# Plugins and Middleware

## New Plugins Format

You can now migrate to the Nuxt 3 plugins API, which is slightly different in format from Nuxt 2.

Plugins now take only one argument ( `nuxtApp` ). You can find out more in the docs.

```
export default defineNuxtPlugin(nuxtApp => {
  nuxtApp.provide('injected', () => 'my injected function')
  // now available on `nuxtApp.$injected`
})
```

> If you want to use the new Nuxt composables (such as `useNuxtApp` or `useRuntimeConfig` ) within your plugins, you will need to use the `defineNuxtPlugin` helper for those plugins.

> Although a compatibility interface is provided via `nuxtApp.vueApp` you should avoid registering plugins, directives, mixins or components this way without adding your own logic to ensure they are not installed more than once, or this may cause a memory leak.

## New Middleware Format

You can now migrate to the Nuxt 3 middleware API, which is slightly different in format from Nuxt 2.

Middleware now take only two argument ( `to` , `from` ). You can find out more in the docs.

```
export default defineNuxtRouteMiddleware((to) => {
  if (to.path !== '/') {
    return navigateTo('/')
  }
})
```

Use of `defineNuxtRouteMiddleware` is not supported outside of the middleware directory.

Nuxt Bridge does not support `definePageMeta` .

Edit on Github