DevelopmentTricks

Java Runtime configuration

It is recommended to customize some variables:

```
export JAVA_OPTS="-Djava.net.preferIPv4Stack=true -Xmx1024m -XX:PermSize=128m -XX:MaxPermSize=256m -XX:
+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled -XX:+UseConcMarkSweepGC -XX:+HeapDumpOnOutOfMemoryError"
```

Paths in Windows-based systems

In Windows some paths, the user home for instance, usually contain spaces, which could be problematic with some development environments. Therefore the recommendations are:

- Place your LMF working copy under a path without spaces
    - Tomcat, and maybe other servlet containers, are not happy with such paths
- Configure related tools (maven and so on) to use a home path also without spaces; these are some of the environment variables involved

Database configuration

Internally LMF uses a relational database to persist the data using a custom and efficient schema.

For instance, for creating such database in PostgreSQL you'd need to execute something like:

```
$ su
# su - postgres
$ psql
postgres=# CREATE USER lmf WITH PASSWORD 'lmf';
postgres=# CREATE DATABASE lmf WITH OWNER lmf;
```

Then you'd need to configure the access to it, using the userConfig.properties file or through the web interface under LMF core.

Performance

These performance tricks may be of your interest.

Maven

See Maven for further details.

Hot Deployment / JRebel

See JRebel for further details.