

ConfigProvider

`ConfigProvider` provides a uniform configuration support for components.

Usage

This component provides a configuration to all React components underneath itself via the [context API](#). In the render tree all components will have access to the provided config.

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

// ...
const Demo: React.FC = () => (
  <ConfigProvider direction="rtl">
    <App />
  </ConfigProvider>
);

export default Demo;
```

Content Security Policy

Some components use dynamic style to support wave effect. You can config `csp` prop if Content Security Policy (CSP) is enabled:

```
<ConfigProvider csp={{ nonce: 'YourNonceCode' }}>
  <Button>My Button</Button>
</ConfigProvider>
```

Examples

Change locale of components: English 中文

<

1

2

3

4

5

>

10 / page

□

□

Select date

☐

Select time

◵

Start date

→

End date

☐

Show Modal

Show info

Show confirm


Click to confirm

☐ ∨ 0 item

Q Search here

>

<




No data

☐ ∨ 0 item

Q Search here

>

<



No data

2023

□

May

□

Month

Year

Su	Mo	Tu	We	Th	Fr	Sa
30	01	02	03	04	05	06
07	08	09	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	01	02	03
04	05	06	07	08	09	10

* Username:


Age:

Submit

Name

▼

Age





No data

QR code expired

↻

 Refresh





Uploading...




image.png

Change direction of components:

LTR

RTL

Cascader example

یک مورد انتخاب کنید



With search:

Select an item



Switch example



Radio Group example

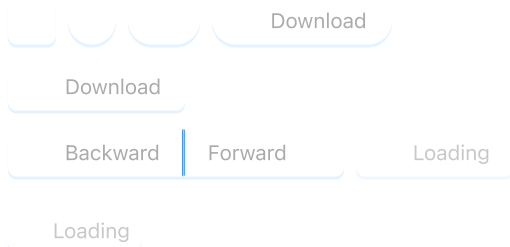
تهران

اصفهان

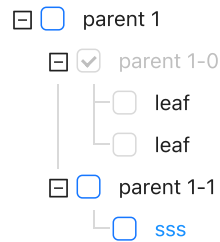
فارس

خوزستان

Button example



Tree example



Input (Input Group) example

0571

26888888

0571

26888888

Option1



input content

input search text

Search

Http://



mysite

.com

Select example

مورچه



مورچه



مورچه



Select a person



TreeSelect example

Please select



Modal example

Open Modal

Steps example

Finished
This is a description.

In Progress
This is a description.

Waiting
This is a description.

1 Step 1

This is a description.

2 Step 2

This is a description.

3 Step 3

This is a description.

Date example

Badge example

Small

Middle

Large

−

+

Tab 1

Tab 2

Tab 3



demo ▾

Select date



Start date



End date



Button

Card

Name

Age

John Brown

32

Jim Green

42

Joe Black

32

< 1 >

Component size

Config component default size.

Content of Tab Pane 1

Direction

Components which support rtl direction are listed here, you can toggle the direction in the demo.

```
import {
  DownloadOutlined,
  LeftOutlined,
  MinusOutlined,
  PlusOutlined,
  RightOutlined,
  SearchOutlined as SearchIcon,
  SmileOutlined,
} from '@ant-design/icons';
import type { RadioChangeEvent } from 'antd';
import {
  Badge,
  Button,
  Cascader,
  Col,
  ConfigProvider,
  Divider,
  Input,
  InputNumber,
  Modal,
  Pagination,
  Radio,
  Rate,
  Row,
  Select,
  Space,
  Steps,
  Switch,
  Tree,
  TreeSelect,
}
```

Primary Color:

[illegible]

Border Radius: 6

Submit

Theme

Modify theme by `theme` prop.

```
import {
  Button,
  Card,
  ConfigProvider,
  DatePicker,
  Divider,
  Input,
  Radio,
  Select,
  Space,
  Table,
  Tabs,
} from 'antd';
import type { SizeType } from 'antd/es/config-provider/SizeContext';
import React, { useState } from 'react';

const { TabPane } = Tabs;

const App: React.FC = () => {
  const [componentSize, setComponentSize] = useState<SizeType>('small');

  return (
    <>
      <Radio.Group
        value={componentSize}
        onChange={(e) => {
          setComponentSize(e.target.value);
        }}
      >
        <Radio.Button value="small">Small</Radio.Button>
        <Radio.Button value="middle">Middle</Radio.Button>
        <Radio.Button value="large">Large</Radio.Button>
      </Radio.Group>
      <Divider />
      <ConfigProvider componentSize={componentSize}>
        <Space size={[0, 16]} style={{ width: '100%' }} direction="vertical">
          <Input />
          <Tabs defaultActiveKey="1">
            <TabPane tab="Tab 1" key="1">
              Content of Tab Pane 1
            </TabPane>
            <TabPane tab="Tab 2" key="2">
              Content of Tab Pane 2
            </TabPane>
          </Tabs>
        </Space>
      </ConfigProvider>
    </>
  );
};
```

