**ANTD Notes**

[**https://ant.design/components/overview**](https://ant.design/components/overview)

a) Button , Input, Select

*import* './App.css';

*import* { Button, Input, Select } *from* 'antd';

*import* { useState } *from* 'react';

*import* { PoweroffOutlined, UserOutlined } *from* '@ant-design/icons'

function *App*() {

  const[*loading*, *setLoading*]= *useState*(*false*);

  const *onButtonClick* =(*e*)=>{

*setLoading*(*true*);

    // *console.log(e); e contain current object*

*setTimeout*(()=>{

*setLoading*(*false*)

},2000);

}

  const *fruits* =['appple','mango','banana','cherry']

*return* (

    <div *className*="App">

      <header *className*="App-header">

        {/\* *<Input.Search style={{ margin: '23px' }}></Input.Search>* \*/} //for search field

        <Input

*placeholder*='Enter Your Details'

*maxLength*={30} //used to define the maximum length

*type*='text'

*prefix*={<UserOutlined />}

          // *suffix={}*

*style*={{ width: "50%", margin: "20px" }}

        // *disabled={true}*

        ></Input>

        <Select

*placeholder*='Select Fruit'

*style*={{ width: '50%', margin: "20px" }}

*mode*='multiple'

*maxTagCount*={2} //*onlt two tags can be shown on screen rest as ...*

*allowClear*

        >

          {*fruits.map*((elm, index) => {

*return* <Select.Option *key*={index} *value*={elm}>{elm}</Select.Option>

          })}

        </Select>

        <Button

*loading*={loading}

*type*='primary'

          // *href='https://www.google.com' for adding a link to the button*

          // *block //for full width*

*onClick*={onButtonClick}

*icon*={<PoweroffOutlined />}

*style*={{ backgroundColor: 'red', fontSize: "23px", padding: '15px', textAlign: 'center', height: 'auto' }}

*className*='custom-class'

        >

          First ATND Button

        </Button>

      </header>

    </div>

  );

}

*export* *default* App;

Form -> alert , message , datapicker , timepicker, select

*import* React, { useState } *from* 'react'

*import* { Form, Input, Button, message, Alert, DatePicker, TimePicker, Select } *from* 'antd'

*import* './App.css'

const *AntForm* =()=>{

const[*showAlert*, *setShowAlert*]= *useState*(*false*);

const *gender* =['Male','Female','Transgender']

const *onFinish* =(*e*)=>{

        //*e will get the all the form data as an objects*

*console.log*(e);

*setTimeout*(()=>{

*setShowAlert*(*true*)

*message.success*('Login Successful')

            // *message.error('Login Successful')*

            // *message.warning('Login Successful')*

},1000);

}

*return* (

<div *className*='container'>

            <h1>Ant form</h1>

            {

                showAlert && <Alert *type*='error' *message*='error' *description*='Some Error Occured While Processing' *closable* />

            }

            <Form *onFinish*={onFinish}>

                <Form.Item *label*='Username' *name*='username'>

                    <Input *placeholder*='Enter your username' *allowClear* *required*></Input>

                </Form.Item>

                <Form.Item *label*='Password' *name*='password' >

                    <Input.Password

*placeholder*='Enter your password'

                        // *type='password'*

*required*

*allowClear*>

                    </Input.Password>

                </Form.Item>

                <Form.Item *label*='Confirm Password' *name*='cpassword' >

                    <Input.Password

*placeholder*='Confirm your password'

                        // *type='password'*

*required*

*allowClear*>

                    </Input.Password>

                </Form.Item>

                <Form.Item *label*='Date' *name*='date' >

                    {/\* *<DatePicker picker='year' />* \*/}

                    <DatePicker.RangePicker *picker*='month' />

                </Form.Item>

                <Form.Item *label*='Time' *name*='time' >

                    <TimePicker />

                </Form.Item>

                <Form.Item *label*="Select Gender">

                    <Select

*placeholder*='Select Fruit'

*style*={{ width: '50%', margin: "20px" }}

*mode*='multiple'

*maxTagCount*={2}

*allowClear*

                    >

                        {*gender.map*((elm, index) => {

*return* <Select.Option *key*={index} *value*={elm}>{elm}</Select.Option>

                        })}

                    </Select>

                </Form.Item>

                <Form.Item >

                    <Button *type*='primary' *htmlType*='submit' *block*>Login</Button>

                </Form.Item>

                <Form.Item>

                    <Input *type*='checkbox'></Input>

                    <p>Agree to our Terms and Condition</p>

                </Form.Item>

            </Form>

        </div>

)

}

*export* *default* AntForm

Form Validations

*import* "./App.css";

*import* { Form, Button, Checkbox, DatePicker, Input, Select, Space } *from* "antd";

function *App*() {

*return* (

        <div *className*="App">

            <header *className*="App-header">

                <Form

*autoComplete*="off"

*labelCol*={{ span: 10 }}

*wrapperCol*={{ span: 14 }}

*onFinish*={(values) => {

*console.log*({ values });

                    }}

*onFinishFailed*={(error) => {

*console.log*({ error });

                    }}

                >

                    <Form.Item

*name*="fullName"

*label*="Full Name"

*rules*={[

                            {

                                required: true,

                                message: "Please enter your name",

                            },

                            { whitespace: true },

                            { min: 3 },

                        ]}

*hasFeedback*

                    >

                        <Input *placeholder*="Type your name" />

                    </Form.Item>

                    <Form.Item

*name*="email"

*label*="Email"

