

# Result

Used to feed back the results of a series of operational tasks.

## When To Use

Use when important operations need to inform the user to process the results and the feedback is more complicated.

## Examples



## Successfully Purchased Cloud Server ECS!

Order number: 2017182818828182881 Cloud server configuration takes 1-5 minutes, please wait.

[Go Console](#)[Buy Again](#)

### Success

Show successful results.

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

const App: React.FC = () => (
  <Result
    status="success"
    title="Successfully Purchased Cloud Server ECS!"
    subTitle="Order number: 2017182818828182881 Cloud server configuration takes 1-5 minutes, please wait."
    extra={[
      <Button type="primary" key="console">
        Go Console
      </Button>,
      <Button key="buy">Buy Again</Button>,
    ]}
  />
);

export default App;
```



Your operation has been executed

[Go Console](#)

Info 

Show processing results.

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

const App: React.FC = () => (
  <Result
    title="Your operation has been executed"
    extra={
      <Button type="primary" key="console">
        Go Console
      </Button>
    }
  />
);

export default App;
```



There are some problems with your operation.

[Go Console](#)

#### Warning

The result of the warning.

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

const App: React.FC = () => (
  <Result
    status="warning"
    title="There are some problems with your operation."
    extra={
      <Button type="primary" key="console">
        Go Console
      </Button>
    }
  />
);

export default App;
```



# 403

Sorry, you are not authorized to access this page.

[Back Home](#)

403 [✎](#)

you are not authorized to access this page.

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

const App: React.FC = () => (
  <Result
    status="403"
    title="403"
    subTitle="Sorry, you are not authorized to access this page."
    extra={<Button type="primary">Back Home</Button>}
  />
);

export default App;
```



# 404

Sorry, the page you visited does not exist.

[Back Home](#)

404 [✎](#)

The page you visited does not exist.

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

const App: React.FC = () => (
  <Result
    status="404"
    title="404"
    subTitle="Sorry, the page you visited does not exist."
    extra={<Button type="primary">Back Home</Button>}
  />
);

export default App;
```



# 500

Sorry, something went wrong.

[Back Home](#)

500 [✎](#)

Something went wrong on server.

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

const App: React.FC = () => (
  <Result
    status="500"
    title="500"
    subTitle="Sorry, something went wrong."
    extra={<Button type="primary">Back Home</Button>}
  />
);

export default App;
```



## Submission Failed

Please check and modify the following information before resubmitting.

[Go Console](#)[Buy Again](#)

### The content you submitted has the following error:

- ⊗ Your account has been frozen. [Thaw immediately >](#)
- ⊗ Your account is not yet eligible to apply. [Apply Unlock >](#)

#### Error

Complex error feedback.

```
import React from 'react';
import { CloseCircleOutlined } from '@ant-design/icons';
import { Button, Result, Typography } from 'antd';

const { Paragraph, Text } = Typography;

const App: React.FC = () => (
  <Result
    status="error"
    title="Submission Failed"
    subTitle="Please check and modify the following information before resubmitting."
    extra={[
      <Button type="primary" key="console">
        Go Console
      </Button>,
      <Button key="buy">Buy Again</Button>,
    ]}
  >
    <div className="desc">
      <Paragraph>
        <Text
          strong
          style={{
            fontSize: 16,
          }}
        >
          The content you submitted has the following error:
        </Text>
      </Paragraph>
      <Paragraph>
        <CloseCircleOutlined className="site-result-demo-error-icon" /> Your account has been
        frozen. <a>Thaw immediately &gt;</a>
      </Paragraph>
      <Paragraph>
        <CloseCircleOutlined className="site-result-demo-error-icon" /> Your account is not yet
        eligible to apply. <a>Apply Unlock &gt;</a>
      </Paragraph>
    </div>
  </Result>
);
```





Great, we have done all the operations!

Next

#### Custom icon

Custom icon.

```
import React from 'react';
import { SmileOutlined } from '@ant-design/icons';
import { Button, Result } from 'antd';

const App: React.FC = () => (
  <Result
    icon={<SmileOutlined />}
    title="Great, we have done all the operations!"
    extra={<Button type="primary">Next</Button>}
  />
);

export default App;
```

## API

Property	Description	Type	Default
extra	Operating area	ReactNode	-
icon	Custom back icon	ReactNode	-
status	Result status, decide icons and colors	<div>success  </div> <div>error  </div> <div>info  </div> <div>warning  </div> <div>404   403  </div> <div>500</div>	<div>info</div>
subTitle	The subTitle	ReactNode	-
title	The title	ReactNode	-

## Design Token

### ▼ Global Token

Token Name	Description	Type	Default Value
colorError	Used to represent the visual elements of the operation failure, such as the error Button, error Result	<div>string</div>	<div><input type="checkbox"/> #ff4d4f</div>

Token Name	Description	Type	Default Value
	component, etc.		
colorFillAlter	Control the alternative background color of element.	string	<code>rgba(0, 0, 0, 0.02)</code>
colorInfo	Used to represent the operation information of the Token sequence, such as Alert, Tag, Progress, and other components use these map tokens.	string	<code>#1677ff</code>
colorSuccess	Used to represent the token sequence of operation success, such as Result, Progress and other components will use these map tokens.	string	<code>#52c41a</code>
colorTextDescription	Control the font color of text description.	string	<code>rgba(0, 0, 0, 0.45)</code>
colorTextHeading	Control the font color of heading.	string	<code>rgba(0, 0, 0, 0.88)</code>
colorWarning	Used to represent the warning map token, such as Notification, Alert, etc. Alert or Control component(like Input) will use these map tokens.	string	<code>#faad14</code>
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	number	14
fontSizeHeading3	Font size of h3 tag.	number	24
lineHeight	Line height of text.	number	1.5714285714285714
lineHeightHeading3	Line height of h3 tag.	number	1.3333333333333333
marginXS	Control the margin of an element, with a small size.	number	8
padding	Control the padding of the element.	number	16
paddingLG	Control the large padding of the element.	number	24
paddingXL	Control the extra large padding of the element.	number	32
paddingXS	Control the extra small padding of the element.	number	8