Table of Contents ›

`reloadNuxtApp`

`reloadNuxtApp` will perform a hard reload of your app, re-requesting a page and its dependencies from the server.

By default, it will also save the current `state` of your app (that is, any state you could access with `useState`). You can enable experimental restoration of this state by enabling the `experimental.restoreState` option in your `nuxt.config` file.

Type

```
reloadNuxtApp(options?: ReloadNuxtAppOptions)

interface ReloadNuxtAppOptions {
  ttl?: number
  force?: boolean
  path?: string
  persistState?: boolean
}
```

`options` (optional)

Type: `ReloadNuxtAppOptions`

An object accepting the following properties:

- `path` (optional)
  Type: `string`

Default: `window.location.pathname`
The path to reload (defaulting to the current path). If this is different from the current window location it will trigger a navigation and add an entry in the browser history.

- `ttl` (optional)
  Type: `number`
  Default: `10000`
  The number of milliseconds in which to ignore future reload requests. If called again within this time period, `reloadNuxtApp` will not reload your app to avoid reload loops.

- `force` (optional)
  Type: `boolean`
  Default: `false`
  This option allows bypassing reload loop protection entirely, forcing a reload even if one has occurred within the previously specified TTL.

- `persistState` (optional)
  Type: `boolean`
  Default: `false`
  Whether to dump the current Nuxt state to sessionStorage (as `nuxt:reload:state`). By default this will have no effect on reload unless `experimental.restoreState` is also set, or unless you handle restoring the state yourself.

Edit on Github