



# NuxtApp

In Nuxt 3, you can access runtime app context within composables, components and plugins. In Nuxt 2, this was referred to as Nuxt context.

## Accessing NuxtApp

Within composables, plugins and components you can access `nuxtApp` with `useNuxtApp` :

```
function useMyComposable () {  
  const nuxtApp = useNuxtApp()  
  // access runtime nuxt app instance  
}
```

Plugins also receive `nuxtApp` as the first argument for convenience. [Read more about plugins.](#)

👉 `useNuxtApp` **(on the server) only works during** `setup` , **inside Nuxt plugins or** `Lifecycle Hooks` .

## Providing Helpers

You can provide helpers to be usable across all composables and application. This usually happens within a Nuxt plugin.


```
const nuxtApp = useNuxtApp()  
nuxtApp.provide('hello', (name) => `Hello ${name}!`)
```

```
console.log(nuxtApp.$hello('name')) // Prints "Hello name!"
```

In Nuxt 2 plugins, this was referred to as inject function.



It is possible to inject helpers by returning an object with a `provide` key. See the [plugins documentation](#) for more information.

 [Edit on Github](#)



© 2016–2023 Nuxt – MIT  
License

Enterprise

Design Kit  
Nuxt Studio

NuxtLabs

