Whirlwind tour of Spring, Hibernate and Maven

Jim Schmidt
jjs@javautil.org

October 10, 2010

Contents

Chapter 1

Introduction

You've just been assigned to a new project group that uses some of the most popular development libraries and methodologies.

This is a series of lessons designed to quickly give you an introduction to Maven, Spring and Hibernate.

1.1 Project Specs

Design a program to scan a disk drive and store the contents of an MP3 files into a database.

1.2 Steps

Download and install Maven.

Create your directories.
create new directory in Eclipse
Domain model
create project
in eclipse install maven plugin
enable dependence management (show screen shots)
We want to use the features available in version 1.5 of java
A lot of tabs, describe them
plugins
click add
need to add
Now add the following property
<junit.version>4.4</junit.version> because the build tool defaulted to a different version.
Working from the command line so that we can isolate out eclipse configuration problems.
maven from the command line
mvn eclipse:clean mvn eclipse:eclipse
configure in junit 4.4

1.3 installing library

We have decided to use jidlib now we must install it into our repository

Need to
run src/main/lib/install_jars_to_maven_repository.sh
Need to change to use java 1.6 but unable to configure through gui in eclipse need to edit pom.
Create a bean for the MP3 metadata

1.4 Design

Need a bean to hold the information from an MP3 file. TODO need to preserve tabs

```
package org.javautil.mp3;

import java.util.ArrayList;
```

```
public class MP3MetaData {
private static final String newline = System.getProperty("line.separator");

private int bitRate;

private String albumTitle1;

private String songTitle1;

private String authorComposer1;

private String leadArtist1;

private String genre1;

private String songComment1;

private String songLyric1;

private String track1;

private String yearReleased1;

private String albumTitle2;

private String songTitle2;

private String authorComposer2;

private String leadArtist2;

private String genre2;

private String songComment2;

private String songLyric2;

private String track2;

private String yearReleased2;

private ArrayList<String> messages = new ArrayList<String>();

private String fileName;

String preferred(String v1, String v2) {
String returnValue = null;
if (v1 == null) {
if (v2 != null) {
returnValue = v2;
}
} else {
if (v2 == null) {
returnValue = v1;
} else {
if (v2.startsWith(v1)) {
returnValue = v2;
} else {
// now what??
returnValue = v2;
```

```
}
}
}
return returnValue;
}

/**
 * @return the bitRate
 */
public int getBitRate() {
return bitRate;
}

/**
 * @param bitRate the bitRate to set
 */
public void setBitRate(int bitRate) {
this.bitRate = bitRate;
}

/**
 * @return the albumTitle1
 */
public String getAlbumTitle1() {
return albumTitle1;
}

/**
 * @param albumTitle1 the albumTitle1 to set
 */
public void setAlbumTitle1(String albumTitle1) {
this.albumTitle1 = albumTitle1;
}

/**
 * @return the songTitle1
 */
public String getSongTitle1() {
return songTitle1;
}

/**
 * @param songTitle1 the songTitle1 to set
 */
public void setSongTitle1(String songTitle1) {
this.songTitle1 = songTitle1;
}

/**
 * @return the authorComposer1
 */
public String getAuthorComposer1() {
return authorComposer1;
}

/**
 * @param authorComposer1 the authorComposer1 to set
 */
public void setAuthorComposer1(String authorComposer1) {
this.authorComposer1 = authorComposer1;
}
```

```
/**
 * @return the leadArtist1
 */
public String getLeadArtist1() {
return leadArtist1;
}

/**
 * @param leadArtist1 the leadArtist1 to set
 */
public void setLeadArtist1(String leadArtist1) {
this.leadArtist1 = leadArtist1;
}

/**
 * @return the genre1
 */
public String getGenre1() {
return genre1;
}

/**
 * @param genre1 the genre1 to set
 */
public void setGenre1(String genre1) {
this.genre1 = genre1;
}

/**
 * @return the songComment1
 */
public String getSongComment1() {
return songComment1;
}

/**
 * @param songComment1 the songComment1 to set
 */
public void setSongComment1(String songComment1) {
this.songComment1 = songComment1;
}

/**
 * @return the songLyric1
 */
public String getSongLyric1() {
return songLyric1;
}

/**
 * @param songLyric1 the songLyric1 to set
 */
public void setSongLyric1(String songLyric1) {
