Nuxt ▾   Search `CTRL` `K`

Table of Contents ›

`<NuxtPage>`

`<NuxtPage>` is a built-in component that comes with Nuxt. `NuxtPage` is required to display top-level or nested pages located in the `pages/` directory.

`NuxtPage` is a wrapper around `<RouterView>` component from Vue Router. `NuxtPage` component accepts same `name` and `route` props.

- **name:** type: `string`

`name` helps `RouterView` render the component with the corresponding name in the matched route record's components option.

- **route:** type: `RouteLocationNormalized`

`route` is a route location that has all of its components resolved.

> **Nuxt automatically resolves the** `name` **and** `route` **by scanning and rendering all Vue component files found in the** `/pages` **directory.**

Apart from the `name` and `route` , `NuxtPage` component also accepts `pageKey` props.

- **pageKey:** type: `string` or `function`

`pageKey` helps control when the `NuxtPage` component is re-rendered.

Example

For example, passing `static` key, `NuxtPage` component is rendered only once when it is mounted.

```
<!-- static key -->
<NuxtPage page-key="static" />
```

You can also use a dynamic key based on the current route. (Don't use `$route` object here as it can cause problems with how `<NuxtPage>` renders pages with `<Suspense>` .)

```
<NuxtPage :page-key="route => route.fullPath" />
```

Alternatively, `pageKey` can be passed as a `key` value via `definePageMeta` from the `<script>` section of your Vue component in the `/pages` directory.

```ts
<script setup lang="ts">
definePageMeta({
  key: route => route.fullPath
})
</script>
```

Open on StackBlitz

Accessing a page's component ref

To get the ref of a page component, access it through `ref.value.pageRef`

```ts
<script setup lang="ts">
const page = ref()
function logFoo () {
  page.value.pageRef.foo()
}
</script>

<template>
  <NuxtPage ref="page" />
</template>
```

Custom Props

In addition, `NuxtPage` also accepts custom props that you may need to pass further down the hierarchy. These custom props are accessible via `attrs` in the Nuxt app.

```
<NuxtPage :foobar="123" />
```

For example, in the above example, the value of `foobar` will be available using `$attrs.foobar` in the template or `useAttrs().foobar` in `<script setup>`.

> 👉 Read more in Docs > Guide > Directory Structure > App.

✏ Edit on Github