



Search



[Table of Contents](#) >

defineNuxtRouteMiddleware

Create named route middleware using `defineNuxtRouteMiddleware` helper function.

Route middleware are stored in the `middleware/` directory of your Nuxt application (unless set otherwise).

Type

```
defineNuxtRouteMiddleware(middleware: RouteMiddleware) => RouteMiddleware

interface RouteMiddleware {
  (to: RouteLocationNormalized, from: RouteLocationNormalized): ReturnType<NavigationGuard>
}
```

Parameters

middleware

- **Type:** `RouteMiddleware`

A function that takes two Vue Router's route location objects as parameters: the next route `to` as the first, and the current route `from` as the second.

Learn more about available properties of `RouteLocationNormalized` in the [Vue Router docs](#).

Examples

Showing Error Page

You can use route middleware to throw errors and show helpful error messages:

```
export default defineNuxtRouteMiddleware((to) => {middleware/error.ts  
  if (to.params.id === '1') {  
    throw createError({ statusCode: 404, statusMessage: 'Page Not Found' })  
  }  
})
```

The above route middleware will redirect a user to the custom error page defined in the `~/error.vue` file, and expose the error message and code passed from the middleware.

Redirection

Use `useState` in combination with `navigateTo` helper function inside the route middleware to redirect users to different routes based on their authentication status:

```
export default defineNuxtRouteMiddleware((to, from) => {middleware/auth.ts  
  const auth = useState('auth')  
  
  if (!auth.value.isAuthenticated) {  
    return navigateTo('/login')  
  }  
  
  if (to.path !== '/dashboard') {  
    return navigateTo('/dashboard')  
  }  
})
```

Both `navigateTo` and `abortNavigation` are globally available helper functions that you can use inside `defineNuxtRouteMiddleware`.



© 2016-2023 Nuxt - MIT
License

Enterprise

Design Kit
Nuxt Studio

NuxtLabs