*rules*={[

                            {

                                required: true,

                                message: "Please enter your email",

                            },

                            { type: "email", message: "Please enter a valid email" },

                        ]}

*hasFeedback*

                    >

                        <Input *placeholder*="Type your email" />

                    </Form.Item>

                    <Form.Item

*name*="password"

*label*="Password"

*rules*={[

                            {

                                required: true,

                            },

                            { min: 6 },

                            {

*validator*: (\_, value) =>

                                    value && *value.includes*("A")

                                        ? Promise*.resolve*()

                                        : Promise*.reject*("Password does not match criteria."),

                            },

                        ]}

*hasFeedback*

                    >

                        <Input.Password *placeholder*="Type your password" />

                    </Form.Item>

                    <Form.Item

*name*="confirmPassword"

*label*="Confirm Password"

*dependencies*={["password"]}

*rules*={[

                            {

                                required: true,

                            },

                            ({ getFieldValue }) => ({

*validator*(\_, value) {

                                    if (!value || *getFieldValue*("password") === value) {

*return* Promise*.resolve*();

                                    }

*return* Promise*.reject*(

                                        "The two passwords that you entered does not match."

                                    );

                                },

                            }),

                        ]}

*hasFeedback*

                    >

                        <Input.Password *placeholder*="Confirm your password" />

                    </Form.Item>

                    <Form.Item *name*="gender" *label*="Gender" *requiredMark*="optional">

                        <Select *placeholder*="Select your gender">

                            <Select.Option *value*="male">Male</Select.Option>

                            <Select.Option *value*="female">Female</Select.Option>

                        </Select>

                    </Form.Item>

                    <Form.Item

*name*="dob"

*label*="Date of Birth"

*rules*={[

                            {

                                required: true,

                                message: "Please provide your date of birth",

                            },

                        ]}

*hasFeedback*

                    >

                        <DatePicker

*style*={{ width: "100%" }}

*picker*="date"

*placeholder*="Chose date of birth"

                        />

                    </Form.Item>

                    <Form.Item

*name*="website"

*label*="Website"

*rules*={[{ type: "url", message: "Please enter a valid url" }]}

*hasFeedback*

                    >

                        <Input *placeholder*="Add your website url" />

                    </Form.Item>

                    <Form.Item

*name*="agreement"

*wrapperCol*={{ span: 24 }}

*valuePropName*="checked"

*rules*={[

                            {

*validator*: (\_, value) =>

                                    value

                                        ? Promise*.resolve*()

                                        : Promise*.reject*(

                                            "To proceed, you need to agree with our terms and conditions"

                                        ),

                            },

                        ]}

                    >

                        <Checkbox>

                            {" "}

                            Agree to our <a *href*="#">Terms and Conditions</a>

                        </Checkbox>

                    </Form.Item>

                    <Form.Item *wrapperCol*={{ span: 24 }}>

                        <Button *block* *type*="primary" *htmlType*="submit">

                            Register

                        </Button>

                    </Form.Item>

                </Form>

            </header>

        </div>

    );

}

*export* *default* App;

Basic Table

const *AntTable* =()=>{

const *data* =[

{

*name*:"Name 1",

*age*:10,

*address*:"Address 1",

*key*:'1'

},

{

*name*:"Name 2",

*age*:20,

*address*:"Address 2",

*key*:'2'

},

{

*name*:"Name 3",

*age*:30,

*address*:"Address 3",

*key*:'3'

},

{

*name*:"Name 4",

*age*:40,

*address*:"Address 4",

*key*:'4'

},

]

const *columns* =[

{

*title*:"Name",

*dataIndex*:'name', // it will map data field value accordingly

*key*:'key',

            // *render: name => {*

            // *return <a onClick={}>{name}</a>*

            // *}*

},

{

*title*:"Age",

*dataIndex*:'age',

*key*:'key',

*sorter*:(*a*, *b*)=> *a.age* - *b.age*

},

{

*title*:"Address",

*dataIndex*:'address',

*key*:'key'

},

{

*title*:"Graduated",

            // *dataIndex: 'graduated',*

*key*:'key',

*render*:(*payload*)=>{

*return* <p>{*payload.age* >20?'True':'False'}</p>

} // payload contain all the rows value as an object

},

        // *{*

        // *title: "Address",*

        // *dataIndex: 'address',*

        // *key: 'key'*

        // *}*

]

*return* (

<div *className*='container'>

            <Table

*dataSource*={data} //this field contain the all the data

*columns*={columns} //columns define the heading o

            // *pagination='1'*

            ></Table>

        </div>

)

}

Ant Spinner

*import* { Spin, Button } *from* 'antd'

*import* React, { useState } *from* 'react'

//*\* most of the components in ant design has spinner but some component not there we can use this spin component*

const *AntSpinner* =()=>{

const[*loading*, *setLoading*]= *useState*(*false*);

*return* (

<div *className*='container'>

      <h1>Spinner</h1>

      <Spin *spinning*={loading} >

        <p>My name is anthony</p>

      </Spin>

      <Button

*type*='primary'

*style*={{ margin: '20px' }}

*onClick*={() => {

*setLoading*(preValue => !preValue)

        }}

      >Toggle Spinning</Button>

    </div>

)

}

*export* *default* AntSpinner

Ant Progress Component

*import* { Progress } *from* 'antd'

*import* React *from* 'react'

const *AntProgress* =()=>{

*return* (

<div *className*='container'>

      <Progress *percent*={33} *status*='active' />

      <Progress *percent*={33} *status*='success' *steps*={3} *type*='line' />

      <Progress *percent*={33} *type*='circle' *strokeColor*='red' *strokeWidth*={20} />