```

        </TabPage>
        <TabPage tab="Tab 3" key="3">
            Content of Tab Pane 3
        </TabPage>
    </Tabs>
    <Input.Search allowClear />
    <Input.TextArea allowClear />
    <Select defaultValue="demo" options={[{ value: 'demo' }]} />
    <DatePicker />
    <DatePicker.RangePicker />
    <Button>Button</Button>
    <Card title="Card">
        <Table
            columns={[
                { title: 'Name', dataIndex: 'name' },
                { title: 'Age', dataIndex: 'age' },
            ]}
            dataSource={[
                { key: '1', name: 'John Brown', age: 32 },
                { key: '2', name: 'Jim Green', age: 42 },
                { key: '3', name: 'Joe Black', age: 32 },
            ]}
        />
    </Card>
</Space>
</ConfigProvider>
</>
);
};

export default App;

```

[illegible]

[illegible]


```

      <TreeNode title="parent 1-1" key="0-0-1">
        <TreeNode title={<span style={{ color: '#1890ff' }}>sss</span>} key="0-0-1-0" />
      </TreeNode>
    </TreeNode>
  </Tree>
</Col>
</Row>
<br />
<Row>
  <Col span={24}>
    <Divider orientation="left">Input (Input Group) example</Divider>
    <InputGroup size="large">
      <Row gutter={8}>
        <Col span={5}>
          <Input defaultValue="0571" />
        </Col>
        <Col span={8}>
          <Input defaultValue="26888888" />
        </Col>
      </Row>
    </InputGroup>
    <br />
    <InputGroup compact>
      <Input style={{ width: '20%' }} defaultValue="0571" />
      <Input style={{ width: '30%' }} defaultValue="26888888" />
    </InputGroup>
    <br />
    <InputGroup compact>
      <Select defaultValue="Option1">
        <Option value="Option1">Option1</Option>
        <Option value="Option2">Option2</Option>
      </Select>
      <Input style={{ width: '50%' }} defaultValue="input content" />
      <InputNumber />
    </InputGroup>
    <br />
    <Search placeholder="input search text" enterButton="Search" size="large" />
    <br />
    <br />
    <div style={{ marginBottom: 16 }}>
      <Input addonBefore={selectBefore} addonAfter={selectAfter} defaultValue="mysite" />
    </div>
    <br />
  </Row>
  <Col span={12}>
    <Divider orientation="left">Select example</Divider>
    <Space wrap>
      <Select mode="multiple" defaultValue="مورچه" style={{ width: 120 }}>
        <Option value="jack">Jack</Option>
        <Option value="مورچه">مورچه</Option>
        <Option value="disabled" disabled>
          Disabled
        </Option>
        <Option value="Yiminghe">yiminghe</Option>
      </Select>
      <Select defaultValue="مورچه" style={{ width: 120 }} disabled>
        <Option value="مورچه">مورچه</Option>
      </Select>
      <Select defaultValue="مورچه" style={{ width: 120 }} loading>
        <Option value="مورچه">مورچه</Option>
      </Select>
      <Select
        showSearch
        style={{ width: 200 }}
        placeholder="Select a person"
        optionFilterProp="children"
        filterOption={(input, option) =>
          option?.props.children.toLowerCase().includes(input.toLowerCase())
        }
      >
        <Option value="jack">Jack</Option>
        <Option value="سعيد">سعيد</Option>
        <Option value="tom">Tom</Option>
      </Select>
    </Col>
  </Row>
</Form>

```

