

zh-CN

使用 [rc-tween-one](#) 给标签增加添加或删除动画。

en-US

Animating the Tag by using [rc-tween-one](#).

```
import { Tag, Input } from 'antd';
import { TweenOneGroup } from 'rc-tween-one';
import { PlusOutlined } from '@ant-design/icons';

class EditableTagGroup extends React.Component {
  state = {
    tags: ['Tag 1', 'Tag 2', 'Tag 3'],
    inputVisible: false,
    inputValue: '',
  };

  handleClose = removedTag => {
    const tags = this.state.tags.filter(tag => tag !== removedTag);
    console.log(tags);
    this.setState({ tags });
  };

  showInput = () => {
    this.setState({ inputVisible: true }, () => this.input.focus());
  };

  handleInputChange = e => {
    this.setState({ inputValue: e.target.value });
  };

  handleInputConfirm = () => {
    const { inputValue } = this.state;
    let { tags } = this.state;
    if (inputValue && tags.indexOf(inputValue) === -1) {
      tags = [...tags, inputValue];
    }
    console.log(tags);
    this.setState({
      tags,
      inputVisible: false,
      inputValue: '',
    });
  };

  saveInputRef = input => {
    this.input = input;
  };
}
```

```

forMap = tag => {
  const tagElem = (
    <Tag
      closable
      onClose={e => {
        e.preventDefault();
        this.handleClose(tag);
      }}
    >
      {tag}
    </Tag>
  );
  return (
    <span key={tag} style={{ display: 'inline-block' }}>
      {tagElem}
    </span>
  );
};

render() {
  const { tags, inputVisible, inputValue } = this.state;
  const tagChild = tags.map(this.forMap);
  return (
    <>
      <div style={{ marginBottom: 16 }}>
        <TweenOneGroup
          enter={{
            scale: 0.8,
            opacity: 0,
            type: 'from',
            duration: 100,
          }}
          onEnd={e => {
            if (e.type === 'appear' || e.type === 'enter') {
              e.target.style = 'display: inline-block';
            }
          }}
          leave={{ opacity: 0, width: 0, scale: 0, duration: 200 }}
          appear={false}
        >
          {tagChild}
        </TweenOneGroup>
      </div>
      {inputVisible && (
        <Input
          ref={this.saveInputRef}
          type="text"
          size="small"
          style={{ width: 78 }}
          value={inputValue}
          onChange={this.handleInputChange}
          onBlur={this.handleInputConfirm}
        >

```

```
        onPressEnter={this.handleInputConfirm}
      />
    )}
    {!inputVisible && (
      <Tag onClick={this.showInput} className="site-tag-plus">
        <PlusOutlined /> New Tag
      </Tag>
    )}
  </>
);
}
}

export default () => <EditableTagGroup />;
```

```
.site-tag-plus {
  background: #fff;
  border-style: dashed;
}
```