

QRCode

Components that can convert text into QR codes, and support custom color and logo. Available since `antd@5.1.0`.

If the QR code cannot be scanned for identification, it may be because the link address is too long, which leads to too dense pixels. You can configure the QR code to be larger through size, or shorten the link through short link services.

When To Use

Used when the text needs to be converted into a QR Code.

Examples



<https://ant.design/>

base [✎](#)

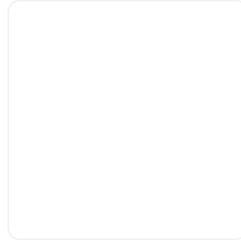
Basic Usage.

```
import { Input, QRCode, Space } from 'antd';
import React from 'react';

const App: React.FC = () => {
  const [text, setText] = React.useState('http

  return (
    <Space direction="vertical" align="center"
      <QRCode value={text || '-'} />
      <Input
        placeholder="-"
        maxLength={60}
        value={text}
        onChange={(e) => setText(e.target.valu
      />
    </Space>
  );
};

export default App;
```



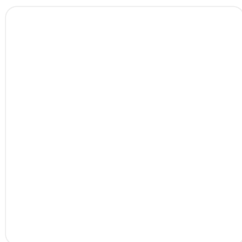
With Icon [✎](#)

QRCode with Icon.

```
import React from 'react';
import { QRCode } from 'antd';

const App: React.FC = () => (
  <QRCode
    errorLevel="H"
    value="https://ant.design/"
    icon="https://gw.alipayobjects.com/zos/rms
  />
);

export default App;
```



QR code expired

[Refresh](#)

other status [✎](#)

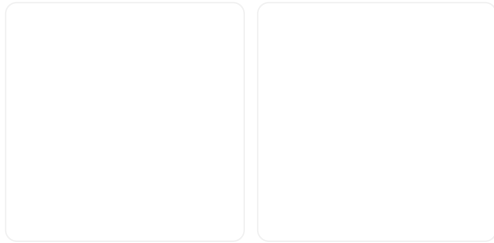
The status can be controlled by the value

`status` .

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

const App: React.FC = () => (
  <Space wrap>
    <QRCode value="https://ant.design/" status
    <QRCode value="https://ant.design/" status
  </Space>
);

export default App;
```



Custom Color

Custom Color.

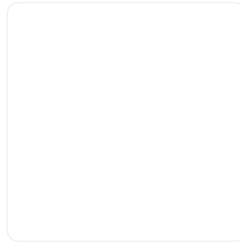
```
import React from 'react';
import { QRCode, Space, theme } from 'antd';

const { useToken } = theme;

const App: React.FC = () => {
  const { token } = useToken();
  return (
    <Space>
      <QRCode
        value="https://ant.design/"
        color={token.colorSuccessText}
        style={{ marginBottom: 16, background:
      />
      <QRCode
        value="https://ant.design/"
        color={token.colorInfoText}
        style={{ marginBottom: 16, background:
      />
    </Space>
  );
};

export default App;
```

— Smaller + Larger



Custom Size

Custom Size.

```
import React, { useState } from 'react';
import { MinusOutlined, PlusOutlined } from '@ant-design/icons';
import { QRCode, Button } from 'antd';

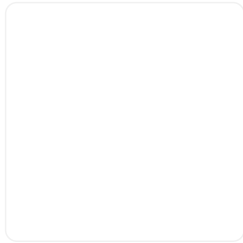
const App: React.FC = () => {
  const [size, setSize] = useState<number>(160);

  const increase = () => {
    setSize((prevSize) => {
      const newSize = prevSize + 10;
      if (newSize > 300) {
        return 300;
      }
      return newSize;
    });
  };

  const decline = () => {
    setSize((prevSize) => {
      const newSize = prevSize - 10;
      if (newSize < 48) {
        return 48;
      }
      return newSize;
    });
  };

  return (
    <>
      <Button.Group style={{ marginBottom: 16 }}>
        <Button onClick={decline} disabled={size === 48}>
          Smaller
        </Button>
        <Button onClick={increase} disabled={size === 300}>
          Larger
        </Button>
      </Button.Group>
      <QRCode
        errorLevel="H"
        size={size}
        iconSize={size / 4}
        value="https://ant.design/"
        icon="https://gw.alipayobjects.com/zos/antfincdn/AJClQ7eq1r1B8k1Vrt8e98/AntDesignLogo.png"
      />
    </>
  );
};

export default App;
```



L M Q H

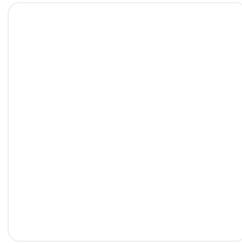
Error Level

set Error Level.

