

```
import type { RefObject } from 'react';
```

```
import { useEffect } from 'react';
```

```
type Event = MouseEvent | TouchEvent;
```

```
export const useOnClickOutside = <T extends HTMLElement = HTMLElement>(
```

```
  ref: RefObject<T>,
```

```
  handler: (event?: Event) => void,
```

```
) => {
```

```
  useEffect(() => {
```

```
    const listener = (event: Event) => {
```

```
      const el = ref?.current;
```

```
      if (!el || el.contains((event?.target as Node) || null)) {
```

```
        return;
```

```
      }
```

```
      handler(event);
```

```
    };
```

```
    document.addEventListener('mousedown', listener);
```

```
    document.addEventListener('touchstart', listener);
```

```
    return () => {
```

```
      document.removeEventListener('mousedown', listener);
```

```
      document.removeEventListener('touchstart', listener);
```

```
    };
```

```
}, [ref, handler]);
```

```
};
```