

[Table of Contents](#) >

<NuxtLayout>

You can use `<NuxtLayout />` component to activate default layout on `app.vue` or `error.vue`.

```
<template>
  <NuxtLayout>
    some page content
  </NuxtLayout>
</template>
```

`/app.vue`

`<NuxtLayout />` can be used to override default layout on `app.vue`, `error.vue` or even page components found in the `/pages` directory.

name prop

`<NuxtLayout />` component accepts the `name` prop, which you can pass to use a non-default layout, where `name` can be a static string, reactive reference or a computed property. It **must** match the name of the corresponding layout file in the `/layouts` directory.

Examples

```
<script setup lang="ts">
  // layouts/custom.vue
  const layout = 'custom'
</script>

<template>
```

`pages/index.vue`

```
<NuxtLayout :name="layout">
  <NuxtPage />
</NuxtLayout>
</template>
```

Please note the layout name is normalized to kebab-case, so if your layout file is named `errorLayout.vue`, it will become `error-layout` when passed as a `name` property to `<NuxtLayout />`.

```
<template>
  <NuxtLayout name="error-layout">
    <NuxtPage />
  </NuxtLayout>
</template>
```

/error.vue

Passing Additional Props

`NuxtLayout` also accepts any additional props that you may need to pass to the layout. These custom props are then made accessible as attributes.

```
<div>
  <NuxtLayout name="custom" title="I am a custom layout">
  </NuxtLayout>
</div>
```

pages/some-page.vue

In the above example, the value of `title` will be available using `$attrs.title` in the template or `useAttrs().title` in `<script setup>` at `custom.vue`.

```
<script setup lang="ts">
  const layoutCustomProps = useAttrs()
  console.log(layoutCustomProps.title) // I am a custom layout
</script>
```

layouts/custom.vue

Layout and Transition

`<NuxtLayout />` renders incoming content via `<slot />`, which is then wrapped around Vue's `<Transition />` component to activate layout transition. For this to work as expected, it is recommended that `<NuxtLayout />` is **not** the root element of the page component.

```
<template>
  <div>
    <NuxtLayout name="custom">
      <template #header> Some header template content. </template>
    </NuxtLayout>
  </div>
</template>
```

pages/index.vue


Accessing a layout's component ref

To get the ref of a layout component, access it through `ref.value.layoutRef`

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

<template>
  <NuxtLayout ref="layout" />
</template>
```

 [Read more in Docs > Guide > Directory Structure > Layouts.](#)

 [Edit on Github](#)



© 2016-2023 Nuxt - MIT
License

Enterprise

Design Kit
Nuxt Studio



NuxtLabs

