Register 7



Plugins and Middleware

Plugins

Plugins now have a different format, and take only one argument (nuxtApp). Read more in the docs.

```
Nuxt 2 Nuxt 3
```

```
export default (ctx, inject) => {
  inject('injected', () => 'my injected function')
})
```

You can read more about the format of nuxtApp in the docs.

Migration

- 1. Migrate your plugins to use the defineNuxtPlugin helper function.
- 2. Remove any entries in your nuxt.config plugins array that are located in your plugins/ folder. All files in this directory at the top level (and any index files in any subdirectories) will be automatically registered. Instead of setting mode to client or server, you can indicate this in the file name. For example, ~/plugins/my-plugin.client.ts will only be loaded on client-side.

Route Middleware

Route middleware has a different format.

Nuxt 2 Nuxt 3

```
export default function ({ store, redirect }) {
   // If the user is not authenticated
   if (!store.state.authenticated) {
      return redirect('/login')
   }
}
```

Much like Nuxt 2, route middleware placed in your ~/middleware folder is automatically registered. You can then specify it by name in a component. However, this is done with definePageMeta rather than as a component option.

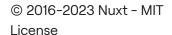
navigateTo is one of a number of route helper functions, which you can read more about in the documentation about route middleware.

Migration

- 1. Migrate your route middleware to use the defineNuxtRouteMiddleware helper function.
- 2. Any global middleware (such as in your nuxt.config) can be placed in your ~/middleware folder with a .global extension, for example ~/middleware/auth.global.ts .

Edit on Github





Enterprise Design Kit NuxtLabs

Nuxt Studio





