Vesijama, the Very Simple Java Mail library

by Benny Bottema - Monday, April 27, 2009

http://www.bennybottema.com/2009/04/27/vesijama-very-simple-java-mail/

Here's the story about why I wrote a mail API wrapper in Java and made it Open Source. It is called Vesijama, Very Simple Java Mail.

—

Vesijama has been renamed and moved to Simple Java Mail!

Prologue

So I was looking into sending some emails from my little Java webapp, right. I thought: easy, java has built in mail support called JavaMail. Well, I was in for an unpleasant surprise when I found out I was merely handed the sand grains to build my castle. I can't just say to Java's mailing library Hey, here's what I want to send, and here are the addresses I need it sent to, no. I can say however I want you to formulate a mixed multipart message with the following attributes and these addresses encoded into special objects and oh yeah, define some mimemessage bodyparts for these attachments and use the following headers and put everything in a very specific nested structure dot dot dot.

You actually need to research the RFC for mail structures to do it properly! In case you're interested (which you're not), check out this page of useful email rfc's. The rfc's 822, 2047 and 2045 are referenced in Java's MimePart alone. That's just stupid, I can't think of anything else where the JDK is forcing developers to work on such a low level where you need to deal with rfc's.

An example of the JavaMail madness can be found on a ~~thread at velocityreviews.com~~, where someone asks for a simple tutorial to learn how to simply send emails using JavaMail. Here's the answer of one of the members:

> First, you need to read and understand the relevant RFCs governing Internet
> email, if you don't already understand this. I'd recommend starting with:
>
> RFC 2821 – Simple Mail Transfer (SMTP)
> RFC 2822 – Format of Internet Messages
> RFC 1939 – Post Office Protocol (POP3)
> RFC 2060 – Internet Mail Access Protocol (IMAP)
>
> then moving on to MIME:
>
> RFCs 2045 – 2049 – Multipurpose Internet Mail Extensions (MIME)
>
> Once you understand this, you should be able to figure out JavaMail using
> the tutorial at http://java.sun.com/developer/onlineTraining/JavaMail/contents.html,
> (which is, incidentally, the first hit returned from a Google search on the
> term "javamail tutorial"), along with the JavaDoc that is shipped along
> with the JavaMail libraries. If you have any specific questions about how
> JavaMail is to be used, please post them to the news group and you'll be

> pretty sure to get an answer.
>
> Cheers!

With all his good intentions, this guy doesn't realize he's completely scaring the mail novice away; people looking for a way to send emails shouldn't be required to be RFC educated email engineers. I just now realize that maybe JavaMail wasn't meant to be used by Joe sixpack, but rather by third-party library vendors. Also, realize that JavaMail is much more than just smtp client API. Still, when I began writing Vesijama there weren't a whole lot of third-party options, let alone one as simple to use as Vesijama.

Don't get me wrong though: it's not that the JavaMail API is so bad, it's great actually. It allows for very finegrained configuration for pop3, smtp and everything mail related. It's just that you don't want regular Java developers to deal with API on such a low level: it takes a lot of time getting it right and the risk of a faulty implementation is very high, as the result will often work in *some* email (web) clients and subtle mistakes with significant consequences will go unnoticed.

Enter Vesijama, the Very Simple Java Mail library

To spare other developers the same fate, I decided to write a proper wrapper around the JavaMail smtp API so no Java developer needs to be bothered by this stuff and just keep pluggin' on an abstract business logic level. You know, just send the damn mails.

It took me quite a while to get it right since email clients, especially web clients, are very sensitive to incorrect nested structures. I'm happy to say I've succesfully used this when I was working for a commercial company, which means it has been professionally tested with a broad range of email (web) clients.

Enter Vesijama. It´s been a while since I wrote this little framework, I´ve only recently had the time to properly prepare the code for an Open Source release. At the time of writing the library I didn't know about Spring's mailing facility or Apache's Commons Mail libraries, but in hindsight I think there's a place for Vesijama since in my completely biased opinion it is by far the easiest to use framework. Look it up over at ~~**Vesjiama at Google Code**~~ simple-java-mail **at GitHub**.

Here's the one and only code example from the Vesijama project page:

```
final Email email = new Email();

email.setFromAddress("lollypop", "lolly.pop@somemail.com");
email.setSubject("hey");
email.addRecipient("C. Cane", "candycane@candyshop.org", RecipientType
.TO);
email.addRecipient("C. Bo", "chocobo@candyshop.org", RecipientType.BCC
);
email.setText("We should meet up! ;)");
```

```
email.setTextHTML("We should meet up!");

// embed images and include downloadable attachments
email.addEmbeddedImage("wink1", imageByteArray, "image/png");
email.addEmbeddedImage("wink2", imageDatesource);
email.addAttachment("invitation", pdfByteArray, "application/pdf");
email.addAttachment("dresscode", odfDatasource);

new Mailer("smtp.host.com", 25, "username", "password").sendMail(email
);
```

Oh wow, is that it? Well, I just imagine masses of people gazing with open mouths thinking that, but that’ll never happen because nobody realizes how much effort went in the rfc research. And the very few lines that make up the mail structure don’t look very impressive either… the thing is it has to be just right; mail clients aren’t very forgiving. It often amazes me how many thousands of lines of rfc text results in miniscule implementations, but luckily in this case at least if you do it right, it works in all email clients that properly handle emails according the rfc’s.

Anyway, the library suited me and now it’s yours too. Enjoy.
