All code samples from the book (except Lesson 37)come as Eclipse projects on the DVD. In order to build and deploy the application

from Lesson 37 and a bonus online store application, you have to use maven build tool

Download GlassFish Enterprise Server v3

Follow the instructions form Lesson 26 of the book

To get the source code, Install Mercurial version control system

To check if you already have Mercurial, open terminal (Mac Os X/Linux) or command window on Windows and type

hg --version

You should see something like

Mercurial Distributed SCM (version 1.7+201...)

In case you don't see this, download mercurial fromhttp://mercurial.selenic.com/

Good Mercurial Tutorial is here: http://hginit.com/

Get source code

In your terminal, go to the your projects directory, let say

cd ~/development/projects

Get a local copy of the book code samples

hg clone https://practicaljava.googlecode.com/hg/ java24hourtrainier

Optional: Install Maven

You'll need maven only for building sample online store application and building the code for the lesson 37 (JavaFX sample)

Mac Os X/Linux users

To check if you already have maven, open the terminal window and type

mvn -v

You should see something like

Apache Maven 2.2.0

Java version ...

In case you don't see this

If you are on Debian/Ubuntu Linux * type 'sudo apt-get install maven2' If you are on Mac Os * download zip version from http://maven.apache.org/download.html * unpack it (by just opening it) and copy the content to, let say, ~/development directory * extend your PATH environment variable to point to newly downloaded maven directory. In your terminal

export PATH=~/development/apache-maven-2.2.1/bin/:$PATH

To build your source code and deploy result on GlassFish

- Open terminal
- Check that all necessary environment variables are set

  Mac Os X:

  ** Maven binary in your PATH*

  *export PATH=~/development/apache-maven-2.2.1/bin/:$PATH * Java home is set to Java 6 (Omit this step for Snow Leopard version) export JAVA_HOME="/System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Home" * If you installed GlassFish to specific directory and not the default one export GLASSFISH_HOME=/path/to/glassfish*

  You can put these lines in ~/.profile file for future terminal sessions
- Change current directory to the specific lesson directory

  cd ~/java24hourtrainier/lesson37
- To build any artifact type

  mvn install
- To rebuild any artifact type

  mvn clean install
- To deploy artifact and create necessary resources (datasource, auth realms, etc) we need to build our own maven plugin

  cd ~/java24hourtrainier/maven-plugins

  mvn install
- Now we can use https://glassfish.dev.java.net/GlassFish manipulation plugin to create domains, resource, etc.

  First, we need to create domain.

  mvn com.practicaljava.maven.plugin:gf-plugin:create-domain

  This looks really verbose, instead we would like to use

  mvn gf:create-domain

  To be able to use our own plugin this way we need to say to maven that plugins from our group **com.practicaljava.maven.plugin** *are plugins we will use manually and quite often.*

  On ***Mac Os X / Linux*** *open ~/.m2/settings.xml file and make sure that file contains*

<settings> <pluginGroups> <pluginGroup>com.practicaljava.maven.plugin</pluginGroup> </pluginGroups> </settings>

  After settings are set we can use

  mvn gf:create-domain
- To start domain

  mvn gf:start-domain
- Now we can deploy our application

  mvn gf:deploy

  we can even combine code building and deploy like

  mvn clean install gf:deploy

  Application online-store which was took for explanation needs database access and uses authentication realm, so deploy on this stage will contains errors, something like

  Lookup failed for 'jdbc/practicaljava' in SerialContext
- Create data source and jdbc resource

  mvn gf:create-data-source

  mvn gf:create-jdbc-resource
- Run database

  mvn gf:start-database
- Create database authentication realm (We need one to perform authentication and authorization)

  mvn gf:create-auth-realm
- Combine all in one step

  You could combine all preparation in one maven call like this

  mvn gf:create-domain gf:start-domain gf:create-datasource gf:create-jdbc-resource gf:start-database gf:create-auth-realm
- Deploy

  mvn gf:deploy
- Check application

  In this case it is a web application with an html GUI so you need to open a web browser and type in an address line

  http://localhost:8080/online-store

Now you can open GlassFish admin console

To do this, type the following in your internet browser address line

http://localhost:5000/

You will be prompted to register the server, you can skip this step

On the left pane there is "Applications" folder icon, click it

On the main window you will see table with deployed modules/applications

| Name | Enabled | Engines | Action | |:------------|:---------|:--------------------|:-----------------------| | online-store | yes | ejb, jpa, web | Launch Redeploy Restart |

During your practice more modules/applications will be added to that list