如下几种放抖方法

1. **节流**

节流阻止函数在给定时间窗口内被调不能超过一次。

import throttle from 'lodash.throttle';

class LoadMoreButton extends React.Component {

  constructor(props) {

    super(props);

    this.handleClick = this.handleClick.bind(this);

    // handleClickThrottled 1s 可以被触发很多此

    // 但 handleClickThrottled 1s 内只会触发一次 handleClick

    this.handleClickThrottled = throttle(this.handleClick, 1000);

  }

  handleClick() {

    this.props.loadMore();

  }

  componentWillUnmount() {

    this.handleClickThrottled.cancel();

  }

  render() {

    return <button onClick={this.handleClickThrottled}>Load More</button>;

  }

}

1. **防抖**

防抖会延迟事件的触发

import debounce from 'lodash.debounce';

class Searchbox extends React.Component {

  constructor(props) {

super(props);

    this.emitChange = this.emitChange.bind(this);

    // emitChangeDebounced 被触发时会延迟 250 毫秒后触发 emitChange

    // 如果在 250 毫秒内再次触发，则重新刷新这个时间

    this.emitChangeDebounced = debounce(this.emitChange, 250);

  }

  emitChange(e) {

    this.props.onChange(e.target.value);

  }

  componentWillUnmount() {

    this.emitChangeDebounced.cancel();

  }

  render() {

    return (

      <input

        type="text"

        onChange={this.emitChangeDebounced}

        placeholder="Search..."

        defaultValue={this.props.value}

      />

    );

  }

}