# How do I control repeated Ajax 'post' submissions?

Asked 15 years, 10 months ago Modified 15 years, 10 months ago

Viewed 654 times

0

I do not want to stop the user from clicking the same ajax button many times, but I want to exercise some control to prevent someone from maliciously clicking the button repeatedly which would cause repeated database requests. What is simple way to handle this on a jquery .post in asp.net mvc? If I put in a time delay, the experience client side will be ruined (?)

asp.net javascript asp.net-mvc asp.net-ajax

Share

Improve this question

Follow

asked Feb 5, 2009 at 22:46

zsharp

13.8k ● 29 ● 89 ● 155

## 7 Answers

Sorted by: Highest score (default)

6

Why not disable the button after it is clicked, and then re-enable it once you have received the response from the server?

Share Improve this answer Follow

answered Feb 5, 2009 at 22:50

ine
14.1k ● 8 ● 58 ● 80

Simple, effective, there are ways to send requests without clicking the button but you just want to make it difficult enough for the average joe to be discouraged from doing it. – Mark Feb 5, 2009 at 23:55

@AdamRalph: sorry! but not intentionally at all. you were 16 seconds ahead of me – ine Feb 6, 2009 at 3:30

3

This is not a client-side issue and can only be handled on the server side. If an AJAX request is being made, then a malicious user can make the AJAX request directly to your server directly through HTTP--it's quite trivial even for newbies using something like WFtech or Curl.

Not to state the obvious, but if a user can click a button multiple times, then so can an adversary and there's no way of really determining who is who. You can do some kind of throttling to limit it to x clicks per y seconds etc but then the adversary can also throttle his click rate.

Share Improve this answer Follow

answered Feb 5, 2009 at 23:01

aleemb
32k ● 20 ● 101 ● 114

+1 for being right ;-) The server-side script that processes the AJAX requests can and should limit the rate at which it makes database requests. – David Z Feb 5, 2009 at 23:18

hmm, request to the server is a request to the server, AJAX or not, malicious user can cause DoS, wether application uses AJAX or not. I belive point is to stop legitimate users from making too many requests at a time. ( tokens could be used to verify requests source ) – CountZero Feb 5, 2009 at 23:18

@CountZero, this is not a DoS issue. Tokens don't work because the adversary can also get tokens. – aleemb Feb 6, 2009 at 12:17

2

set a flag in your javascript indicating pending request, once response arrives unset the flag.

This way user can click button as many times as he wishes, there's gonna be only one 'active' request to the server.

Share Improve this answer Follow

answered Feb 5, 2009 at 22:55

CountZero
2,872 ● 4 ● 22 ● 20

2

Why don't you keep track of how many times the user is pressing the button, and every one or one-half second, you run the method to submit the query along with the number of clicks (or don't do anything if no clicks are made). Also, in the method reset the number of clicks to 0 so the next query will be accurate.

Share Improve this answer

Follow

answered Feb 5, 2009 at 22:57

Mike

3,247 ● 24 ● 29

---

1

Disable the button until the request is complete?

Share Improve this answer

Follow

answered Feb 5, 2009 at 22:50

Adam Ralph

29.9k ● 5 ● 61 ● 68

---

1

You can put the button on a timer so once pressed, it will be disabled for X seconds or until it receives a response from the server.

Share Improve this answer

Follow

answered Feb 5, 2009 at 22:52

Logan5

626 ● 4 ● 11

---

1

An approach that applies in fact to any post made is to add a random token generated by client. When the server receives the post the first time it records the token, any subsequent request coming with the same token is then ignored.

I heard about that on .Net Rocks #367 with Udi Dahan

Share Improve this answer Follow

edited Feb 5, 2009 at 23:31

answered Feb 5, 2009 at 23:15

Simon Laroche

**542** ● 2 ● 11