      {/\* *some other status options are active circle success exception* \*/}

    </div>

)

}

*export* *default* AntProgress

Ant TablePagination , filters

*import* React, { useEffect, useState } *from* 'react'

*import* { Table } *from* 'antd'

const *AntTable* =()=>{

const[*dataSource*, *setDataSource*]= *useState*([]);

const[*loading*, *setLoading*]= *useState*(*false*);

const[*page*, *setPage*]= *useState*(1)

const[*pageSize*, *setPagesize*]= *useState*(5)

const *columns* =[

        //*sample key getting from api*

        // *{*

        // *"userId": 1,*

        // *"id": 1,*

        // *"title": "delectus aut autem",*

        // *"completed": false*

        // *},*

{

*key*:'1',

*title*:'ID',

*dataIndex*:'id',

},

{

*key*:'2',

*title*:'User ID',

*dataIndex*:'userId',

*sorter*:(*record1*, *record2*)=>{

*return record1.userId* - *record2.userId*;

} // sorter function used to sort the field value in ascending or descending order as required

},

{

*key*:'3',

*title*:'Status',

*dataIndex*:'completed',

*render*:(*completed*)=>{

*return* <p>{completed?'Completed':"In Progress"}</p>

},

*filters*:[

{ *text*:'Complete', *value*: *true* },

{ *text*:'In Progress', *value*: *false* }

],

*onFilter*:(*value*, *record*)=>{

*return record.completed* ===value

}

},

]

*useEffect*(()=>{

*setLoading*(*true*)

*fetch*('https://jsonplaceholder.typicode.com/todos')*.then*(*response* => *response.json*())*.then*(*data* =>{

*setDataSource*(data)

})*.catch*(*err* => *console.log*(err))*.finally*(()=>{

*setLoading*(*false*);

})

},[])

*return* <>

<h1>*Ant Design Table*</h1>

<Table

*loading*={loading}

*dataSource*={dataSource}

*columns*={columns}

*pagination*={{

                current: page, //*by default page starts from 3rd page*

                pageSize: pageSize,

                // *total: 500,*

*onChange*: (page, pageSize) => {

*setPage*(page);

*setPagesize*(pageSize)

                }

            }

            }

        ></Table>

</>

}

*export* *default* AntTable

Ant Select Table

*import* React, { useState } *from* 'react'

*import* { Table, Tag } *from* 'antd'

const *AntSelectTable* =()=>{

const[*alreadySelectedRows*, *setAlreadySelectedRows*]= *useState*(['1','3'])

const *columns* =[

{

*title*:"Student Id",

*dataIndex*:'id',

},

{

*title*:"Student Name",

*dataIndex*:'name',

},

{

*title*:"Student Grade",

*dataIndex*:'grade',

*render*:(*tag*)=>{

const *color* = *tag.includes*('A')?'Green': *tag.includes*('B')?"purple": *tag.includes*('C')?

"red":"volcano";

*return* <Tag *color*={color} *key*={tag}>{tag}</Tag>

}

},

]

const *dataSource* =[

{

*key*:'1',

*id*:'1',

*name*:'Student Name 1',

*grade*:'A'

},

{

*key*:'2',

*id*:'2',

*name*:'Student Name 2',

*grade*:'B+'

},

{

*key*:'3',

*id*:'3',

*name*:'Student Name 3',

*grade*:'C'

},

{

*key*:'4',

*id*:'4',

*name*:'Student Name 4',

*grade*:'D'

}

]

*return* (

<div>

<h1>*Select Fields*</h1>

<Table

*style*={{ width: '50%' }}

*columns*={columns}

*dataSource*={dataSource}

*rowSelection*={{

                    type: 'checkbox',

                    selectedRowKeys: alreadySelectedRows,

*onChange*: (keys) => {

*setAlreadySelectedRows*(keys)

                    },

*onSelect*: (record) => {

*console.log*(record);

                    },

                    // *hideSelectAll: true,*

                    selections: [

*Table.*SELECTION\_NONE,

*Table.*SELECTION\_ALL,

*Table.*SELECTION\_INVERT,

                        {

                            key: 'even',

                            text: 'Select Even Rows',

*onSelect*: (allKeys) => {

                                const *selectedKeys* = *allKeys.filter*(*key* =>{

*return* key%2==0

})

*setAlreadySelectedRows*(selectedKeys)

                            }

                        },

                        {

                            key: 'excellent',

                            text: 'Select Student with excellent grades',

*onSelect*: (allKeys) => {

                                const *selectedKeys* = *allKeys.filter*(*key* =>{

const *isExcellent* = *dataSource.find*(*student* =>{

*return student.key* ==key&& *student.grade.includes*('A');

})

*return* isExcellent;

})

*setAlreadySelectedRows*(selectedKeys)

                            }

                        }

                    ]

                    // *getCheckboxProps: (record) => ({*

                    // *disabled: record.grade === 'C'*

                    // *})*

                    // *getCheckboxProps: (record) => {*

                    // *if (alreadySelectedRows.includes(record.id)) {*

                    // *return ({ disabled: true })*

                    // *} else {*

                    // *return {}*

                    // *}*

                    // *}*

                }}

            ></Table>

</div>

)

}

*export* *default* AntSelectTable

Ant ICONS

*import* { LoadingOutlined, PieChartFilled } *from* '@ant-design/icons'

*import* { Divider } *from* 'antd'

