



Table of Contents >

SEO and Meta

Improve your Nuxt app's SEO with powerful head config, composables and components.

Defaults

Out-of-the-box, Nuxt provides sane defaults, which you can override if needed.

- `charset` : `utf-8`
- `viewport` : `width=device-width, initial-scale=1`

```
export default defineNuxtConfig({  
  app: {  
    head: {  
      charset: 'utf-8',  
      viewport: 'width=device-width, initial-scale=1',  
    }  
  }  
})
```

nuxt.config.ts

Providing an `app.head` property in your `nuxt.config.ts` allows you to customize the head for your entire app.

This method does not allow you to provide reactive data. We recommend to use `useHead()` in `app.vue`.

Shortcuts are available to make configuration easier: `charset` and `viewport`. You can also provide any of the keys listed below in Types.

useHead

The `useHead` composable function allows you to manage your head tags in a programmatic and reactive way, powered by `Unhead`.

As with all composables, it can only be used with a components `setup` and lifecycle hooks.

```
<script setup lang="ts">
useHead({
  title: 'My App',
  meta: [
    { name: 'description', content: 'My amazing site.' }
  ],
  bodyAttrs: {
    class: 'test'
  },
  script: [ { innerHTML: 'console.log(\'Hello world\')' } ]
})
</script>
```

app.vue

We recommend to take a look at the `useHead` and `useHeadSafe` composables.

useSeoMeta and useServerSeoMeta

The `useSeoMeta` and `useServerSeoMeta` composables let you define your site's SEO meta tags as a flat object with full TypeScript support.

This helps you avoid typos and common mistakes, such as using `name` instead of `property` .

```
<script setup lang="ts">
useSeoMeta({
  title: 'My Amazing Site',
  ogTitle: 'My Amazing Site',
  description: 'This is my amazing site, let me tell you all about it.',
  ogDescription: 'This is my amazing site, let me tell you all about it.',
  ogImage: 'https://example.com/image.png',
  twitterCard: 'summary_large_image',
})
</script>
```

app.vue

Read more on the `useSeoMeta` and `useServerSeoMeta` composables.

Components

Nuxt provides `<Title>` , `<Base>` , `<NoScript>` , `<Style>` , `<Meta>` , `<Link>` , `<Body>` , `<Html>` and `<Head>` components so that you can interact directly with your metadata within your component's template.

Because these component names match native HTML elements, it is very important that they are capitalized in the template.

`<Head>` and `<Body>` can accept nested meta tags (for aesthetic reasons) but this has no effect on *where* the nested meta tags are rendered in the final HTML.

```
<script setup lang="ts">
const title = ref('Hello World')
</script>

<template>
  <div>
    <Head>
      <Title>{{ title }}</Title>
      <Meta name="description" :content="title" />
      <Style type="text/css" children="body { background-color: green; }" />
    </Head>

    <h1>{{ title }}</h1>
  </div>
</template>
```

app.vue

Types

Below are the non-reactive types used for `useHead` , `app.head` and components.

```
interface MetaObject {
  title?: string
  titleTemplate?: string | ((title?: string) => string)
  templateParams?: Record<string, string | Record<string, string>>
  base?: Base
```

```

link?: Link[]
meta?: Meta[]
style?: Style[]
script?: Script[]
noscript?: Noscript[];
htmlAttrs?: HtmlAttributes;
bodyAttrs?: BodyAttributes;
}

```

See [@unhead/schema](#) for more detailed types.

Features

Reactivity

Reactivity is supported on all properties, as computed, getters and reactive.

It's recommended to use getters (`() => value`) over computed (`computed(() => value)`).

useHead **useSeoMeta** **Components**

```

<script setup lang="ts">
const description = ref('My amazing site.')

useHead({
  meta: [
    { name: 'description', content: description }
  ],
})
</script>

```

useHead

Title Template

You can use the `titleTemplate` option to provide a dynamic template for customizing the title of your site. for example, by adding the name of your site to the title of every page.

The `titleTemplate` can either be a string, where `%s` is replaced with the title, or a function.

If you want to use a function (for full control), then this cannot be set in your `nuxt.config`, and it is recommended instead to set it within your `app.vue` file, where it will apply to all pages on your site:

useHead

```
<script setup lang="ts">
useHead({
  titleTemplate: (titleChunk) => {
    return titleChunk ? `${titleChunk} - Site Title` : 'Site Title';
  }
})
</script>
```

useHead

Now, if you set the title to `My Page` with `useHead` on another page of your site, the title would appear as `'My Page - Site Title'` in the browser tab. You could also pass `null` to default to the site title.

Body Tags

You can use the `tagPosition: 'bodyClose'` option on applicable tags to append them to the end of the `<body>` tag.

For example:

```
<script setup lang="ts">
useHead({
  script: [
    {
      src: 'https://third-party-script.com',
      // valid options are: 'head' | 'bodyClose' | 'bodyOpen'
      tagPosition: 'bodyClose'
    }
  ]
})
</script>
```

Examples

With `definePageMeta`

Within your `pages/` directory, you can use `definePageMeta` along with `useHead` to set metadata based on the current route.

For example, you can first set the current page title (this is extracted at build time via a macro, so it can't be set dynamically):

```
<script setup lang="ts">
definePageMeta({
  title: 'Some Page'
})
</script>
```

`pages/some-page.vue`

And then in your layout file, you might use the route's metadata you have previously set:

```
<script setup lang="ts">
const route = useRoute()

useHead({
  meta: [{ property: 'og:title', content: `App Name - ${route.meta.title}` }]
})
</script>
```

`layouts/default.vue`

 [Read and edit a live example in Docs > Examples > Features > Meta Tags.](#)

 [Read more in Docs > Guide > Directory Structure > Pages > #page Metadata.](#)

Dynamic Title

In the example below, `titleTemplate` is set either as a string with the `%s` placeholder or as a function, which allows greater flexibility in setting the page title dynamically for each route of your Nuxt app:

```
<script setup lang="ts">
useHead({
  // as a string,
  // where `%s` is replaced with the title
```

`app.vue`

```

titleTemplate: '%s - Site Title',
// ... or as a function
titleTemplate: (productCategory) => {
  return productCategory
    ? `${productCategory} - Site Title`
    : 'Site Title'
}
})
</script>

```

`nuxt.config` is also used as an alternative way of setting the page title. However, `nuxt.config` does not allow the page title to be dynamic. Therefore, it is recommended to use `titleTemplate` in the `app.vue` file to add a dynamic title, which is then applied to all routes of your Nuxt app.

External CSS

The example below shows how you might enable Google Fonts using either the `link` property of the `useHead` composible or using the `<Link>` component:

useHead Components

```

<script setup lang="ts">
useHead({
  link: [
    {
      rel: 'preconnect',
      href: 'https://fonts.googleapis.com'
    },
    {
      rel: 'stylesheet',
      href: 'https://fonts.googleapis.com/css2?family=Roboto&display=swap',
      crossorigin: ''
    }
  ]
})
</script>

```

useHead



© 2016-2023 Nuxt - MIT License

Enterprise

Design Kit

NuxtLabs

Nuxt Studio

