

# **REACT'S TACKLE BOX : USING THE RIGHT HOOKS FOR THE JOB.**

Bryan Pitt / Signal AI - London

# **OUR CHALLENGE**

**2,409 Organisations**

**6,807 Users**

**54,405 Reports**

**BIG  
FORMS.**

# WHAT WE NEED

- △ Performant controls.
- △ Efficient state management.
- △ Control over our UI.

# HOW WE DO IT

# React-Use

- useDebounce

```
import { DependencyList } from 'react';
import useUpdateEffect from './useUpdateEffect';

const useDebounce = (fn: () => any, ms: number = 0, deps:
DependencyList = []) => {
  useUpdateEffect(() => {
    const timeout = setTimeout(fn, ms);

    return () => {
      clearTimeout(timeout);
    };
  }, deps);
};

export default useDebounce;
```

# useDebounce for efficient API calls

```
<UserInput />
```

Saved Value:

# React-Use

- useUpdateEffect

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

const useUpdateEffect: typeof useEffect = (effect, deps) => {
  const isInitialMount = useRef(true);

  useEffect(() => {
    if (isInitialMount.current) {
      isInitialMount.current = false;
    } else {
      return effect();
    }
  }, deps);
};

export default useUpdateEffect;
```



# UI Hooks

- useFormBlur

```
function useFormBlur(innerRef, setIsEdit, omitSelectors) {  
  function handleBlur(e) {  
    if (document.querySelectorAll(omitSelectors).length) return;  
    if (innerRef.current && !innerRef.current.contains(e.target))  
  {  
    setIsEdit(false);  
  }  
}  
  
useEffect(() => {  
  document.addEventListener('mousedown', handleBlur);  
  return () => {  
    document.removeEventListener('mousedown', handleBlur);  
  };  
});
```

# UI Hooks

- useHover

```
function useHover() {
  const [value, setValue] = useState(false);
  const ref = useRef(null);
  const handleMouseOver = () => setValue(true);
  const handleMouseOut = () => setValue(false);

  useEffect(() => {
    const node = ref.current;

    if (node) {
      node.addEventListener('mouseover', handleMouseOver);
      node.addEventListener('mouseout', handleMouseOut);

      return () => {
        node.removeEventListener('mouseover', handleMouseOver);
        node.removeEventListener('mouseout', handleMouseOut);
      };
    }
  }, []);
```

# Why

CSS (in its current version) cannot do this. You'll need to turn to your jQuery solution.

2 not possible with css – [Mr\\_Green](#) Jun 22 '16 at 5:59

Implement jQuery. :) – [GROVER.](#) Jun 22 '16 at 5:59

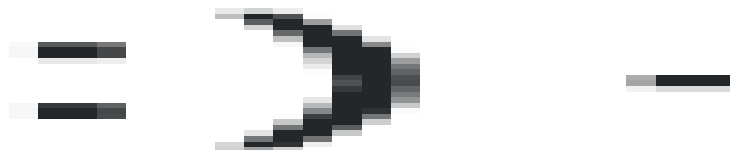
2 Possible duplicate of [Is there a CSS parent selector?](#) – [Mr\\_Green](#) Jun 22 '16 at 5:59

@CaelanGrgurovic : i want to try it with css only – [WebStarter](#) Jun 22 '16 at 6:00

@Mr\_Green : Can you suggest something. – [WebStarter](#) Jun 22 '16 at 6:00

[show 8 more comments](#)

## Implement jQuery. :) –

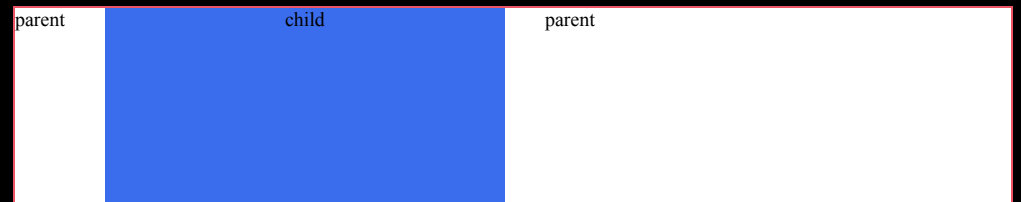


- *Stackoverflow, CSS-tricks*

# UI Hooks

- useHover

```
<HoverElement />
```



- *fin*