*import* React *from* 'react'

const *AntIcons* =()=>{

    // *we can also add custom icons watch the tutorial from code with aamir*

*return* (

<div>

<h1>*Ant ICONS*</h1>

<PieChartFilled *style*={{

                color: 'purple',

                fontSize: 100

            }} />

<LoadingOutlined

*twoToneColor*='green' *spin*

*rotate*={45} *style*={{

                    color: 'red',

                    fontSize: 100

                }} />

<Divider />

</div>

)

}

*export* *default* AntIcons

Ant CRUD Table

// *import "antd/dist/antd.css";*

*import* "./App.css";

*import* { Button, Table, Modal, Input } *from* "antd";

*import* { useState } *from* "react";

*import* { EditOutlined, DeleteOutlined } *from* "@ant-design/icons";

function *App*() {

    const[*isEditing*, *setIsEditing*]= *useState*(*false*);

    const[*editingStudent*, *setEditingStudent*]= *useState*(*null*);

    const[*dataSource*, *setDataSource*]= *useState*([

{

*id*:1,

*name*:"John",

*email*:"john@gmail.com",

*address*:"John Address",

},

{

*id*:2,

*name*:"David",

*email*:"david@gmail.com",

*address*:"David Address",

},

{

*id*:3,

*name*:"James",

*email*:"james@gmail.com",

*address*:"James Address",

},

{

*id*:4,

*name*:"Sam",

*email*:"sam@gmail.com",

*address*:"Sam Address",

},

]);

    const *columns* =[

{

*key*:"1",

*title*:"ID",

*dataIndex*:"id",

},

{

*key*:"2",

*title*:"Name",

*dataIndex*:"name",

},

{

*key*:"3",

*title*:"Email",

*dataIndex*:"email",

},

{

*key*:"4",

*title*:"Address",

*dataIndex*:"address",

},

{

*key*:"5",

*title*:"Actions",

*render*:(*record*)=>{

*return* (

<>

<EditOutlined

*onClick*={() => {

*onEditStudent*(record);

                            }}

                        />

<DeleteOutlined

*onClick*={() => {

*onDeleteStudent*(record);

                            }}

*style*={{ color: "red", marginLeft: 12 }}

                        />

</>

);

},

},

];

    const *onAddStudent* =()=>{

const *randomNumber* = *parseInt*(*Math.random*()\*1000);

const *newStudent* ={

*id*:randomNumber,

*name*:"Name "+randomNumber,

*email*:randomNumber+"@gmail.com",

*address*:"Address "+randomNumber,

};

*setDataSource*((*pre*)=>{

*return* [...pre,newStudent];

});

};

    const *onDeleteStudent* =(*record*)=>{

*Modal.confirm*({

*title*:"Are you sure, you want to delete this student record?",

*okText*:"Yes",

*okType*:"danger",

*onOk*:()=>{

*setDataSource*((*pre*)=>{

*return pre.filter*((*student*)=> *student.id* !== *record.id*);

});

},

});

};

    const *onEditStudent* =(*record*)=>{

*setIsEditing*(*true*);

*setEditingStudent*({...record});

};

    const *resetEditing* =()=>{

*setIsEditing*(*false*);

*setEditingStudent*(*null*);

};

*return* (

        <div *className*="App">

            <header *className*="App-header">

                <Button *onClick*={onAddStudent}>Add a new Student</Button>

                <Table *columns*={columns} *dataSource*={dataSource}></Table>

                <Modal

*title*="Edit Student"

*visible*={isEditing}

*okText*="Save"

*onCancel*={() => {

*resetEditing*();

                    }}

*onOk*={() => {

*setDataSource*((pre) => {

*return* *pre.map*((student) => {

                                if (*student.*id === *editingStudent.*id) {

*return* editingStudent;

                                } else {

*return* student;

                                }

                            });

                        });

*resetEditing*();

                    }}

                >

                    <Input

*value*={*editingStudent?.*name}

*onChange*={(e) => {

*setEditingStudent*((pre) => {

*return* { ...pre, name: *e.target.*value };

                            });

                        }}

                    />

                    <Input

*value*={*editingStudent?.*email}

*onChange*={(e) => {

*setEditingStudent*((pre) => {

*return* { ...pre, email: *e.target.*value };

                            });

                        }}

                    />

                    <Input

*value*={*editingStudent?.*address}

*onChange*={(e) => {

*setEditingStudent*((pre) => {

*return* { ...pre, address: *e.target.*value };

                            });

                        }}

                    />

                </Modal>

            </header>

        </div>

    );

}

*export* *default* App;

Ant Table Search Columns

// *import "antd/dist/antd.css";*

*import* "./App.css";

*import* { Button, Input, Table } *from* "antd";

*import* { useState } *from* "react";

*import* { SearchOutlined } *from* "@ant-design/icons";

function *AntTableSearch*() {

    const[*dataSource*, *setDataSource*]= *useState*([

{

*name*:"John",

*age*:32,

*address*:"New York",

},

{

*name*:"Jim",

*age*:33,

*address*:"Sydney",

},

{

*name*:"David",

*age*:40,

*address*:"Japan",

},

{

*name*:"James",

*age*:32,

*address*:"New York",

},

{

*name*:"Sam",

*age*:40,

*address*:"Sydney",

},

{

*name*:"John",

*age*:32,

*address*:"New York",

},

{

*name*:"Jim",

*age*:33,

*address*:"Sydney",

},

{

*name*:"David",

*age*:40,

*address*:"Japan",

},

{

*name*:"James",

*age*:32,

*address*:"New York",

},

{

*name*:"Sam",

*age*:40,

*address*:"Sydney",

},

]);

    const *columns* =[

{

*title*:"Name",

*dataIndex*:"name",

*filterDropdown*:({

setSelectedKeys,

selectedKeys,

confirm,

clearFilters,

})=>{

*return* (

<>

<Input

*autoFocus*

*placeholder*="Type text here"

