



## 17-2 React 中的事件注册

### React 中的事件

react/packages/react-dom-bindings/src/events/DOMEventNames.js

```
JavaScript
import getVendorPrefixedEventName from './getVendorPrefixedEventName';

export type DOMEventName =
  | 'abort'
  | 'afterblur' // Not a real event. This is used by event experiments
  | 'beforeblur' // Not a real event. This is used by event experiment
  | 'beforeinput'
  | 'blur'
  | 'canplay'
  | 'canplaythrough'
  | 'cancel'
  | 'change'
  | 'click'
  | 'close'
  | 'compositionend'
  | 'compositionstart'
  | 'compositionupdate'
  | 'contextmenu'
```

```
| 'copy'  
| 'cut'  
| 'dblclick'  
| 'auxclick'  
| 'drag'  
| 'dragend'  
| 'dragenter'  
| 'dragexit'  
| 'dragleave'  
| 'dragover'  
| 'dragstart'  
| 'drop'  
| 'durationchange'  
| 'emptied'  
| 'encrypted'  
| 'ended'  
| 'error'  
| 'focus'  
| 'focusin'  
| 'focusout'  
| 'fullscreenchange'  
| 'gotpointercapture'  
| 'hashchange'  
| 'input'  
| 'invalid'  
| 'keydown'  
| 'keypress'  
| 'keyup'  
| 'load'  
| 'loadstart'  
| 'loadeddata'  
| 'loadedmetadata'  
| 'lostpointercapture'  
| 'message'  
| 'mousedown'  
| 'mouseenter'  
| 'mouseleave'  
| 'mousemove'  
| 'mouseout'  
| 'mouseover'  
| 'mouseup'
```

```
| 'paste'
| 'pause'
| 'play'
| 'playing'
| 'pointercancel'
| 'pointerdown'
| 'pointerenter'
| 'pointerleave'
| 'pointermove'
| 'pointerout'
| 'pointerover'
| 'pointerup'
| 'popstate'
| 'progress'
| 'ratechange'
| 'reset'
| 'resize'
| 'scroll'
| 'scrollend'
| 'seeked'
| 'seeking'
| 'select'
| 'selectstart'
| 'selectionchange'
| 'stalled'
| 'submit'
| 'suspend'
| 'textInput' // Intentionally camelCase. Non-standard.
| 'timeupdate'
| 'toggle'
| 'touchcancel'
| 'touchend'
| 'touchmove'
| 'touchstart'
| 'volumechange'
| 'waiting'
| 'wheel';
```

```
export const ANIMATION_END: DOMEventName =
  getVendorPrefixedEventName('animationend');
export const ANIMATION_ITERATION: DOMEventName =
```

```
getVendorPrefixedEventName('animationiteration');
export const ANIMATION_START: DOMEventName =
  getVendorPrefixedEventName('animationstart');
export const TRANSITION_END: DOMEventName =
  getVendorPrefixedEventName('transitionend');
```

react/packages/react-dom-bindings/src/events/EventRegistry.js

JavaScript

```
// React中的事件
export const allNativeEvents: Set<DOMEventName> = new Set();

export const registrationNameDependencies: {
  [registrationName: string]: Array<DOMEventName>,
} = {};

// 事件注册
export function registerTwoPhaseEvent(
  registrationName: string,
  dependencies: Array<DOMEventName>,
): void {
  registerDirectEvent(registrationName, dependencies);
  registerDirectEvent(registrationName + 'Capture', dependencies);
}

export function registerDirectEvent(
  registrationName: string,
  dependencies: Array<DOMEventName>,
) {
  registrationNameDependencies[registrationName] = dependencies;

  for (let i = 0; i < dependencies.length; i++) {
    allNativeEvents.add(dependencies[i]);
  }
}
```

## 不同类型的事件注册

react/packages/react-dom-bindings/src/events/DOMPluginEventSystem.js

JavaScript

```
SimpleEventPlugin.registerEvents();
EnterLeaveEventPlugin.registerEvents();
ChangeEventPlugin.registerEvents();
SelectEventPlugin.registerEvents();
BeforeInputEventPlugin.registerEvents();
```

