I’m currently investigating on how to create a Zimbra server extension. In case you didn’t know already, Zimbra is a great groupware system, that can easily replace your Microsoft Exchange server. More about Zimbra can be found here: [[http://www.zimbra.com]].

What I’m thinking about is a Zimbra server handler, that can communicate with backend ERP systems to update client data. But that’s something for the future.

As I’m not the great Java guy around, I fiddled around with the correct options and classpaths and stuff to create a very simple server extension, that just outputs a funny string when you call it using a web browser. And for your entertainment and as a reminder for myself, I made this blog post.

Some information on this can be found in Vishal Mahajan’s old blog post available [[http://blog.zimbra.com/blog/archives/2010/04/extending-zimbra-with-server-extensions.html|here]].

What we’re going to do is create an extension that outputs a string into our browser when calling a Zimbra-specific URL.

So, for Java misfits like me, let’s start by creating the following directory structure:

```
* TestExtension
* build
* dist
* src
* conf
```

* build: Output directory for the java compiler
* dist: Output directory for the JAR-file
* src: The source directory structure (see below)
* conf: Place for a manifest-file the way Zimbra wants it

To start the easy way: Vishal mentioned something about a jar manifest file, that needs to have some information about our extension. A „jar manifest file“ is something like a configuration file, that sits inside a jar file. And a „jar-file“ is a zip file of compiled java files in a specific directory structure.

So I created a file MANIFEST.MF in the conf directory consisting only of this line:

```
Zimbra-Extension-Class: de.dieploegers.TestExtension
```

About the right-hand side there: That's the name of our test extension. In Java terms, this is the Java class „TestExtension" in the package „de.dieploegers". So we create the following directory structure under the „src"-directory:

```
* de
* dieploegers
```

Now onto the code. We create a file called „TestExtension.java" below „src/de/dieploegers".

In this file, we first have to tell Java in which package we are:

```
package de.dieploegers;
```

Then we have to import some classes, so that we have everything we need:

```
import com.zimbra.cs.extension.ZimbraExtension;
import com.zimbra.cs.extension.ExtensionDispatcherServlet;
import com.zimbra.common.service.ServiceException;
```

Now, trying to develop a useful server extension, you definitely need some basic Java knowledge, that I won't teach you here. Google a bit, there are lots of good java tutorials.

What do we import here?

* ZimbraExtension: That will be our class' parent.
* ExtensionDispatcherServlet: That's a class that does the whole „HTTP-Connection to Servlet"-dispatching. Remember, that we wanted to do something once someone calls a specific URL.
* ServiceException: That's a basic exception you can throw or catch in this whole Zimbra extension business

Again, following Vishal here, we need to make a class, that basically overwrites „init" and „destroy". That is fairly easy. But we want to gain control of a HTTP connection connecting at http://