*value*={selectedKeys[0]}

*onChange*={(e) => {

*setSelectedKeys*(*e.target.*value ? [*e.target.*value] : []);

*confirm*({ closeDropdown: false });

                            }}

*onPressEnter*={() => {

*confirm*();

                            }}

*onBlur*={() => {

*confirm*();

                            }}

                        ></Input>

<Button

*onClick*={() => {

*confirm*();

                            }}

*type*="primary"

                        >

                            Search

                        </Button>

<Button

*onClick*={() => {

*clearFilters*();

                            }}

*type*="danger"

                        >

                            Reset

                        </Button>

</>

);

},

*filterIcon*:()=>{

*return* <SearchOutlined />;

},

*onFilter*:(*value*, *record*)=>{

*return record.name.toLowerCase*()*.includes*(*value.toLowerCase*());

},

},

{

*title*:"Age",

*dataIndex*:"age",

*filterDropdown*:({

setSelectedKeys,

selectedKeys,

confirm,

clearFilters,

})=>{

*return* (

<>

<Input

*autoFocus*

*placeholder*="Type text here"

*value*={selectedKeys[0]}

*onChange*={(e) => {

*setSelectedKeys*(*e.target.*value ? [*e.target.*value] : []);

*confirm*({ closeDropdown: false });

                            }}

*onPressEnter*={() => {

*confirm*();

                            }}

*onBlur*={() => {

*confirm*();

                            }}

                        ></Input>

<Button

*onClick*={() => {

*confirm*();

                            }}

*type*="primary"

                        >

                            Search

                        </Button>

<Button

*onClick*={() => {

*clearFilters*();

                            }}

*type*="danger"

                        >

                            Reset

                        </Button>

</>

);

},

*filterIcon*:()=>{

*return* <SearchOutlined />;

},

*onFilter*:(*value*, *record*)=>{

*return record.age* ==value;

},

},

{

*title*:"Address",

*dataIndex*:"address",

*filterDropdown*:({

setSelectedKeys,

selectedKeys,

confirm,

clearFilters,

})=>{

*return* (

<div *style*={{ display: "flex", flex: 1, justifyContent: "center" }}>

                        <Input

*autoFocus*

*placeholder*="Type text here"

*value*={selectedKeys[0]}

*onChange*={(e) => {

*setSelectedKeys*(*e.target.*value ? [*e.target.*value] : []);

*confirm*({ closeDropdown: false });

                            }}

*onPressEnter*={() => {

*confirm*();

                            }}

*onBlur*={() => {

*confirm*();

                            }}

                        ></Input>

                        <Button

*onClick*={() => {

*confirm*();

                            }}

*type*="primary"

                        >

                            Search

                        </Button>

                        <Button

*onClick*={() => {

*clearFilters*();

                            }}

*type*="danger"

                        >

                            Reset

                        </Button>

                    </div>

);

},

*filterIcon*:()=>{

*return* <SearchOutlined />;

},

*onFilter*:(*value*, *record*)=>{

*return record.address.toLowerCase*()*.includes*(*value.toLowerCase*());

},

},

];

*return* (

        <div *className*="App">

            <header *className*="App-header">

                <Table

*style*={{ display: "flex", flex: 1, margin: 10 }}

*columns*={columns}

*dataSource*={dataSource}

                ></Table>

            </header>

        </div>

    );

}

*export* *default* AntTableSearch;

Ant Editable Table

*import* { Button, Table, Form, Input } *from* "antd";

*import* "antd/dist/antd.css";

*import* { useEffect } *from* "react";

*import* { useState } *from* "react/cjs/react.development";

*import* "./App.css";

function *App*() {

    const[*dataSource*, *setDataSource*]= *useState*([]);

    const[*editingRow*, *setEditingRow*]= *useState*(*null*);

    const[*form*]= *Form.useForm*();

*useEffect*(() => {

        const *data* =[];

*for* (letindex=0; index < 7; index++) {

*data.push*({

                key: `${index}`,

                name: `Name ${index}`,

                address: `Address ${index}`,

            });

        }

*setDataSource*(data);

    }, []);

    const *columns* =[

{

*title*:"Name",

*dataIndex*:"name",

*render*:(*text*, *record*)=>{

if(editingRow=== *record.key*){

*return* (

<Form.Item

*name*="name"

*rules*={[

                                {

                                    required: true,

                                    message: "Please enter your name",

                                },

                            ]}

                        >

                            <Input />

                        </Form.Item>

);

}else{

*return* <p>{text}</p>;

}

},

},

{

*title*:"Address",

*dataIndex*:"address",

*render*:(*text*, *record*)=>{

if(editingRow=== *record.key*){

*return* (

<Form.Item *name*="address">

                            <Input />

                        </Form.Item>

);

}else{

*return* <p>{text}</p>;

}

},

},

{

*title*:"Actions",

*render*:(*\_*, *record*)=>{

*return* (

<>

<Button

*type*="link"

*onClick*={() => {

*setEditingRow*(*record.*key);

*form.setFieldsValue*({

                                    name: *record.*name,

                                    address: *record.*address,

                                });

                            }}

                        >

                            Edit

                        </Button>

<Button *type*="link" *htmlType*="submit">

                            Save

                        </Button>

</>

);

