

Display a notification message globally.

When To Use

To display a notification message at any of the four corners of the viewport. Typically it can be used in the following cases:

- A notification with complex content.
- A notification providing a feedback based on the user interaction. Or it may show some details about upcoming steps the user may have to follow.
- A notification that is pushed by the application.

API

- `notification.success(config)`
- `notification.error(config)`
- `notification.info(config)`
- `notification.warning(config)`
- `notification.warn(config)`
- `notification.open(config)`
- `notification.close(key: String)`
- `notification.destroy()`

The properties of config are as follows:

Property	Description	Type	Default
bottom	Distance from the bottom of the viewport, when placement is <code>bottomRight</code> or <code>bottomLeft</code> (unit: pixels)	number	24
btn	Customized close button	ReactNode	-
className	Customized CSS class	string	-
closeIcon	Custom close icon	ReactNode	-
description	The content of notification box (required)	ReactNode	-
duration	Time in seconds before Notification is closed. When set to 0 or null, it will never be closed automatically	number	4.5
getContainer	Return the mount node for Notification	() => HTMLNode	() => document.body
icon	Customized icon	ReactNode	-
key	The unique identifier of the Notification	string	-
message	The title of notification box (required)	ReactNode	-
placement	Position of Notification, can be one of <code>topLeft</code> <code>topRight</code> <code>bottomLeft</code> <code>bottomRight</code>	string	<code>topRight</code>
style	Customized inline style	CSSProperties	-
top	Distance from the top of the viewport, when placement	number	24

	is <code>topRight</code> or <code>topLeft</code> (unit: pixels)		
<code>onClick</code>	Specify a function that will be called when the notification is clicked	function	-
<code>onClose</code>	Trigger when notification closed	function	-

`notification` also provides a global `config()` method that can be used for specifying the default options. Once this method is used, all the notification boxes will take into account these globally defined options when displaying.

- `notification.config(options)`

When you use `ConfigProvider` for global configuration, the system will automatically start RTL mode by default.(4.3.0+)

When you want to use it alone, you can start the RTL mode through the following settings.

```
notification.config({
  placement: 'bottomRight',
  bottom: 50,
  duration: 3,
  rtl: true,
});
```

Property	Description	Type	Default	Version
<code>bottom</code>	Distance from the bottom of the viewport, when <code>placement</code> is <code>bottomRight</code> or <code>bottomLeft</code> (unit: pixels)	number	24	
<code>closeIcon</code>	Custom close icon	ReactNode	-	
<code>duration</code>	Time in seconds before Notification is closed. When set to 0 or null, it will never be closed automatically	number	4.5	
<code>getContainer</code>	Return the mount node for Notification	() => HTMLNode	() => document.body	
<code>placement</code>	Position of Notification, can be one of <code>topLeft</code> <code>topRight</code> <code>bottomLeft</code> <code>bottomRight</code>	string	<code>topRight</code>	
<code>rtl</code>	Whether to enable RTL mode	boolean	false	
<code>top</code>	Distance from the top of the viewport, when <code>placement</code> is <code>topRight</code> or <code>topLeft</code> (unit: pixels)	number	24	
<code>maxCount</code>	Max Notification show, drop oldest if exceed limit	number	-	4.17.0

FAQ

Why I can not access context, redux, ConfigProvider locale/prefixCls in notification?

antd will dynamic create React instance by `ReactDOM.render` when call notification methods. Whose context is different with origin code located context.

When you need context info (like ConfigProvider context), you can use `notification.useNotification` to get `api` instance and `contextHolder` node. And put it in your children:

```
const [api, contextHolder] = notification.useNotification();

return (
  <Context1.Provider value="Ant">
    { /* contextHolder is inside Context1 which means api will get value of Context1 */ }
    {contextHolder}
    <Context2.Provider value="Design">
      { /* contextHolder is outside Context2 which means api will **not** get value of Context2 */ }
    </Context2.Provider>
  </Context1.Provider>
);
```

Note: You must insert `contextHolder` into your children with hooks. You can use origin method if you do not need context connection.

How to set static methods prefixCls ?

You can config with [ConfigProvider.config](#)