://v3.nuxtjs.org/getting-started/state-management>

### Create

A screen shot of a computer program

Description automatically generated with low confidence

### Using

A screen shot of a computer program

Description automatically generated with medium confidence

useState được đặt trong mục composables sử dụng để

* Truyền đi tất cả đến các compoment
* Khi giá trị trong useState thay đổi ở trang A (ví dụ là isLogin ở trên từ true sang fasle), sau đó chuyển qua trang B, rồi quay lại trang A thì giá trị false ở trang A đã chuyển trước đó không mất giá trị (trừ khi refresh lại trang). Note: sử dụng isLogin = ref(true) sẽ bị restart lại giá trị ban đầu là true, không là false.
* Nếu isLogin được gọi ở component A và B, thì nếu component A thay đổi isLogin thì ở component B cũng tự động được update và thay đổi theo.

# 1-13 Save Course Progress with VueUse and the ClientOnly

### Component

In this lesson we’ll see how we can work with third-party modules. We’ll use VueUse to help us store our state in localStorage. Then we’ll dive into understanding the ClientOnly component, why we’d want to use it, and a convenient alternative.

Using VueUse with Nuxt: <https://vueuse.org/guide/#nuxt>

useLocalStorage: <https://vueuse.org/core/uselocalstorage/#uselocalstorage>

ClientOnly: <https://v3.nuxtjs.org/api/components/client-only>

### useLocalStorage

Hình mục 1-12, sử dụng **useLocalStorage** để lưu giá trị **process** khi thực hiện **handleChangeLogin**.

A screenshot of a computer

Description automatically generated

### ClientOnly

Sử dụng để xác định đoạn mã js trong file hoặc file cần render ở Client (không render ở Server sau đó trả về khi refresh trang). Điều này giúp cho giữ được trạng thái của Client, không bị ảnh hưởng khi fresh trang.

**Cho js trong file**

A picture containing text, font, screenshot

Description automatically generated

**Cho cả file**

A picture containing text, screenshot, font, line

Description automatically generated

## 1-14 Deploying our MVP to Netlify

Finally, we get to deploy our app!

In this lesson, we’ll see how we can deploy our app using Netlify. From this point on, you’ll be able to deploy your app whenever you want throughout the course.

A great way to share your progress with friends and co-workers!

Update, March 2023: If you’re using pnpm you will need to add a .nprmc file to the root of your project with the single line shamefully-hoist=true. This will make sure that when Netlify builds your project it runs pnpm with the flag --shamefully-hoist.

Netlify: <https://www.netlify.com/>

Zero-configuration deploy with Nitro: <https://nitro.unjs.io/deploy/#zero-config-providers>

# Chapter 2 - Architecture and Organizing Our Code

## Chapter 1 - Building Our MVP

In this chapter, we’ll take a step back and look deeper into Nuxt 3 architecture.

We’ll see how we can use Pages, Layouts, and Components to organize our code better, and increase the maintainability of our code over time.

## 2-1 Smaller is better / but not too small

In this lesson, we explore how smaller pieces of functionality are generally easier to work with.

We also outline the different tools that Nuxt 3 gives us to help us organize our app code in an effective way.

## 2-2 Understanding App.vue

In this lesson, we take a quick look at the app.vue file, the component where everything begins.

app.vue: <https://v3.nuxtjs.org/guide/directory-structure/app>