Modal dialogs.

When To Use

When requiring users to interact with the application, but without jumping to a new page and interrupting the user's workflow, you can use `Modal` to create a new floating layer over the current page to get user feedback or display information. Additionally, if you need show a simple confirmation dialog, you can use `antd.Modal.confirm()`, and so on.

API

Property	Description	Type	Default	Version
afterClose	Specify a function that will be called when modal is closed completely	function	-	
bodyStyle	Body style for modal body element. Such as height, padding etc	CSSProperties		
cancelButtonProps	The cancel button props	ButtonProps	-	
cancelText	Text of the Cancel button	ReactNode	`Cancel`	
centered	Centered Modal	boolean	false	
closable	Whether a close (x) button is visible on top right of the modal dialog or not	boolean	true	
closeIcon	Custom close icon	ReactNode	<CloseOutlined />	
confirmLoading	Whether to apply loading visual effect for OK button or not	boolean	false	
destroyOnClose	Whether to unmount child components on onClose	boolean	false	
focusTriggerAfterClose	Whether need to focus trigger element after dialog is closed	boolean	true	4.9.0
footer	Footer content, set as `footer={null}` when you don't need default buttons	ReactNode	(OK and Cancel buttons)	
forceRender	Force render Modal	boolean	false	
getContainer	Return the mount node for Modal	HTMLElement | () => HTMLElement | Selectors | false	document.body	

keyboard	Whether support press esc to close	boolean	true	
mask	Whether show mask or not	boolean	true	
maskClosable	Whether to close the modal dialog when the mask (area outside the modal) is clicked	boolean	true	
maskStyle	Style for modal's mask element	CSSProperties		
modalRender	Custom modal content render	(node: ReactNode) => ReactNode	-	4.7.0
okButtonProps	The ok button props	ButtonProps	-	
okText	Text of the OK button	ReactNode	`OK`	
okType	Button `type` of the OK button	string	`primary`	
style	Style of floating layer, typically used at least for adjusting the position	CSSProperties	-	
title	The modal dialog's title	ReactNode	-	
visible	Whether the modal dialog is visible or not	boolean	false	
width	Width of the modal dialog	string | number	520	
wrapClassName	The class name of the container of the modal dialog	string	-	
zIndex	The `z-index` of the Modal	number	1000	
onCancel	Specify a function that will be called when a user clicks mask, close button on top right or Cancel button	function(e)	-	
onOk	Specify a function that will be called when a user clicks the OK button	function(e)	-	

Note

- The state of Modal will be preserved at it's component lifecycle by default, if you wish to open it with a brand new state everytime, set `destroyOnClose` on it.
- There is a situation that using `<Modal />` with Form, which won't clear fields value when closing Modal even you have set `destroyOnClose`. You need `<Form preserve={false} />` in this case.
- `Modal.method()` RTL mode only supports hooks.

Modal.method()

There are five ways to display the information based on the content's nature:

- `Modal.info`
- `Modal.success`
- `Modal.error`
- `Modal.warning`
- `Modal.confirm`

The items listed above are all functions, expecting a settings object as parameter. The properties of the object are follows:

Property	Description	Type	Default	Version
afterClose	Specify a function that will be called when modal is closed completely	function	-	4.9.0
autoFocusButton	Specify which button to autofocus	null | `ok` | `cancel`	`ok`	
bodyStyle	Body style for modal body element. Such as height, padding etc	CSSProperties		4.8.0
cancelButtonProps	The cancel button props	ButtonProps	-	
cancelText	Text of the Cancel button with Modal.confirm	string	`Cancel`	
centered	Centered Modal	boolean	false	
className	The className of container	string	-	
closable	Whether a close (x) button is visible on top right of the confirm dialog or not	boolean	false	4.9.0
closeIcon	Custom close icon	ReactNode	undefined	4.9.0
content	Content	ReactNode	-	
getContainer	Return the mount node for Modal	HTMLElement | () => HTMLElement | Selectors | false	document.body	
icon	Custom icon	ReactNode	<QuestionCircle />	
keyboard	Whether support press esc to close	boolean	true	
mask	Whether show mask or not.	boolean	true	
maskClosable	Whether to close the modal dialog	boolean	false	

	when the mask (area outside the modal) is clicked			
maskStyle	Style for modal's mask element	object	{}	
okButtonProps	The ok button props	ButtonProps	-	
okText	Text of the OK button	string	`OK`	
okType	Button `type` of the OK button	string	`primary`	
style	Style of floating layer, typically used at least for adjusting the position	CSSProperties	-	
title	Title	ReactNode	-	
width	Width of the modal dialog	string | number	416	
wrapClassName	The class name of the container of the modal dialog	string	-	4.18.0
zIndex	The `z-index` of the Modal	number	1000	
onCancel	Specify a function that will be called when the user clicks the Cancel button. The parameter of this function is a function whose execution should include closing the dialog. If the function does not take any parameter (`!onCancel.length`) then modal dialog will be closed unless returned value is `true` (`!!onCancel()`). You can also just return a promise and when the promise is resolved, the modal dialog will also be closed	function(close)	-	
onOk	Specify a function that will be called when the user clicks the OK button. The parameter of this function is a function whose execution should include closing the dialog. If the function does not take any parameter (`!onOk.length`) then modal dialog will be closed unless returned value is `true` (`!!onOk()`). You can also just return a promise and when the promise is resolved, the modal dialog will also be closed	function(close)	-	

All the `Modal.method` s will return a reference, and then we can update and close the modal dialog by the reference.

```
const modal = Modal.info();

modal.update({
  title: 'Updated title',
  content: 'Updated content',
});

// on 4.8.0 or above, you can pass a function to update modal
modal.update(prevConfig => ({
  ...prevConfig,
  title: `${prevConfig.title} (New)`,
}));

modal.destroy();
```

- `Modal.destroyAll`

`Modal.destroyAll()` could destroy all confirmation modal dialogs(`Modal.confirm|success|info|error|warning`). Usually, you can use it in router change event to destroy confirm modal dialog automatically without use modal reference to close(it's too complex to use for all modal dialogs)

```
import { browserHistory } from 'react-router';

// router change
browserHistory.listen(() => {
  Modal.destroyAll();
});
```

Modal.useModal()

When you need using Context, you can use `contextHolder` which created by `Modal.useModal` to insert into children. Modal created by hooks will get all the context where `contextHolder` are. Created `modal` has the same creating function with `Modal.method` .

```
const [modal, contextHolder] = Modal.useModal();

React.useEffect(() => {
  modal.confirm({
    // ...
  });
}, []);

return <div>{contextHolder}</div>;
```

FAQ

Why I can not access context, redux, ConfigProvider `locale/prefixCls` in Modal.xxx?

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

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

```
const [modal, contextHolder] = Modal.useModal();

// then call modal.confirm instead of Modal.confirm

return (
  <Context1.Provider value="Ant">
    {/* contextHolder is in Context1, which means modal will get context of Context1
*/}
    {contextHolder}
    <Context2.Provider value="Design">
      {/* contextHolder is out of Context2, which means modal will not get context
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 disable motion?

You can config `transitionName=""` and `maskTransitionName=""` to remove motion class. But you should note that these prop is internal usage which we don't promise exist in next major version.

How to set static methods prefixCls ?

You can config with `ConfigProvider.config`