Getting started with *lift*

- Install Java 5 on your machine (http://java.sun.com)
- Install Scala 2.3.3 on your machine (http://scala-lang.org)
- Install Maven 2 on your machine (http://maven.apache.org/)
- Download the *lift* source code from GoogleCode:
  `svn checkout http://liftweb.googlecode.com/svn/trunk/ liftweb`
- cd into "liftweb" and type: `mvn install`
- cd into "example" and type: `mvn jetty:run`
- Point your browser to http://localhost:8888 and you should see *lift* running

Making changes to the sample application using Jetty

- Open a second terminal window
- cd to "liftweb"
- edit a file (e.g., lift/example/src/main/scala/net/liftweb/example/controller/Chat.scala)
- type: `mvn install`
- When the compilation is done, Jetty will automatically restart itself and you can reload the browser and your new code will be executing. You will, however, lose all your session state and it takes a long time to go through this cycle.

Making changes to the sample application using Tomcat and Eclipse

- Install Tomcat (http://tomcat.apache.org/download-55.cgi#5.5.20)
- Install Eclipse (http://www.eclipse.org/downloads/)
- Install the Scala Eclipse Plugin (http://www.scala-lang.org/downloads/eclipse/index.html)
- Install the SVN Eclipse Plugin (http://subclipse.tigris.org/)
- Create a new SVN location for `http://liftweb.googlecode.com/svn/trunk/`
- Check out the *lift* project
- From the command line, cd into "workspace/liftweb" and type: `mvn install`
- cd into "example/src/main/webapp/WEB-INF/"
- Link your "bin" (compiler output) directory to WEB-INF/classes by typing: `ln -s ../../../../../bin/ classes`
- Copy the JARs loaded into the WAR during the "mvn install" by typing: `cp ../../../../target/lift-example-0.1.0/WEB-INF/lib/* lib/` and then `rm lib/lift*.jar`
- Yeah... I know, this is a lot of steps, but it will get easier as *lift* matures
- In your tomcat conf/Catalina/localhost directory, create a file named lift.xml with the following contents:

```
<Context docBase="path_to_workspace/workspace/lift/example/src/main/webapp"
         path="/lift" reloadable="true" allowLinking='true'>
</Context>
```

- cd into tomcat/bin and type: `./startup.sh` and point your browser to http://localhost:8080/lift and you should see *lift* running
- In Eclipse, make a change to one of the classes, wait 10 seconds for Tomcat to reload the context, reload the page and you'll see the change.

This sucks compared to Rails where I can make a change and reload

Yep. A big part of the problem is that Scala Actors interact badly with Tomcat's classloader. I'm working on the situation in two ways:

1. Trying to get Actors to suspend themselves during the servlet context reload cycle. This may address the issue.
2. Working on finding a JVM that supports better hot code replacement. Apparently, the IBM JVM allows changing class signatures and other fun stuff without requiring program reload. There's a bug running Scala under the IBM JVM, but the Scala team is looking into it.