```

        </Space>
    </Col>
    <Col span={12}>
        <Divider orientation="left">TreeSelect example</Divider>
        <TreeSelect
            showSearch
            style={{ width: '100%' }}
            dropdownStyle={{ maxHeight: 400, overflow: 'auto' }}
            placeholder="Please select"
            allowClear
            treeDefaultExpandAll
        >
            <TreeNode title="parent 1" key="0-1">
                <TreeNode title="parent 1-0" key="0-1-1">
                    <TreeNode title="my leaf" key="random" />
                    <TreeNode title="your leaf" key="random1" />
                </TreeNode>
                <TreeNode title="parent 1-1" key="random2">
                    <TreeNode title=<b style={{ color: '#08c' }}>sss</b> key="random3" />
                </TreeNode>
            </TreeNode>
        </TreeSelect>
    </Col>
</Row>
<br />
<Row>
    <Col span={24}>
        <Divider orientation="left">Modal example</Divider>
        <Button type="primary" onClick={showModal}>
            Open Modal
        </Button>
        <Modal title="پنجره ساده" open={modalOpen} onOk={handleOk} onCancel={handleCancel}:
            <p>نگاشته های خود را اینجا قرار دهید</p>
            <p>نگاشته های خود را اینجا قرار دهید</p>
            <p>نگاشته های خود را اینجا قرار دهید</p>
        </Modal>
    </Col>
</Row>
<br />
<Row>
    <Col span={24}>
        <Divider orientation="left">Steps example</Divider>
        <Steps
            progressDot
            current={currentStep}
            items={[
                {
                    title: 'Finished',
                    description: 'This is a description.',
                },
                {
                    title: 'In Progress',
                    description: 'This is a description.',
                },
                {
                    title: 'Waiting',
                    description: 'This is a description.',
                },
            ]}
        />
    </Col>
<br />
    <Steps
        current={currentStep}
        onChange={onStepsChange}
        items={[
            {
                title: 'Step 1',
                description: 'This is a description.',
            },
            {
                title: 'Step 2',
                description: 'This is a description.',
            },
        ]}
    </Steps>

```

```

        title: 'Step 3',
        description: 'This is a description.',
      },
    ]}
  />
</Col>
</Row>
<br />
<Row>
  <Col span={12}>
    <Divider orientation="left">Rate example</Divider>
    <Rate defaultValue={2.5} />
    <br />
    <strong>* Note:</strong> Half star not implemented in RTL direction, it will be
    supported after <a href="https://github.com/react-component/rate">rc-rate</a>{' '}
    implement rtl support.
  </Col>
  <Col span={12}>
    <Divider orientation="left">Badge example</Divider>
    <Badge count={badgeCount}>
      <a href="#" className="head-example" />
    </Badge>
    <ButtonGroup>
      <Button onClick={declineBadge}>
        <MinusOutlined />
      </Button>
      <Button onClick={increaseBadge}>
        <PlusOutlined />
      </Button>
    </ButtonGroup>
    <div style={{ marginTop: 12 }}>
      <Badge dot={showBadge}>
        <a href="#" className="head-example" />
      </Badge>
      <Switch onChange={onChangeBadge} checked={showBadge} />
    </div>
  </Col>
</Row>
</Col>
</Row>
<br />
<br />
<Row>
  <Col span={24}>
    <Divider orientation="left">Pagination example</Divider>
    <Pagination showSizeChanger defaultCurrent={3} total={500} />
  </Col>
</Row>
<br />
<Row>
  <Col span={24}>
    <Divider orientation="left">Grid System example</Divider>
    <div className="grid-demo">
      <div className="code-box-demo">
        <p>
          <strong>* Note:</strong> Every calculation in RTL grid system is from right side
          (offset, push, etc.)
        </p>
        <Row>
          <Col span={8}>col-8</Col>
          <Col span={8} offset={8}>
            col-8
          </Col>
        </Row>
        <Row>
          <Col span={6} offset={6}>
            col-6 col-offset-6
          </Col>
          <Col span={6} offset={6}>
            col-6 col-offset-6
          </Col>
        </Row>
        <Row>
          <Col span={12} offset={6}>

```

```

        col-12 col-offset-6
      </Col>
    </Row>
    <Row>
      <Col span={18} push={6}>
        col-18 col-push-6
      </Col>
      <Col span={6} pull={18}>
        col-6 col-pull-18
      </Col>
    </Row>
  </div>
</div>
</Col>
</Row>
</div>
);
};

const App: React.FC = () => {
  const [direction, setDirection] = useState<DirectionType>('ltr');
  const [popupPlacement, setPopupPlacement] = useState<Placement>('bottomLeft');

  const changeDirection = (e: RadioChangeEvent) => {
    const directionValue = e.target.value;
    setDirection(directionValue);
    if (directionValue === 'rtl') {
      setPopupPlacement('bottomRight');
    } else {
      setPopupPlacement('bottomLeft');
    }
  };

