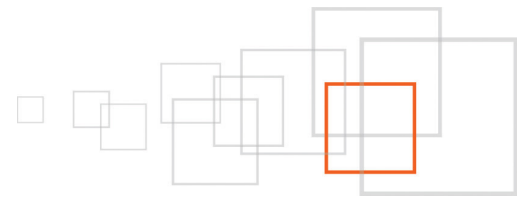


Encapsulating e-mails for usability and spam protection

By Jorge Estévez Rams
www.elfosdesign.com



Index

1 Goal description.....	3
2 Introduction.....	3
3 Pre-requisites and target population.....	3
4 Step 1 : Setting up template.ini.append.php.....	3
5 Step 2 : Adding usability using jQuery.....	4
6 Conclusion.....	5
7 Resources.....	5
8 About the author : Jorge Estévez Rams.....	5
9 License	5

1 Goal description

At the end of this tutorial you will have a new way of encapsulating e-mails avoiding crawlers to collect and use them as spam. Users at the frontend will have a well-known e-mail sequence to click thus avoiding confusion. Developers will not sacrifice usability versus spam-protection.

2 Introduction

Having spam is something we as developers, must deal with daily, avoiding spam is a must that we should keep in mind in all our projects. The usual method of hiding an e-mail is to wrap it up with words like **at** and **dot**, this is a trend but when it comes to usability versus spam-fighting, users could get confused. Using the e-mail generation method of eZ Publish combined with jQuery leads us to a safer html code that will avoid further more e-mail-spam-grabbers and will bring up usability in our projects. Developers will gain protection for e-mails and users will rely on well written e-mail sequences.

3 Pre-requisites and target population

To understand this tutorial you will need a basic knowledge of jQuery (www.jquery.com) and knowledge on how to configure .ini files in an eZ Publish installation.

4 Step 1 : Setting up template.ini.append.php

Recently I wanted to make e-mail addresses "uncatchable" at a site to avoid crawlers that collect them, a solution was to configure sitesettings/your-site/template.ini.append.php and add under [WashSettings] the following:

```
# Replacements strings for the . and @ in an email
# Used to display emails in a way that SPAM engines
# cannot fetch automatically
EmailDotText= [punto]
EmailAtText= [@]
```

Adding such definitions e-mail addresses become something like "myname [@] domain [punto] com", you can also change it to the conventional "at" and "dot".

Hint: You can leave spaces before and after each definition " [punto] " or " [@] " to achieve spaces within the e-mail sequence

The code generated as html by eZ Publish looks something like this:

```
<a href="mailto:myname@domain.com">myname [@] domain [punto] com</a>
```

This kind of sequence for an e-mail addresses is not always a straight-forward solution. Avoiding the mailto

also keeps crawlers away, so with some jQuery code we can obtain a more interesting approach.

You can still let the eZ Publish kernel handle the e-mail address generation but this time surrounded by the <div> that jQuery code will then handle later on when the page has loaded.

An example of template code:

```
{ $owner_map.user_account.content.email|wash('email') }
```

Will generate something like this:

```
myname [@] domain [punto] com
```

Wrapped around the div and adding an id, this will become:

```
<div id="mail">myname [@] domain [punto] com</div>
```

5 Step 2 : Adding usability using jQuery

Enabling jQuery at an eZ Publish project can be done by activating the eZ JS Core certified extension as you would with any extension in eZ Publish. For more information, see <http://doc.ez.no/Extensions/eZ-JS-Core/>. You could also add the jQuery JS code in your pagelayout (visit www.jquery.com).

The jQuery code that deals with the <div> is:

```
<script type="text/javascript">
$(function(){
    //scrambles mails with [@] and [punto]
    var spt = $('div#mail');
    var at = " [@] ";
    var dot = " [punto] ";
    var addr = $(spt).text().replace(at,"@").replace(dot,".");

    $(spt).after( '<a href="mailto:' + addr + '" title="Send an email">' +
addr + '</a>' ).hover( function()
    {
        window.status = "Send a letter!";
    },
    function()
    {
        window.status="";
    }
    );
    $(spt).remove();
});
</script>
```

You can also add within the js code some eZ code that fetches from the template.ini the values of EmailDotText and EmailAtText so whenever you change the definition at the .ini the jQuery works as well.

Once the jQuery runs you get a nicer and understandable e-mail "myname@domain.com" at the frontend site

with its own "mailto" once you hover over the link, but this time crawlers will not see it at all as the code is not understandable for them, as for users they will get the well-known e-mail addresses sequence as it should always be.

6 Conclusion

Developers make projects for users, but not all of them have the knowledge enough to understand scrambled e-mail address sequences that should be interpreted as simple e-mails sequences and thus trust and click them. Usability is a must in the development of a project. In this tutorial we have learned how to achieve better code to protect our users from spam and an understandable e-mail address sequence for our users. Happy developer, happy user!

7 Resources

jQuery's site can be reached at www.jquery.com

eZ JS Core extension at <http://doc.ez.no/Extensions/eZ-JS-Core/>

8 About the author : Jorge Estévez Rams



Jorge Estévez Rams is using eZ Publish since eZ Systems was founded; he leads a web development group in Havana, Cuba (www.elfosdesign.com).

9 License

This work is licensed under the Creative Commons – Share Alike license (<http://creativecommons.org/licenses/by-sa/3.0>).