},

},

];

    const *onFinish* =(*values*)=>{

const *updatedDataSource* =[...dataSource];

*updatedDataSource.splice*(editingRow,1,{...values, *key*:editingRow});

*setDataSource*(updatedDataSource);

*setEditingRow*(*null*);

};

*return* (

        <div *className*="App">

            <header *className*="App-header">

                <Form *form*={form} *onFinish*={onFinish}>

                    <Table *columns*={columns} *dataSource*={dataSource}></Table>

                </Form>

            </header>

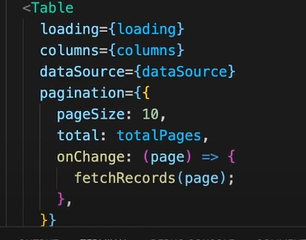
        </div>

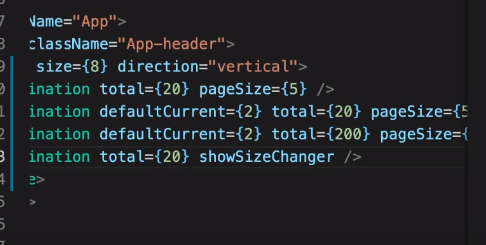
    );

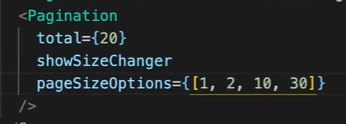
}

*export* *default* App;

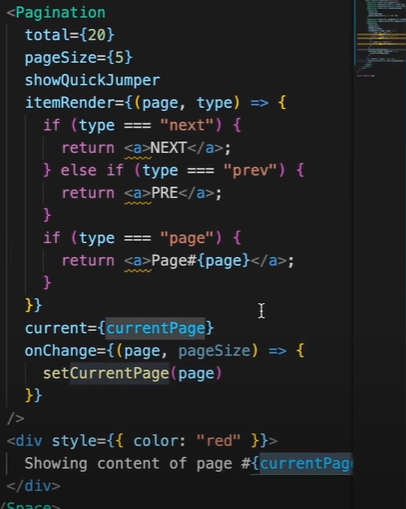
Ant Table Pagination







Goto page : 



Ant File Upload

*import* { Button, Spin, Upload } *from* 'antd'

*import* React *from* 'react'

const *AntFileUpload* =()=>{

*return* (

<div *style*={{

            display: 'flex',

            flexDirection: 'column',

            justifyContent: 'center',

            alignItems: 'center',

            height: '100vh'

        }}>

            {/\* *<Upload action='<URLWEHRETOSAVEFILE>'>* \*/}

            <Upload.Dragger

*multiple*

*listType*='picture'

*showUploadList*={{ showRemoveIcon: false }}

                // *accept='.png,.jpeg,.doc'*

*beforeUpload*={file => {

*console.log*(*file.*size)

                }}

                // *fileList={}*

*defaultFileList*={[

                    {

                        uid: 'abc',

                        name: 'file.png',

                        url: 'https://www.google.com',

                        status: "uploading",

                        percent: 50

                    }

                ]}

*iconRender*={() => {

*return* <Spin></Spin>

                }}

                // *itemRender={(existingCom, file) => {*

                // *return <p>{file.name}</p>*

                // *}}*

*progress*={{

                    strokeWidth: 3,

                    strokeColor: 'red'

                }}

            >

                Drag Files here or <br />

                <Button *style*={{ width: '100%', margin: '12px 0' }}>Upload</Button>

            </Upload.Dragger>

        </div>

)

}

*export* *default* AntFileUpload

Ant Tags

*import* { Space, Tag } *from* 'antd'

*import* React *from* 'react'

const *AntTags* =()=>{

*return* (

<div>

<h2>*Ant TAGS*</h2>

<Space *direction*='vertical' *size*={2}>

                <Tag>Tag</Tag>

                <Tag>Tag2</Tag>

                <Tag

*color*='warning'

*closable*

*onClose*={() => {

*console.log*('Tag Closed')

                    }}

                > closable </Tag>

            </Space>

</div>

)

}

*export* *default* AntTags

Ant Typography

*import* React, { useState } *from* 'react'

*import* { Typography } *from* 'antd'

*import* { CheckCircleFilled, CopyFilled, SearchOutlined } *from* '@ant-design/icons';

const{ *Title*, *Text*, *Paragraph*, *Link* }=Typography;

const *AntTypogarphy* =()=>{

const[*text*, *setText*]= *useState*('code with aamir');

*return* (

<div>

{/\* *by default h1 to change use level*\*/}

<Title>*Code with Anupam*</Title>

<Title *level*={5}>Code with Anupam</Title>

<Text>*Learn Ant Design*</Text>

<Text *strong*>Learn Ant Design</Text>

<Text *underline*>Learn Ant Design</Text>

<Text *mark*>Learn Ant Design</Text>

<Text *disabled*>Learn Ant Design</Text>

<Text *type*='danger'>Learn Ant Design</Text>

<Link *underline* *href*='https://google.com'>Code with aamir</Link>

<Paragraph

*strong*

*editable*={{

*onChange*: (value) => {

*setText*(value);

                    },

                    triggerType: "icon/text",

                    tooltip: 'click on the icon to edit',

                    icon: < SearchOutlined />,

                    enterIcon: <CheckCircleFilled />

                }}

*copyable*={{

                    tooltips: 'tool to copy',

                    text: 'copy code',

                    icon: <CopyFilled />

                }}

                // *...*

*ellipsis*={{

                    rows: 2,

                    expandable: true,

                    symbol: 'Show more'

