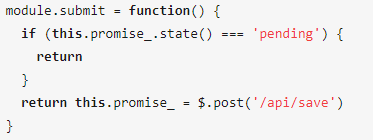
[[PromiseStatus]]

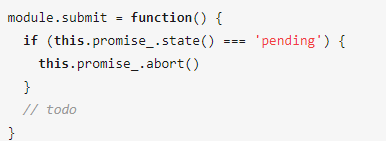
[[[PromiseValue]]](https://segmentfault.com/q/1010000010670739)

# 怎样防止重复发送 Ajax 请求？

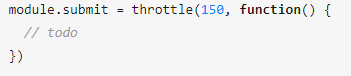
A. 独占**型提交**



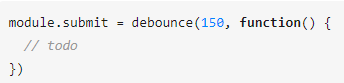
B. **贪婪型提交**



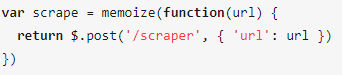
**C. 节制型提交**



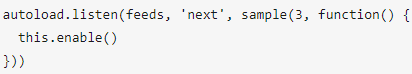
**D. 懒惰型提交**



**E. 记忆型**



**G. 采样型**



对于独占型及贪婪型提交需要查询promise的状态，而promise的状态是内部属性只能在then中访问，不能在外部获取结果，先找到一个获取状态而不改变原有的逻辑。



function MakeQuerablePromise(promise) {

// Don't modify any promise that has been already modified.

if (promise.isResolved) return promise;

// Set initial state

var isPending = true;

var isRejected = false;

var isFulfilled = false;

// Observe the promise, saving the fulfillment in a closure scope.

var result = promise.then(

function(v) {

isFulfilled = true;

isPending = false;

return v;

},

function(e) {

isRejected = true;

isPending = false;

throw e;

}

);

result.isFulfilled = function() { return isFulfilled; };

result.isPending = function() { return isPending; };

result.isRejected = function() { return isRejected; };

return result;

}

var originalPromise = new Promise(function(resolve,reject){

setTimeout(function(){

resolve("Yeah !");

},10000);

});

var myPromise = MakeQuerablePromise(originalPromise);