UIKIT 2.0

Menu On this page

# useMessage

Documentación y ejemplos del composable `useMessage` . Pensado para **mostrar mensajes** en los componentes alert y toast.

## messageToastDefault

Composable pensado para **activar mensajes simples** dentro del componente toast.

### Parámetros

- `store` → Dentro del store actual, debes definir las constantes `errorBack` y `messageToast` para controlar el error backend y definir el mensaje que se mostrará.
- `delay` → Define el tiempo en que el toast estará visible.
- `message` → Debes definir el objeto `{ title: String, message: String }`

ts

```ts
import { utils } from 'uikit-3it-vue'

const { messageToastDefault } = composables.useMessage()

messageToastDefault(store, delay, message)
```

## messageToastCreateUpdate

Composable pensado para **activar mensajes de creación o actualización** de un registro del componente toast.

### Parámetros

- `store` → Dentro del store actual, debes definir las constantes `errorBack` y `messageToast` para controlar el error backend y definir el mensaje que se mostrará.

- `id` → La ID del registro que será actualizado. Si es creación, este valor no debe incluirse.
- `delay` → Define el tiempo en que el toast estará visible.
- `message` → Debes definir el objeto `{ title: String, message: String }`

```ts
import { utils } from 'uikit-3it-vue'

const { messageToastCreateUpdate } = composables.useMessage()

messageToastCreateUpdate(store, id, delay, message)
```

## messageToastEnableDisabled

Composable pensado para activar mensajes de **habilitar o deshabilitar** un registro en el componente [toast](#).

### Parámetros

- `store` → Dentro del store actual, debes definir las constantes `errorBack` y `messageToast` para controlar el error backend y definir el mensaje que se mostrará.
- `status` → Valor `boolean` actual del estado del registro a modificar.
- `delay` → Define el tiempo en que el toast estará visible.
- `message` → Debes definir el objeto `{ title: String, message: String }`

```ts
import { utils } from 'uikit-3it-vue'

const { messageToastEnableDisabled } = composables.useMessage()

messageToastEnableDisabled(store, status, delay, message)
```

## messageAlertDefault

Composable pensado para mostrar **alertas explicativas o de error** dentro de los formularios.

### Parámetros

- `store` → Dentro del store, debes definir la constante `messageAlert` para definir el mensaje que se mostrará dentro del componente alert.
- `error` → Valor `boolean` del error global de validación en el formulario.
- `message` → Debes definir el objeto `{ title: String, message: String }`

ts

```ts
import { utils } from 'uikit-3it-vue'

const { messageAlertDefault } = composables.useMessage()

messageAlertDefault(store, error, message)
```

## messageAlertDialogDefault

Composable pensado para mostrar **alertas explicativas o de error** dentro de los formularios que se mostrarán en el componente dialog.

### Parámetros

- `store` → Dentro del store, debes definir la constante `messageDialogAlert` para definir el mensaje que se mostrará dentro del componente alert.
- `error` → Valor `boolean` del error global de validación en el formulario.
- `message` → Debes definir el objeto `{ title: String, message: String }`

ts

```ts
import { utils } from 'uikit-3it-vue'

const { messageAlertDialogDefault } = composables.useMessage()

messageAlertDialogDefault(store, error, message)
```

## messageAlertEnableDisabled

Composable pensado para mostrar **alertas explicativas al intentar habilitar o deshabilitar** un registro con el valor `status` del registro dentro del componente dialog.

### Parámetros

- `store` → Dentro del store, debes definir la constante `messageDialogAlert` para definir el mensaje que se mostrará dentro del componente alert.
- `status` → Valor `boolean` actual del estado del registro a modificar.
- `message` → Debes definir el objeto `{ title: String, message: String }`

ts

```ts
import { utils } from 'uikit-3it-vue'

const { messageAlertEnableDisabled } = composables.useMessage()

messageAlertEnableDisabled(store, status, message)
```

Previous page
useLogos

Next page
useRequired