                }}

            >

                {text}

            </Paragraph>

</div>

)

}

*export* *default* AntTypogarphy

Ant Divider

*import* React *from* 'react'

*import* { Divider } *from* 'antd'

const *AntDivider* =()=>{

*return* (

<div>

<Divider *style*={{ borderColor: 'red' }} *dashed* />

<Divider *style*={{ borderColor: 'red' }}

*orientation*='left'

*orientationMargin*={23}

*type*='vertical'

            >TExt</Divider>

</div>

)

}

*export* *default* AntDivider

Ant MENU

*import* React *from* 'react'

*import* { Input, Menu, Space } *from* 'antd'

*import* { AccountBookFilled, ContactsOutlined, HomeOutlined } *from* '@ant-design/icons'

const *AntMenu* =()=>{

*return* (

<div>

<h1>*Menu*</h1>

<Space>

<Menu

                    // *mode='inline'*

                    // *defaultOpenKeys={['about-us']}*

*onClick*={(info) => {

*console.log*(*info.*key)

                    }}

*items*={

                        [

                            {

                                label: <Input.Search *placeholder*='Search Here...'></Input.Search>,

                                key: 'search',

                            },

                            {

                                label: "Home",

                                key: 'home',

                                icon: <HomeOutlined />

                            },

                            {

                                label: "About Us",

                                key: 'about-us',

                                icon: <AccountBookFilled />,

                                children: [

                                    {

                                        label: 'Our Storya',

                                        key: 'our-storya'

                                    },

                                    {

                                        label: 'Meet the Team',

                                        key: 'meet-the-teama'

                                    }

                                ]

                            },

                            {

                                label: <span *style*={{ color: 'red' }}>Red</span>,

                                key: '-s',

                                type: 'group',

                                icon: <AccountBookFilled />,

                                children: [{

                                    label: 'Our Story',

                                    key: 'our-storyaa'

                                }, {

                                    label: 'Meet the Team',

                                    key: 'meet-the-teams'

                                }]

                            },

                            {

                                label: "Uss",

                                key: '-ss',

                                type: 'divider',

                                icon: <AccountBookFilled />,

                                children: [{

                                    label: 'Our Storys',

                                    key: 'our-story'

                                }, {

                                    label: 'Meet the Teasm',

                                    key: 'meet-the-teamas'

                                }]

                            },

                            {

                                label: "Contact Us",

                                key: 'contact-us',

                                icon: <ContactsOutlined />,

                                danger: true

                            }

                        ]}

                ></Menu>

</Space>

</div>

)

}

*export* *default* AntMenu

Ant TAB

*import* React, { useState } *from* 'react'

*import* { Space, Tabs } *from* 'antd'

*import* { ApiOutlined } *from* '@ant-design/icons'

const *AntTab* =()=>{

const[*tabList*, *setTabList*]= *useState*([

{

*tab*:'tab1', *key*:'1'

},

{

*tab*:'tab2', *key*:'2'

},

]);

*return* (

<div>

{/\* *tabs on top* \*/}

<Space *size*={30} *direction*='vertical'>

                <Tabs *defaultActiveKey*='1'>

                    <Tabs.TabPane *tab*="Tab 1" *key*="1">Content of Tab Pane 1</Tabs.TabPane>

                    <Tabs.TabPane *tab*="Tab 2" *key*="2">Content ohttps://www.youtube.com/f Tab Pane 2</Tabs.TabPane>

                    <Tabs.TabPane *tab*="Tab 3" *key*="3">Content of Tab Pane 3</Tabs.TabPane>

                </Tabs>

                {/\* *tabs on left* \*/}

                <Tabs *defaultActiveKey*='1' *tabPosition*='left'>

                    <Tabs.TabPane *tab*="Tab 1" *key*="1">Content of Tab Pane 1</Tabs.TabPane>

                    <Tabs.TabPane *tab*="Tab 2" *key*="2">Content of Tab Pane 2</Tabs.TabPane>

                    <Tabs.TabPane *tab*={

                        <span><ApiOutlined />Icon</span>

                    } *key*="3">Content of Tab Pane 3</Tabs.TabPane>

                </Tabs>

                {/\* *tabs with add or delete* \*/}

                <Tabs *defaultActiveKey*='1' *tabPosition*='left'>

                    <Tabs.TabPane *tab*="Tab 1" *key*="1">Content of Tab Pane 1</Tabs.TabPane>

                    <Tabs.TabPane *tab*="Tab 2" *key*="2">Content of Tab Pane 2</Tabs.TabPane>

                    <Tabs.TabPane *tab*={

                        <span><ApiOutlined />Icon</span>

                    } *key*="3">Content of Tab Pane 3</Tabs.TabPane>

                </Tabs>

                {/\* *tabs with add or delete* \*/}

                <Tabs *defaultActiveKey*='1' *tabPosition*='left'>

                    {

*tabList.map*((tabInfo, index) => {

*return* <Tabs.TabPane *tab*={`Tab ${*tabInfo.*tab}`} *key*="1">Content of Tab Pane 1adasdfadsfsf</Tabs.TabPane>

                        })

                    }

                </Tabs>

            </Space>

</div>

)

}

*export* *default* AntTab

ANT AVATAR

*import* { Avatar } *from* 'antd'

*import* React *from* 'react'

