



# <NuxtLink>

Nuxt provides `<NuxtLink>` component to handle any kind of links within your application.

`<NuxtLink>` is a drop-in replacement for both Vue Router's `<RouterLink>` component and HTML's `<a>` tag. It intelligently determines whether the link is *internal* or *external* and renders it accordingly with available optimizations (prefetching, default attributes, etc.)

## Examples

### Basic Usage

In this example, we use `<NuxtLink>` component to link to a website.

```
<template>
  <NuxtLink to="https://nuxtjs.org">
    Nuxt website
  </NuxtLink>
  <!-- <a href="https://nuxtjs.org" rel="noopener noreferrer">...</a> -->
</template>
```

app.vue

### Internal Routing

In this example, we use `<NuxtLink>` component to link to another page of the application.

&lt;template&gt;

pages/index.vue

```

<NuxtLink to="/about">
  About page
</NuxtLink>
<!-- <a href="/about">...</a> (+Vue Router & prefetching) -->
</template>

```

## target and rel Attributes

In this example, we use `<NuxtLink>` with `target`, `rel`, and `noRel` props.

```

<template>
  <NuxtLink to="https://twitter.com/nuxt_js" target="_blank">
    Nuxt Twitter
  </NuxtLink>
  <!-- <a href="https://twitter.com/nuxt_js" target="_blank" rel="noopener noreferrer">...</a> -->

  <NuxtLink to="https://discord.nuxtjs.org" target="_blank" rel="noopener">
    Nuxt Discord
  </NuxtLink>
  <!-- <a href="https://discord.nuxtjs.org" target="_blank" rel="noopener">...</a> -->

  <NuxtLink to="https://github.com/nuxt" no-rel>
    Nuxt GitHub
  </NuxtLink>
  <!-- <a href="https://github.com/nuxt">...</a> -->

  <NuxtLink to="/contact" target="_blank">
    Contact page opens in another tab
  </NuxtLink>
  <!-- <a href="/contact" target="_blank" rel="noopener noreferrer">...</a> -->
</template>

```

app.vue

## Props

- **to**: Any URL or a route location object from Vue Router
- **href**: An alias for `to`. If used with `to`, `href` will be ignored
- **target**: A `target` attribute value to apply on the link

- **rel:** A `rel` attribute value to apply on the link. Defaults to `"noopener noreferrer"` for external links.
- **noRel:** If set to `true`, no `rel` attribute will be added to the link
- **activeClass:** A class to apply on active links. Works the same as Vue Router's `active-class` prop on internal links. Defaults to Vue Router's default (`"router-link-active"`)
- **exactActiveClass:** A class to apply on exact active links. Works the same as Vue Router's `exact-active-class` prop on internal links. Defaults to Vue Router's default `"router-link-exact-active"`
- **replace:** Works the same as Vue Router's `replace` prop on internal links
- **ariaCurrentValue:** An `aria-current` attribute value to apply on exact active links. Works the same as Vue Router's `aria-current-value` prop on internal links
- **external:** Forces the link to be considered as external (`true`) or internal (`false`). This is helpful to handle edge-cases
- **prefetch** and **noPrefetch:** Whether to enable prefetching assets for links that enter the view port.
- **prefetchedClass:** A class to apply to links that have been prefetched.
- **custom:** Whether `<NuxtLink>` should wrap its content in an `<a>` element. It allows taking full control of how a link is rendered and how navigation works when it is clicked. Works the same as Vue Router's `custom` prop

👉 Defaults can be overwritten, see [overwriting defaults](#) if you want to change them.

## Overwriting Defaults

You can overwrite `<NuxtLink>` defaults by creating your own link component using `defineNuxtLink`.

```
export default defineNuxtLink({
  componentName: 'MyNuxtLink',
  /* see signature below for more */
})
```

components/MyNuxtLink.ts

You can then use `<MyNuxtLink />` component as usual with your new defaults.

## defineNuxtLink Signature

```
defineNuxtLink({
  componentName?: string;
  externalRelAttribute?: string;
  activeClass?: string;
  exactActiveClass?: string;
  prefetchedClass?: string;
  trailingSlash?: 'append' | 'remove'
}) => Component
```

- **componentName:** A name for the defined `<NuxtLink>` component.
- **externalRelAttribute:** A default `rel` attribute value applied on external links. Defaults to `"noopener noreferrer"`. Set it to `""` to disable
- **activeClass:** A default class to apply on active links. Works the same as Vue Router's `linkActiveClass` option. Defaults to Vue Router's default (`"router-link-active"`)
- **exactActiveClass:** A default class to apply on exact active links. Works the same as Vue Router's `linkExactActiveClass` option. Defaults to Vue Router's default (`"router-link-exact-active"`)
- **prefetchedClass:** A default class to apply to links that have been prefetched.
- **trailingSlash:** An option to either add or remove trailing slashes in the `href`. If unset or not matching the valid values `append` or `remove`, it will be ignored.

 [Read and edit a live example in Docs > Examples > Routing > Pages.](#)

 [Edit on Github](#)



© 2016-2023 Nuxt - MIT  
License

Enterprise

Design Kit  
Nuxt Studio

NuxtLabs

