# 防抖的实现

在许多的提交场景都是一个及其复杂的处理请求，后端处理每个请求都是非常耗时的，所以这个时候，需要限制用户在一定时间内重复提交，来节省后端的资源，防止服务器宕机或者内存泄漏等问题的出现，具体的实现如下所示。

|  |
| --- |
| // func代表一个函数，里面可能会频繁发生回调或ajax请求，delay表示隔多少秒触发才触发，immediate表示是否立即执行，可传true或false，默认false  function debounce(func,delay,immediate) {  // result表示返回值  let timeout,result  let debounced = function () {  // 存储触发当前事件的this  let \_this = this  // 存储event对象  let args = arguments  clearTimeout(timeout)  // 判断是否立即执行，如果不传默认不立即执行  if (immediate){  // 立即执行  let callNow = !timeout  timeout = setTimeout(()=>{  timeout = null  },delay)  if (callNow) result = func.apply(\_this,args)  } else{  timeout = setTimeout(function () {  func.apply(\_this,args)  },delay)  }  return result  }  debounced.cancel=function () {  clearTimeout(timeout)  timeout = null  }  return debounced  } |