# Building an email xmlrpc client

Posted At : May 14, 2012 1:27 AM | Posted By : Tim Garver
Related Categories: How Too's

This is a post I sent to my blog via email. I know there are lots of blog clients out there, but a email editor I think would work just as good.

I will post a few more with this to test it out more.

Ok, so I did post this with my cell phone email client, but I then came to the computer to finish it up so I could include the code sample.

I basically took the xmlrpc section from BlogCFC and made an email responder that checks for new email and it builds a blog post from those emails.

Here is the code which I have set as a scheduled task. There is a lot going on here, so ill explain it after:

```
<cfpop action="getheaderonly" name="qHeader2" server="mail.server" username="user@server" password="password">
 <cfquery name="qHeader" dbtype="query">
  SELECT uid,[from],[date]
        FROM qHeader2
        WHERE [from] = 'youremail@server.com'
        ORDER BY messagenumber ASC
 </cfquery>

 <cfif qHeader.RecordCount>
  <cfpop action="getall" name="qMessage" server="mail.server" username="user" password="pass" uid="#ValueList( qHeader.uid )#
        attachmentpath="#tempDir#"/>

  <cfset att = arrayNew(1)>
        <cfset p = structNew()>

        <cfloop query="qMessage">
      <!---first lets handle the image attachments if any--->
            <cfif Len(qMessage.attachments)>
             <cfloop list="#StructKeyList(qMessage.CIDS)#" index="I">
                    <cfset ns = structNew()>
                    <cfset tmp = replace(qMessage.CIDS[i],"<","","ALL")>
                    <cfset tmp = replace(tmp,">","","ALL")>
      <cfset newName = hash(listFirst(i,"."), "md5") & "." & listlast(i, ".") />
                    <cfset ns.old = i><cfset ns.new = newName><cfset ns.cid = tmp>
      <cfset arrayAppend(att,ns)>
                </cfloop>
            </cfif>
                <!--- next our actual message for the blog entry.
                        I am using the subject line to hold variables delimited with a ~ --->
      <cfset p.body = qMessage.body>
                <cfset p.cats = ListGetAt(qMessage.subject,1,"~")>
         <cfset p.subject = listGetAt(qMessage.subject,2"~")>
                <cfset p.released = listGetAt(qMessage.subject,3"~")>
                <cfset p.sendemail = listGetAt(qMessage.subject,4"~")>

            <cfif arrayLen(att)>
  <cfset p.body = replaceNoCase(p.body,"cid:","","ALL")>
                <cfset u = "../../enclosures/">
             <cfloop from="1" to="#ArrayLen(att)#" index="I">
              <cfset p.body = ReplaceNoCase(p.body,att[i].cid,u & att[i].new,"ALL")>
             </cfloop>
            </cfif>



        <!--- now we will use the xmlrpc service that is already in place to send our blog entry into blogcfc --->
        <cfset xmlrpc = createObject("component","xmlrpc.xmlrpc")>
        <cfset entry = structNew()>
  <cfset entry.title = p.subject>
  <cfset entry.body = htmleditformat(p.body)>
  <!---<cfset application.body = htmleditformat(p.body)>
   TODO: Handle <more/> --->

  <!---// replace the ellipse character with the HTML entity //--->
  <cfset entry.body = replace(entry.body, chr(8230), "&##8230;", "all") />
  <!---// replace the em dash character with the HTML entity //--->
  <cfset entry.body = replace(entry.body, chr(8212), "&##8212;", "all") />
  <cfset entry.body = replace(entry.body, chr(151), "&##8212;", "all") />
  <cfset entry.body = replace(entry.body, "--", "&##8212;", "all") />
  <cfset entry.body = replace(entry.body, chr(8211), "&##8211;", "all") />
```

```
 <cfset entry.body = replace(entry.body, chr(150), "&##8211;", "all") />
 <cfset entry.body = replace(entry.body, "-", "&##8211;", "all") />
 <cfset entry.body = xmlrpc.unescapeMarkup(entry.body) />
 <!--- Handle potential <more/> --->
  <!--- fix by Andrew --->
 <cfset strMoreTag = "<more/>">
       <cfset moreStart = findNoCase(strMoreTag,entry.body)>
       <cfif moreStart gt 1>
           <cfset moreText = trim(mid(entry.body,(moreStart+len(strMoreTag)),len(entry.body)))>
           <cfset entry.body = trim(left(entry.body,moreStart-1))>
       <cfelse>
           <cfset moreText = "">
       </cfif>

       <cfset entry.morebody = moretext>
       <cfset entry.posted = now() />
       <cfset entry.allowcomments = true>

       <cfset entry.enclosure = "" />
<cfset entry.filesize = 0 />
<cfset entry.mimetype = "" />
<cfset entry.released = p.released/>

       <cfset entry.sendemail = p.sendemail>
       <cfset entry.alias = application.blog.makeTitle(entry.title)>

<!--- next log into blogcfc and post our entry --->
       <cfloginuser name="#variables.username#" password="#variables.password#" roles="admin">

       <cfinvoke component="#application.blog#" method="addEntry" returnVariable="newid">
           <cfloop item="key" collection="#entry#">
               <cfinvokeargument name="#key#" value="#entry[key]#">
           </cfloop>
       </cfinvoke>
       <cfset entryid = newid>
       <cfset result = newid>

    <!--- associate a category if exists --->
       <cfset catlist = "">
<cfif structKeyExists(p, "cats")>
           <cfloop index="x" list="#p.cats#">
               <cfset catid = translateCategory(x)>
               <cfif isValid("UUID",catid)>
                <cfset catlist = listAppend(catlist, catid)>
               </cfif>
           </cfloop>
       <cfelse>
        <cfset catlist = "41C1CEA9-C0F7-B0AE-D24BE7399F23ED8B">
       </cfif>

       <cfif len(catlist)>
           <cfset application.blog.assignCategories(entryid, catlist)>
       </cfif>

       <!--- clear cache --->
       <cfmodule template="../tags/scopecache.cfm" scope="application" clearall="true">




    </cfloop>

       <cfpop action="delete" server="mail.server" username="user" password="pass" uid="#ValueList( qHeader.uid )#"/>

   <!--- mail yourself a copy just in case???
   <cfmail to="you" from="blog" subject="ePost" type="html">
     #now()#<BR>
     #m#<BR>
     <cfdump var="#qHeader#" label="mail">
   </cfmail>--->

   </cfif>
```

Lots going on here. its a big chunk of code. First it gets a list of emails that were sent BY YOU! to the specified blog email box. Next loop over those and builds out a struct with our vars to send into BlogCFC to build our post. Then at the end it deletes the email and sends you a copy if you want it.

So there you have it, a simple way to post a blog entry from a phone or any other email client.

Let me know if there is a better way to handle this, I would love to see it.