*import* { UserOutlined, } *from* '@ant-design/icons'

const *AntAvatar* =()=>{

*return* (

<div>

<h1>*Ant Design*</h1>

<p><b>*Avatars:* </b></p>

<Avatar *src*="https://zos.alipayobjects.com/rmsportal/ODTLcjx" />

<br /><br />

<p><b>*Letter Avatars:* </b></p>

<Avatar *style*={{ backgroundColor: '#f56a00', color: '#fff' }} *icon*={<UserOutlined />} shape=’square’ />

{/\* *<Avatar style={{ backgroundColor: '#87d068', color: '#fff' }} icon={<VideoCameraOut />} />* \*/}

<Avatar.Group></Avatar.Group>

</div>

)

}

*export* *default* AntAvatar

Ant Caraousel

*import* React, { useRef } *from* 'react'

*import* { Carousel, Button } *from* 'antd'

const *AntCaraousel* =()=>{

const *ref* = *useRef*();

const *contentStyle* ={

*height*:'250px',

*color*:'#fff',

*lineHeight*:'160px',

*textAlign*:'center',

*background*:'#364d79',

};

const *MyCarousel* =()=>(

<Carousel *dotPosition*="bottom"

*autoplay*

*dots*={{

                className: 'my-dots',

            }}

*pauseOnHover*={true}

*pauseOnDotsHover*={true}

        >

            <div>

                <h3 *style*={contentStyle}>1</h3>

            </div>

            <div>

                <h3 *style*={contentStyle}>2</h3>

            </div>

            <div>

                <h3 *style*={contentStyle}>3</h3>

            </div>

            <div>

                <h3 *style*={contentStyle}>4</h3>

            </div>

        </Carousel >

);

*return* (

<div>

<h1>*Caraousel*</h1>

<div>

<Carousel

*autoplay*

*dots*={true}

                    // *dotPosition='bottom'*

                    // *pauseOnHover={true}*

                    // *pauseOnDotsHover={true}*

                    // *draggable*

*ref*={ref}

*effect*='fade'

                >

                    <div><h3 *style*={contentStyle}>1</h3></div>

                    <div><h3 *style*={contentStyle}>2</h3></div>

                    <div><h3 *style*={contentStyle}>3</h3></div>

                    <div><h3 *style*={contentStyle}>4</h3></div>

                </Carousel>

<div>

<Button

*onClick*={() => {

*ref.current.prev*();

                        }}

                    >Prev</Button>

<Button *onClick*={() => {

*ref.current.goTo*(0);

                    }}>Reset</Button>

<Button *onClick*={() => {

*ref.current.next*();

                    }}>Next</Button>

</div>

</div>

</div>

)

}

*export* *default* AntCaraousel

AntSteps

*import* React, { useState } *from* 'react'

*import* { Divider, Space, Steps } *from* 'antd';

const{ *Step* }=Steps;

const *AntSteps* =()=>{

const[*current*, *setCurrent*]= *useState*(1);

*return* (

<div>

<Space *direction*='vertical'>

                <Steps *current*={1}>

                    <Step *title*='Finished'></Step>

                    <Step *title*='In Progress'></Step>

                    <Step *title*='Started'></Step>

                </Steps>

                <Divider />

                <Steps *current*={current} *labelPlacement*='vertical' *onChange*={(c) => {

*setCurrent*(c)

                }}>

                    <Step *title*='Finished'></Step>

                    <Step *title*='In Progress'></Step>

                    <Step *title*='Started'></Step>

                </Steps>

                <Divider />

                <Steps *direction*='vertical' *current*={current} *labelPlacement*='vertical' *onChange*={(c) => {

*setCurrent*(c)

                }} *progressDot*={true} *percent*={75} *status*='error'>

                    <Step *title*='Finished' ></Step>

                    <Step *title*='In Progress' *subTitle*='2 min'></Step>

                    <Step *title*='Started' *description*='initiate after step 2'></Step>

                </Steps>

            </Space>

</div>

)

}

*export* *default* AntSteps

AntAccordian

*import* React *from* 'react';

*import* { Collapse } *from* 'antd';

const *text* =`

*A dog is a type of domesticated animal.*

*Known for its loyalty and faithfulness,*

*it can be found as a welcome guest in many households across the world.*

`;

const *items* =[

{

*key*:'1',

*label*:'This is panel header 1',

*children*:<p>{text}</p>,

},

{

*key*:'2',

*label*:'This is panel header 2',

*children*:<p>{text}</p>,

},

{

*key*:'3',

*label*:'This is panel header 3',

*children*:<p>{text}</p>,

},

];

const *App* =()=>{

const *onChange* =(*key*)=>{

*console.log*(key);

};

*return* (

<Collapse

*defaultActiveKey*={['1']}

*onChange*={onChange}

*expandIconPosition*="right"

        >

            {*items.map*(item => (

                <Collapse.Panel *header*={*item.*label} *key*={*item.*key}>

                    {*item.*children}

                </Collapse.Panel>

            ))}

        </Collapse>

);

};

*export* *default* App;

const *AntAccordian* =()=>{

*return* (

<div>

<Collapse *accordion*={true}

*defaultActiveKey*={['1']}

*expandIconPosition*='right'

*bordered*={false}

*ghost*={true}>

                <Collapse.Panel *header*=

                    "This is panel header 1" *key*="1">

                    <Typography.Text>

                        This is panel content 1

                    </Typography.Text>

                </Collapse.Panel>

                <Collapse.Panel *header*=

                    "This is panel header 2" *key*="2" *disabled*>

                    <Typography.Text>

                        This is panel content 2

                    </Typography.Text>

                </Collapse.Panel>

            </Collapse>

</div>