  return (
    <>
      <div style={{ marginBottom: 16 }}>
        <span style={{ marginRight: 16 }}>Change direction of components:</span>
        <Radio.Group defaultValue="ltr" onChange={changeDirection}>
          <Radio.Button key="ltr" value="ltr">
            LTR
          </Radio.Button>
          <Radio.Button key="rtl" value="rtl">
            RTL
          </Radio.Button>
        </Radio.Group>
      </div>
      <ConfigProvider direction={direction}>
        <Page popupPlacement={popupPlacement} />
      </ConfigProvider>
    </>
  );
};

export default App;

```


API

Property	Description	Type	Default
autoInsertSpaceInButton	Set false to remove space between 2 chinese characters on Button	boolean	true
componentDisabled	Config antd component <div>disabled</div>	boolean	-
componentSize	Config antd component size	<div>small</div> <div>middle</div> <div>large</div>	-
csp	Set Content Security Policy config	{ nonce: string }	-

Property	Description	Type	Default
direction	Set direction of layout. See demo	<code>ltr</code> <code>rtl</code>	<code>ltr</code>
dropdownMatchSelectWidth	Determine whether the dropdown menu and the select input are the same width. Default set <code>min-width</code> same as input. Will ignore when value less than select width. <code>false</code> will disable virtual scroll	<code>boolean</code> <code>number</code>	-
form	Set Form common props	<code>{ validateMessages?: ValidateMessages, requiredMark?: boolean <code>optional</code>, scrollToFirstError?: boolean Options }</code>	-
getPopupContainer	To set the container of the popup element. The default is to create a <code>div</code> element in <code>body</code>	<code>function(triggerNode)</code>	<code>() => document</code>
getTargetContainer	Config Affix, Anchor scroll target container	<code>() => HTMLElement</code>	<code>() => wir</code>
iconPrefixCls	Set icon prefix className	<code>string</code>	<code>anticon</code>
input	Set Input common props	<code>{ autoComplete?: string }</code>	-
select	Set Select common props	<code>{ showSearch?: boolean }</code>	-
locale	Language package setting, you can find the packages in antd/locale	<code>object</code>	-
prefixCls	Set prefix className	<code>string</code>	<code>ant</code>
renderEmpty	Set empty content of components. Ref Empty	<code>function(componentName: string): ReactNode</code>	-
space	Set Space <code>size</code> , ref Space	<code>{ size: <code>small</code> <code>middle</code> <code>large</code> <code>number</code> }</code>	-
theme	Set theme, ref Customize Theme	-	-
virtual	Disable virtual scroll when set to <code>false</code>	<code>boolean</code>	-

ConfigProvider.config() 4.13.0+

Setting Modal、Message、Notification rootPrefixCls.

```
ConfigProvider.config({
  prefixCls: 'ant', // 4.13.0+
  iconPrefixCls: 'anticon', // 4.17.0+
});
```

ConfigProvider.useConfig() 5.3.0+

Available since 5.2.0. Get the value of the parent Provider. Such as DisabledContextProvider, SizeContextProvider.

```
const {
  componentDisabled, // 5.3.0+
  componentSize, // 5.3.0+
} = ConfigProvider.useConfig();
```

返回值	说明	类型	默认值	版本
componentDisabled	antd component disabled state	boolean	-	5.3.0
componentSize	antd component size state	<div>small middle large</div>	-	5.3.0

FAQ

How to contribute a new language?

See <Adding new language>.

Date-related components locale is not working?

See FAQ <Date-related-components-locale-is-not-working?>

Modal throw error when setting getPopupContainer ?

Related issue: <https://github.com/ant-design/ant-design/issues/19974>

When you config getPopupContainer to parentNode globally, Modal will throw error of triggerNode is undefined because it did not have a triggerNode. You can try the fix below.

```
<ConfigProvider
-  getPopupContainer={triggerNode => triggerNode.parentNode}
+  getPopupContainer={node => {
+    if (node) {
+      return node.parentNode;
+    }
+    return document.body;
+  }}
>
<App />
```

</ConfigProvider>

Why can't ConfigProvider props (like `prefixCls` and `theme`) affect ReactNode inside `message.info` , `notification.open` , `Modal.confirm` ?

antd will dynamic create React instance by `ReactDOM.render` when call message methods. Whose context is different with origin code located context. We recommend `useMessage` , `useNotification` and `useModal` which , the methods came from `message/notification/Modal` has been deprecated in 5.x.

Locale is not working with Vite in production mode?

Related issue: [#39045](#)

In production mode of Vite, default exports from cjs file should be used like this: `enUS.default` . So you can directly import locale from `es/` directory like `import enUS from 'antd/es/locale/en_US'` to make dev and production have the same behavior.