



# Adding custom routes

In Nuxt 3, your routing is defined by the structure of your files inside the pages directory. However, since it uses vue-router under the hood, Nuxt offers you several ways to add custom routes in your project.

## Using router config

Using router options, you can optionally override or extend your routes using a function that accepts the scanned routes and returns customized routes.

If it returns `null` or `undefined`, Nuxt will fall back to the default routes (useful to modify input array).

```
import type { RouterConfig } from '@nuxt/schema'
// https://router.vuejs.org/api/interfaces/routeroptions.html
export default <RouterConfig> {
  routes: (_routes) => [
    {
      name: 'home',
      path: '/',
      component: () => import('~pages/home.vue').then(r => r.default || r)
    }
  ],
}
```

app/router.options.ts

Nuxt will not augment any new routes you return from the `routes` function with metadata defined in `definePageMeta` of the component you provide. If you want that to happen, you should use the `pages:extend` hook which is called at build-time.

# Using the `pages:extend` hook

You can add, change or remove pages from the scanned routes with the `pages:extend` nuxt hook. For example, to prevent creating routes for any `.ts` files:

```
export default defineNuxtConfig({nuxt.config.ts
  hooks: {
    'pages:extend' (pages) {
      // add a route
      pages.push({
        name: 'profile',
        path: '/profile',
        file: '~/extra-pages/profile.vue'
      })

      // remove routes
      function removePagesMatching (pattern: RegExp, pages: NuxtPage[] = []) {
        const pagesToRemove = []
        for (const page of pages) {
          if (pattern.test(page.file)) {
            pagesToRemove.push(page)
          } else {
            removePagesMatching(pattern, page.children)
          }
        }
        for (const page of pagesToRemove) {
          pages.splice(pages.indexOf(page), 1)
        }
      }
      removePagesMatching(/\.ts$/, pages)
    }
  }
})
```

## Using a module

If you plan to add a whole set of pages related with a specific functionality, you might want to use a Nuxt module.

The Nuxt kit provides a few ways to add routes:

- `extendPages` (callback: `pages => void`)
- `extendRouteRules` (route: string, rule: `NitroRouteConfig`, options: `ExtendRouteRulesOptions`)

# Router Options

It is possible to customize vue-router options.

## Using `app/router.options`

It is possible to customize vue-router options.

This is the recommended way to specify router options.

```
import type { RouterConfig } from '@nuxt/schema'

// https://router.vuejs.org/api/interfaces/routeroptions.html
export default <RouterConfig> {
}
```

`app/router.options.ts`

## Using `nuxt.config`

**Note:** Only JSON serializable options are configurable:

- `linkActiveClass`
- `linkExactActiveClass`
- `end`
- `sensitive`
- `strict`
- `hashMode`

```
export default defineNuxtConfig({
  router: {
```

`nuxt.config`

```
// https://router.vuejs.org/api/interfaces/routeroptions.html
options: {}
}
})
```

## Hash Mode (SPA)

You can enable hash history in SPA mode. In this mode, router uses a hash character (#) before the actual URL that is internally passed. When enabled, the **URL is never sent to the server** and **SSR is not supported**.

```
export default defineNuxtConfig({                                     nuxt.config.ts
  ssr: false,
  router: {
    options: {
      hashMode: true
    }
  }
})
```

## Custom History (advanced)

You can optionally override history mode using a function that accepts the base URL and returns the history mode. If it returns `null` or `undefined`, Nuxt will fallback to the default history.

```
import type { RouterConfig } from '@nuxt/schema'                  app/router.options.ts
import { createMemoryHistory } from 'vue-router'

// https://router.vuejs.org/api/interfaces/routeroptions.html
export default <RouterConfig> {
  history: base => process.client ? createMemoryHistory(base) : null /* default */
}
```



© 2016-2023 Nuxt - MIT  
License

Enterprise

Design Kit  
Nuxt Studio

NuxtLabs

