

Exploring the variations in email UN subscription methods

Posted At : 20 November 2009 16:20 | Posted By : Shaun McCran
Related Categories: Software Architecture, Best practices, Web technologies

If you are anything like me, everyday you are smothered in a stack of emails from various companies all delivering html formatted, image heavy flyers persuading you to buy whatever dead horse they are flogging. I know it is my own fault, after all I signed up to them in the first place, I've only got myself to blame. As I find myself checking (and deleting) these email on an Android mobile device now it has become hassle-some. So I set out to unsubscribe from the majority of them, which is where the inspiration for this article came from.

In a short period of time I have unsubscribed from around a dozen different email newsletters, and because of this the differences in the methods used to accomplish this have become glaringly obvious. I was very surprised at the variations in the methods used, some are very user friendly, and others are very much not. I won't name the companies involved.

Case one:

By far the easiest user interaction, this is a simple link from an email that provides you with a screen confirming your UN subscription. No prompt is required from the user, you are simply unsubscribed. All the data is passed seamlessly behind the scenes.

Case two:

Almost the same as above except that you are taken to a screen where you have to commit an action. You are presented with a screen displaying your email address, and prompted to click a button to confirm the UN subscribe action. This may be a handy safety net to step errors from the casual clicker.

Case three:

Clicking on the link from an email takes you to a generic page with a form on it. They have not transferred any form of token from the email so you are prompted to enter the email address you want to UN subscribe.

Entering your email address takes you to a screen confirming your action. This additional step is unnecessary, and doesn't add any value at all.

Case four:

In this scenario we are passed to a page where we have to login. The subscription details are attached to an account application, and as such your credentials need to be verified. Based on passing an encrypted token from the email a user could easily be logged in automatically. Actually requiring a username and password adds an additional level of complexity. Once I logged in I then had to navigate to the subscription settings and choose to unsubscribe with a form similar to the ones mentioned above.

Case five:

In the most convoluted example I have come across so far I actually had to create an account. I arrived at the UN subscribe page and found that to subscribe did not require an account, but to change my subscription options did! This was a very strange scenario, requiring a user to create an account to stop emails. Once you have created an account the subscription options were managed through a form like the ones mentioned above.

Conclusions

I think the thing to keep in mind with functionality like this is the balance between usability and the business objectives. An obvious point to note is that businesses do not want their users to UN subscribe to their emails. This is their user base and they do not want to lose it. Still it is worth considering just how you interact with your customers, even when providing functionality you do not want them to use.

After all you can always just 'Mark as Spam'.