

Display global messages as feedback in response to user operations.

## When To Use

- To provide feedback such as success, warning, error etc.
- A message is displayed at top and center and will be dismissed automatically, as a non-interrupting light-weighted prompt.

## API

This components provides some static methods, with usage and arguments as following:

- `message.success(content, [duration], onClose)`
- `message.error(content, [duration], onClose)`
- `message.info(content, [duration], onClose)`
- `message.warning(content, [duration], onClose)`
- `message.warn(content, [duration], onClose)` // alias of warning
- `message.loading(content, [duration], onClose)`

Argument	Description	Type	Default
content	The content of the message	ReactNode   config	-
duration	Time(seconds) before auto-dismiss, don't dismiss if set to 0	number	1.5
onClose	Specify a function that will be called when the message is closed	function	-

`afterClose` can be called in thenable interface:

- `message[level](content, [duration]).then(afterClose)`
- `message[level](content, [duration], onClose).then(afterClose)`

where `level` refers one static methods of `message`. The result of `then` method will be a Promise.

Supports passing parameters wrapped in an object:

- `message.open(config)`
- `message.success(config)`
- `message.error(config)`
- `message.info(config)`
- `message.warning(config)`
- `message.warn(config)` // alias of warning
- `message.loading(config)`

The properties of config are as follows:

Property	Description	Type	Default
className	Customized CSS class	string	-
content	The content of the message	ReactNode	-
duration	Time(seconds) before auto-dismiss, don't dismiss if set to 0	number	3

icon	Customized Icon	ReactNode	-
key	The unique identifier of the Message	string   number	-
style	Customized inline style	<a href="#">CSSProperties</a>	-
onClick	Specify a function that will be called when the message is clicked	function	-
onClose	Specify a function that will be called when the message is closed	function	-

## Global static methods

Methods for global configuration and destruction are also provided:

- `message.config(options)`
- `message.destroy()`

use `message.destroy(key)` to remove a message.

### message.config

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.

```
message.config({
  top: 100,
  duration: 2,
  maxCount: 3,
  rtl: true,
  prefixCls: 'my-message',
});
```

Argument	Description	Type	Default	Version
duration	Time before auto-dismiss, in seconds	number	3	
getContainer	Return the mount node for Message	() => HTMLElement	() => document.body	
maxCount	Max message show, drop oldest if exceed limit	number	-	
prefixCls	The prefix className of message node	string	ant-message	4.5.0
rtl	Whether to enable RTL mode	boolean	false	
top	Distance from top	number	8	

## FAQ

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

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

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

```
const [api, contextHolder] = message.useMessage();

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](#)