

When To Use

- Provide reset styles based on `.ant-app` element.
- You could use static methods of `message/notification/Modal` from `useApp` without writing `contextHolder` manually.

Examples

Basic

```
import React from 'react';
import { App, Button, Space } from 'antd';

// Sub page
const MyPage = () => {
  const { message, modal, notification } = App.useApp();

  const showMessage = () => {
    message.success('Success!');
  };

  const showModal = () => {
    modal.warning({
      title: 'This is a warning message',
      content: 'some messages...some messages...',
    });
  };

  const showNotification = () => {
    notification.info({
      message: 'Notification topLeft',
      description: 'Hello, Ant Design!!',
      placement: 'topLeft',
    });
  };

  return (
    <Space wrap>
      <Button type="primary" onClick={showMessage}>
        Open message
      </Button>
      <Button type="primary" onClick={showModal}>
        Open modal
      </Button>
      <Button type="primary" onClick={showNotification}>
        Open notification
      </Button>
    </Space>
  );
};
```

```

    </Space>
  );
};

// Entry component
export default () => (
  <App>
    <MyPage />
  </App>
);

```

Hooks config

```

import React from 'react';
import { App, Button, Space } from 'antd';

// Sub page
const MyPage = () => {
  const { message, notification } = App.useApp();

  const showMessage = () => {
    message.success('Success!');
  };

  const showNotification = () => {
    notification.info({
      message: 'Notification',
      description: 'Hello, Ant Design!!',
    });
  };

  return (
    <Space wrap>
      <Button type="primary" onClick={showMessage}>
        Message for only one
      </Button>
      <Button type="primary" onClick={showNotification}>
        Notification for bottomLeft
      </Button>
    </Space>
  );
};

// Entry component
export default () => (
  <App message={{ maxCount: 1 }} notification={{ placement: 'bottomLeft'

```

```

}}>
  <MyPage />
</App>
);

```

How to use

Basic usage

App provides upstream and downstream method calls through `Context`, because `useApp` needs to be used as a subcomponent, we recommend encapsulating App at the top level in the application.

```

import React from 'react';
import { App } from 'antd';

const MyPage: React.FC = () => {
  const { message, notification, modal } = App.useApp();
  message.success('Good!');
  notification.info({ message: 'Good' });
  modal.warning({ title: 'Good' });
  // ....
  // other message, notification, modal static function
  return <div>Hello word</div>;
};

const MyApp: React.FC = () => (
  <App>
    <MyPage />
  </App>
);

export default MyApp;

```

Note: `App.useApp` must be available under `App`.

Sequence with ConfigProvider

The App component can only use the token in the `ConfigProvider`, if you need to use the Token, the `ConfigProvider` and the `App` component must appear in pairs.

```

<ConfigProvider theme={{ ... }}>
  <App>
    ...
  </App>
</ConfigProvider>

```

Embedded usage scenarios (if not necessary, try not to do nesting)

```
<App>
  <Space>
    ...
    <App>...</App>
  </Space>
</App>
```

Global scene (redux scene)

```
// Entry component
import { App } from 'antd';
import type { MessageInstance } from 'antd/es/message/interface';
import type { ModalStaticFunctions } from 'antd/es/modal/confirm';
import type { NotificationInstance } from 'antd/es/notification/interface';

let message: MessageInstance;
let notification: NotificationInstance;
let modal: Omit<ModalStaticFunctions, 'warn'>;

export default () => {
  const staticFunction = App.useApp();
  message = staticFunction.message;
  modal = staticFunction.modal;
  notification = staticFunction.notification;
  return null;
};

export { message, modal, notification };
```

```
// sub page
import React from 'react';
import { Button, Space } from 'antd';

import { message } from './store';

export default () => {
  const showMessage = () => {
    message.success('Success!');
  };

  return (
    <Space>
```

```
    <Button type="primary" onClick={showMessage}>
      Open message
    </Button>
  </Space>
);
};
```

API

Common props ref: [Common props](#)

This component is available since `antd@5.1.0`.

App

Property	Description	Type	Default	Version
component	Config render element, if <code>false</code> will not create DOM node	ComponentType <code>false</code>	<code>div</code>	5.11.0
message	Global config for Message	MessageConfig	-	5.3.0
notification	Global config for Notification	NotificationConfig	-	5.3.0

Design Token

FAQ

CSS Var doesn't work inside `<App component={false}>`

Make sure the App `component` is a valid html tag, so when you're turning on CSS variables, there's a container to hold the CSS class name. If not set, it defaults to the `div` tag. If set to `false`, no additional DOM nodes will be created, and no default styles will be provided.