Twitter Thread by ■ Ed Guiness

■ Ed Guiness
@KiwiCoder

A long time ago I coded up a feature for SocialCoder which converts a link - any link - to a short URL. I use it all the time for sharing links to volunteer profiles and to volunteer opportunity listings.

The feature doesn't care what the link contains, it just hashes the URL, including any query params, and returns a shortened code.

More recently, around Christmas time, I added another feature where I can click a button to send a template notification to a charity that their new listing has been reviewed and published. I call it a next-steps email.

For the past few weeks, its all been working well. Its nice to keep automating things that previously were done by hand.

Fast forward to today.

A new opportunity listing is posted by a charity.

I review and publish it, tweet about it, then click a button to send the charity an emailed notification of next steps.

All very much business as usual, except that I normally tweet about it *after* sending that next-steps email to the charity. I don't know why I deviated from my normal routine, not that its a big deviation, nor should it matter.

But, oh, what a difference that slight deviation from my normal workflow did make.

Turns out I accidentally tweeted out a shortened link that results in the site sending a next-steps email to the charity. Anyone following that link will result in another email being sent.

Almost immediately my inbox gets spammed with flying in as people click on that link, also spamming the poor charity, and proper flustering me I can tell you.

At this point I had no idea what was going on.

Could it be a mail provider problem? Unlikely.

Maybe a bug, an infinite loop or recursion in my code? More likely.

I restarted the site, but the emails kept coming.

I changed the notification email address so that it would only spam me, and not the charity.

And still the emails came.

But now that it was just me being spammed, and not the charity rep, I was able to calm down enough to see what had happened.

A relatively easy fix, and a forehead-slapping moment.

You could say it was a learning moment. So what did I learn?

Lesson 1.

Don't use HTTP GET when a POST is more appropriate. If the request parameters were in the body of the request, and not in the URL, the link would have been fine to share.

Lesson 2.

The feature that sends email should not have been available to anonymous users.

In coding terms, the controller action was missing an [Authorize] attribute. yeah ...oops.

Lesson 3.

Although the consequences of this mistake were relatively minor, affecting only me and the charity rep's Inbox, I still needed to calm down before I could see the problem clearly enough to effectively trouble-shoot. Maybe I'm drinking too much coffee. Maybe.

Lesson 4.

Being able to code a fix, run unit tests, and deploy that fix, all within within minutes is such a valuable thing.

Thank you to the @Azure team who made this so easy.

- end for now -