15/3/24

Debouncing in JS

Case study:

Observe any e-commercee search bar, eg. flipkart, there are always a auto suggest api called when we search something in search bar, but it doesn't call api for every key stroke, it checks if there is some delay between keys and if there is delay, then it fires one query.

Code

1) Create a search bar in html

```
<input type="text" onkeyup="getDataAfterPause()" >
```

JS

```
function getData() {
    console.log('data called', counter++);
}

const getDataAfterPause = doSomeMagic(getData, 3000)
```

↳ This fn is used to call get Data after delay

```
const doSomeMagic = function (fn, d) {
        let timer;
        return function () {
          console.log(this);
            let context = this;
          clearTimeout (timer);
          timer = setTimeout (() => {
              Console.log ('setTimeout called');
              fn.apply (context);
          }, d)
        }
```

Debouncing & Throttling in JS

Limiting the rate of execution of function calls can optimize your application

Throttling

Same as debouncing, but it does not depend on user pause, it always calls apis after certain amount of time.

Debouncing → call if user pauses for 300ms
Throttling → Call if previous fn was called 300ms ago

*Debouncing makes more sense

Use Case 2:

On Resizing the window, it triggers fns almost 1000 times, So to trackResize(), we use debouncing and throttling to limit the rate

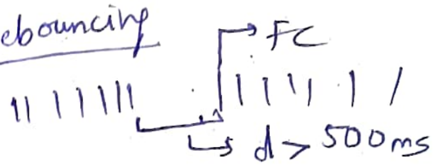
```
add Ele ("resize", () => { trackResize() });
```

User may resize slow or fast
Throttling may seem fine in this scenario

Use Case 3:

Button clicked frequently (While playing games, person clicking to shoot bullets)

Debouncing



Throttling

If you have machine gun, you can observe throttle case

Which is better debouncing / throttling?

A: Always depends on use case

16/3 Throttling

Throttling is a mechanism to limit the rate of execution of methods on some events.

Eg: window resize,
on window resizing there will be 100's of calls for 10ms, how do we not trigger an expensive fn when resizing, it will be by writting some better expensive function by indulging throttle

* Expensive function refers to the function which is making api call (not a JS terminology)

Code //Refer to github.

html

```
<button onClick= "getDataAfterPause ('weather') > Click for
temperature </button>
```

js

```
let counter = 0
getWeatherData ( ) {        counter ++;
  console.log ('weather Data');
}

Const throttleWeatherData = function (~~getWeatherData~~ fn, delay) {
    return function ( ) {

}
```

```
function ( ) {
 if let fnInProgress = false;
   return   function ( ) {  let args = arguments;
          let context = this;
          if ( ! fn inProgress) { fn.apply (context, args);
              fnInProgress = true;   sto (( ) => { }, delay)
}
```

const getDataAtloPause = throttle [illegible]

Interview questions on debouncing and throttling

Debouncing → flipkart Search

Throttling → Twitter scroll bar

Implement throttle, debounce, throttle pollyfill, debounce pollyfill

Refer to github JS Topicwise