An important step in any development is the installation and configuration of the development tools needed to do the work.

In keeping with my philosophy on this project, almost all the software described here is open source.

My operating system on the development PC is XP. This is because other, paying, customers of mine require development targeted at XP and I saw no reason to change just for this project.

For a coding/debug environment (IDE) I used eclipse - I really cannot think of any reason why any sane person would use anything else for largely Java projects? To this were added a number of plugins, notably the php coding plugin, database development, hibernate tools and google web toolkit tools. All the plugins except
GWT
and
hibernate
were from the standad eclipse update sites.

GWT debugging required a firefox plugin which would NOT automatically install correctly, so I had to manually remove it and find a download for a version compatible with my firefox version and then install that by hand.

For debugging, I used the GWT tools and the server built into eclipse. This worked quite well, but I also required firebug (a firefox plugi in which allows debugging of web pages - I cannot imagine doing any web development without this now!) in order to debug the javascript at the browser end of things.

For my test deployment I installed apache 2.2 , php 5.33 and apache tomcat 6.0.26 and 5.0.29. I then setup Apache to forward servlet requests to tomcat using
mod_jk

with automatic configuration by tomcat. i.e. Tomcat looks at what apps it has and then tells apache that all of those urls need to go to tomcat. I had the mod_jk setup with a very simple worker.properties that used a single (i.e. non-clustered) AJP13 interface.

I had to duplicate my Joomla installation from my live server, so I copied the Joomla directories and created a new local MySQL database with the same name and users as the one on the live host. I then ran a database backup on the server which generated a large SQL dump that i could then use to populate the local copy of the database on my development machine. The same procedure in reverse worked fine when I had to update the live joomla to match changes that I had made locally, with the exception of joomla plug-ins and modules that I had added - they had to be installed manually onthe live server too.

To calculate the SHA in a javascript client I used jsSHA, a free javascript SHA implementation, accessed via a native interface call embedded in the GWT java.

In Joomla I needed to use pHP to write a module and plugin to tie together the joomla dataspace with my GWT clients. More on that in a later blog.

GWT-RPC requires referencing the servlet in web.xml on the tomcat server. This would cause problems when moving to the shared tomcat server on mochahost - again, more on this later.

To create the cryptopuzzle tables in the MySQL database, I coded the SQL create table statements by hand and sent them via the MySQL command line console. I am sure there are better tools for this, but it worked for me. I also used the same technique to populate the tables with some test data.

For version control of my source code I setup SubVersion with the svnserve server (i.e. the simple alternative) running as a windows service, but started manually, since I did not want it to run all the time draining resources on my PC. I also used the
subclipse
Eclipse SVN plug in for SVN which makes it very simple to keep a project workspace under control. (I at no moment ever considered trying to do this or any other project without some form of version control.)

When moving to the Mochahost server, I discovered the value of log4j logging to the MySQL database, since this was practically then only way to get any diagnostic info out of the installation.

This setup consisted of quite a lot of disconnected tasks and might seem a bit daunting, but I did not set up everything at once - instead, I addressed each of them as they became necessary for the part of the project that I was working on.

In the next installment I will look at database access and using hibernate.