

Tomcat Logging

MARKDOWN I guess this is something pretty obvious for java users (except me). I have to admit that in development I did tend to use a `System.out.println(..)` logging. And everytime at a late time when it comes to deployment you realize (again) that syso-logging is not optimal. Switching to that late time to `log4j` and such is tedious work. Whatever, now that I make the wood-games-server fit for deployment again, I am at this point again. Actually 9 years ago I seemed to have used an ugly hack, discarding the `System.out-` and `System.err-Stream` like this: Don't:

```
``` System.setOut(new PrintStream(new OutputStream() { @Override public void write(int arg0) throws IOException { } })); System.setErr(new PrintStream(new OutputStream() { @Override public void write(int arg0) throws IOException { // TODO Auto-generated method stub } })); ``` Ok,...now I will do things right. And when using the servlet-based-server rest-server in tomcat (and I guess i any other 'application'-server as well) you can just grab the current active logger. (I'm actually not sure if that is the right way since I still think this goes into catalina.out-log which is shared by every servlet-application!?) (Really not sure,...) Nonetheless for my special case I'm going to create a docker image with just one application, so this should be fine for me...(If you want the logs to (also) go into a separate file, you would have to need to add an additional logger-setup in your logging.properties-file) TL;DR; Get the current logger: 

```
``` import java.util.logging.Logger; ... public class MyClass{ ... private Logger logger = Logger.getLogger(DatabaseLayer.class.toString()); ... } ```
```

 [Some video about tomcat-logging(speechless but good)](https://www.youtube.com/watch?v=LRC38v-fico)
```

Published by Thomas Trocha

Fri May 29 18:18:00 CEST 2020

<http://thomas.trocha.com:80/pebble/tomcat-logging>