```
import React, { useState } from 'react';
import type { QRCodeProps } from 'antd';
import { Segmented, QRCode } from 'antd';

const App: React.FC = () => {
  const [level, setLevel] = useState<string |
  return (
    <
      <QRCode
        style={{ marginBottom: 16 }}
        errorLevel={level as QRCodeProps['errc
        value="https://gw.alipayobjects.com/zc
      />
      <Segmented options={['L', 'M', 'Q', 'H']
    </>
  );
};

export default App;
```



Download

Download QRCode

A way to download QRCode.

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

const downloadQRCode = () => {
  const canvas = document.getElementById('myqr
  if (canvas) {
    const url = canvas.toDataURL();
    const a = document.createElement('a');
    a.download = 'QRCode.png';
    a.href = url;
    document.body.appendChild(a);
    a.click();
    document.body.removeChild(a);
  }
};

const App: React.FC = () => (
  <div id="myqrcode">
    <QRCode value="https://ant.design/" style=
    <Button type="primary" onClick={downloadQF
      Download
    </Button>
  </div>
);

export default App;
```



Advanced Usage [✎](#)

With Popover.

```
import React from 'react';
import { QRCode, Popover } from 'antd';

const src = 'https://gw.alipayobjects.com/zos/';

const App: React.FC = () => (
  <Popover overlayInnerStyle={{ padding: 0 }}>
    <img width={100} height={100} src={src} alt="QR Code" />
  </Popover>
);

export default App;
```

API

This component is available since `antd@5.1.0`

Property	Description	Type	Default
value	scanned text	string	–
icon	include image url (only image link are supported)	string	–
size	QRCode size	number	128
iconSize	include image size	number	32
color	QRCode Color	string	#000
bordered	Whether has border style	boolean	true
errorLevel	Error Code Level	'L' 'M' 'Q' 'H'	M
status	QRCode status	active expired loading	active
onRefresh	callback	() => void	–

Design Token

▼ Global Token

Token Name	Description	Type	Default Value
colorSplit	Used as the color of separator, this color is the same as colorBorderSecondary but with	string	rgba(5, 5, 5, 0.06)

Token Name	Description	Type	Default Value
	transparency.		
colorText	Default text color which comply with W3C standards, and this color is also the darkest neutral color.	string	<code>rgba(0, 0, 0, 0.88)</code>
colorWhite	Pure white color don't changed by theme	string	<code>#fff</code>
borderRadiusLG	LG size border radius, used in some large border radius components, such as Card, Modal and other components.	number	8
controlHeight	The height of the basic controls such as buttons and input boxes in Ant Design	number	32
fontFamily	The font family of Ant Design prioritizes the default interface font of the system, and provides a set of alternative font libraries that are suitable for screen display to maintain the readability and readability of the font under different platforms and browsers, reflecting the friendly, stable and professional characteristics.	string	-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, 'Helvetica Neue', Arial, 'Noto Sans', sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji'
fontSize	The most widely used font size in the design system, from which the text gradient will be derived.	number	14
lineHeight	Line height of text.	number	1.5714285714285714
lineType	Border style of base components	string	solid
lineWidth	Border width of base components	number	1
marginXS	Control the margin of an element, with a small size.	number	8
paddingSM	Control the small padding of the element.	number	12

FAQ

About QRCode ErrorLevel

The ErrorLevel means that the QR code can be scanned normally after being blocked, and the maximum area that can be blocked is the error correction rate.

Generally, the QR code is divided into 4 error correction levels: Level `L` can correct about `7%` errors, Level `M` can correct about `15%` errors, Level `Q` can correct about `25%` errors, and Level `H` can correct about `30%` errors.

When the content encoding of the QR code carries less information, in other words, when the value link is short, set different error correction levels, and the generated image will not change.

For more information, see the: https://www.qrcode.com/en/about/error_correction