## SimpleEventPlugin

普通事件，如 click、drag、drop 等。

react/packages/react-dom-bindings/src/events/plugins/SimpleEventPlugin.js

JavaScript

```
import {
  registerSimpleEvents,
} from '../DOMEventProperties';

export {registerSimpleEvents as registerEvents};
```

react/packages/react-dom-bindings/src/events/DOMEventProperties.js

JavaScript

```
const simpleEventPluginEvents = [
  'abort',
  'auxClick',
  'cancel',
  'canPlay',
  'canPlayThrough',
  'click',
  'close',
  'contextMenu',
  'copy',
  'cut',
  'drag',
  'dragEnd',
  'dragEnter',
  'dragExit',
  'dragLeave',
```

```
'dragOver',
'dragStart',
'drop',
'durationChange',
'emptied',
'encrypted',
'ended',
'error',
'gotPointerCapture',
'input',
'invalid',
'keyDown',
'keyPress',
'keyUp',
'load',
'loadedData',
'loadedMetadata',
'loadStart',
'lostPointerCapture',
'mouseDown',
'mouseMove',
'mouseOut',
'mouseOver',
'mouseUp',
'paste',
'pause',
'play',
'playing',
'pointerCancel',
'pointerDown',
'pointerMove',
'pointerOut',
'pointerOver',
'pointerUp',
'progress',
'rateChange',
'reset',
'resize',
'seeked',
'seeking',
'stalled',
```

```

    'submit',
    'suspend',
    'timeUpdate',
    'touchCancel',
    'touchEnd',
    'touchStart',
    'volumeChange',
    'scroll',
    'scrollEnd',
    'toggle',
    'touchMove',
    'waiting',
    'wheel',
  ];

function registerSimpleEvent(domEventName: DOMEventName, reactName: string) {
  topLevelEventsToReactNames.set(domEventName, reactName);
  registerTwoPhaseEvent(reactName, [domEventName]);
}

export function registerSimpleEvents() {
  for (let i = 0; i < simpleEventPluginEvents.length; i++) {
    const eventName = ((simpleEventPluginEvents[i]: any): string);
    const domEventName = ((eventName.toLowerCase(): any): DOMEventName);
    const capitalizedEvent = eventName[0].toUpperCase() + eventName.slice(1);
    registerSimpleEvent(domEventName, 'on' + capitalizedEvent);
  }
  // Special cases where event names don't match.
  registerSimpleEvent(ANIMATION_END, 'onAnimationEnd');
  registerSimpleEvent(ANIMATION_ITERATION, 'onAnimationIteration');
  registerSimpleEvent(ANIMATION_START, 'onAnimationStart');
  registerSimpleEvent('dblclick', 'onDoubleClick');
  registerSimpleEvent('focusin', 'onFocus');
  registerSimpleEvent('focusout', 'onBlur');
  registerSimpleEvent(TRANSITION_END, 'onTransitionEnd');
}

```

## EnterLeaveEventPlugin

react/packages/react-dom-  
bindings/src/events/plugins/EnterLeaveEventPlugin.js

JavaScript

```
function registerEvents() {  
  registerDirectEvent('onMouseEnter', ['mouseout', 'mouseover']);  
  registerDirectEvent('onMouseLeave', ['mouseout', 'mouseover']);  
  registerDirectEvent('onPointerEnter', ['pointerout', 'pointerover']);  
  registerDirectEvent('onPointerLeave', ['pointerout', 'pointerover']);  
}
```

## ChangeEventPlugin

react/packages/react-dom-bindings/src/events/plugins/ChangeEventPlugin.js

JavaScript

```
function registerEvents() {  
  registerTwoPhaseEvent('onChange', [  
    'change',  
    'click',  
    'focusin',  
    'focusout',  
    'input',  
    'keydown',  
    'keyup',  
    'selectionchange',  
  ]);  
}
```

## SelectEventPlugin

react/packages/react-dom-bindings/src/events/plugins/SelectEventPlugin.js

JavaScript

```
function registerEvents() {  
  registerTwoPhaseEvent('onSelect', [  
    'focusout',  
  ]);  
}
```



```

    'contextmenu',
    'dragend',
    'focusin',
    'keydown',
    'keyup',
    'mousedown',
    'mouseup',
    'selectionchange',
  ]);
}

```

## BeforeInputEventPlugin

react/packages/react-dom-

bindings/src/events/plugins/BeforeInputEventPlugin.js

JavaScript

```

function registerEvents() {
  registerTwoPhaseEvent('onBeforeInput', [
    'compositionend',
    'keypress',
    'textInput',
    'paste',
  ]);
  registerTwoPhaseEvent('onCompositionEnd', [
    'compositionend',
    'focusout',
    'keydown',
    'keypress',
    'keyup',
    'mousedown',
  ]);
  registerTwoPhaseEvent('onCompositionStart', [
    'compositionstart',
    'focusout',
    'keydown',
    'keypress',
    'keyup',
    'mousedown',
  ]);
}

```

```
registerTwoPhaseEvent('onCompositionUpdate', [  
  'compositionupdate',  
  'focusout',  
  'keydown',  
  'keypress',  
  'keyup',  
  'mousedown',  
]);  
}
```