this.songLyric1 = songLyric1;
}

/**
 * @return the track1
 */
public String getTrack1() {
```

```
return track1;
}

/**
 * @param track1 the track1 to set
 */
public void setTrack1(String track1) {
this.track1 = track1;
}

/**
 * @return the yearReleased1
 */
public String getYearReleased1() {
return yearReleased1;
}

/**
 * @param yearReleased1 the yearReleased1 to set
 */
public void setYearReleased1(String yearReleased1) {
this.yearReleased1 = yearReleased1;
}

/**
 * @return the albumTitle2
 */
public String getAlbumTitle2() {
return albumTitle2;
}

/**
 * @param albumTitle2 the albumTitle2 to set
 */
public void setAlbumTitle2(String albumTitle2) {
this.albumTitle2 = albumTitle2;
}

/**
 * @return the songTitle2
 */
public String getSongTitle2() {
return songTitle2;
}

/**
 * @param songTitle2 the songTitle2 to set
 */
public void setSongTitle2(String songTitle2) {
this.songTitle2 = songTitle2;
}

/**
 * @return the authorComposer2
 */
public String getAuthorComposer2() {
return authorComposer2;
}

/**
 * @param authorComposer2 the authorComposer2 to set
```

```
 */
public void setAuthorComposer2(String authorComposer2) {
this.authorComposer2 = authorComposer2;
}

/**
 * @return the leadArtist2
 */
public String getLeadArtist2() {
return leadArtist2;
}

/**
 * @param leadArtist2 the leadArtist2 to set
 */
public void setLeadArtist2(String leadArtist2) {
this.leadArtist2 = leadArtist2;
}

/**
 * @return the genre2
 */
public String getGenre2() {
return genre2;
}

/**
 * @param genre2 the genre2 to set
 */
public void setGenre2(String genre2) {
this.genre2 = genre2;
}

/**
 * @return the songComment2
 */
public String getSongComment2() {
return songComment2;
}

/**
 * @param songComment2 the songComment2 to set
 */
public void setSongComment2(String songComment2) {
this.songComment2 = songComment2;
}

/**
 * @return the songLyric2
 */
public String getSongLyric2() {
return songLyric2;
}

/**
 * @param songLyric2 the songLyric2 to set
 */
public void setSongLyric2(String songLyric2) {
this.songLyric2 = songLyric2;
}
```

```
/**
 * @return the track2
 */
public String getTrack2() {
return track2;
}

/**
 * @param track2 the track2 to set
 */
public void setTrack2(String track2) {
this.track2 = track2;
}

/**
 * @return the yearReleased2
 */
public String getYearReleased2() {
return yearReleased2;
}

/**
 * @param yearReleased2 the yearReleased2 to set
 */
public void setYearReleased2(String yearReleased2) {
this.yearReleased2 = yearReleased2;
}

/**
 * @return the messages
 */
public ArrayList<String> getMessages() {
return messages;
}

/**
 * @param messages the messages to set
 */
public void setMessages(ArrayList<String> messages) {
this.messages = messages;
}

/**
 * @return the fileName
 */
public String getFileName() {
return fileName;
}

/**
 * @param fileName the fileName to set
 */
public void setFileName(String fileName) {
this.fileName = fileName;
}

/**
 * @return the newline
 */
public static String getNewline() {
return newline;
```

```
}


}
```

TODO want to test that it works correctly
test the class

1.5 Create the database objects

Talk about what comes first database objects or domain objects.

1.6 Creating an argument Beans

see javautil-commandline alter the pom.
Cut and paste this into the pom.

```
<dependency>
                    <groupId>org.javautil</groupId>
                    <artifactId>javautil-commandline</artifactId>
                    <version>${org.javautil.commandline.version}</version>
 </dependency>
```

Create the bean Create the properties file, we will show you how to use to process command line arguments later.
configure a log4j.xml in src/test/resources and make sure it gets deployed.
How much have we tested? Code coverage.
CSV's as test and expected results.

1.7 Now what

Now we have MP3MetaDataExtractorTest how can we configure this another way?

1.8 Inject over factories

public interface Mp3Persistence
How the persistence came into being is not a matter of concern of the application that wishes to store data.
Is it writing to a socket? A database? A file? Why do I care?
Every one of these implementations require different constuctors and or setter methods before than can do their work

1.9 write to flat file as

1.10 Implementations

Document Object Model Streaming XML CSV Excel Workbook HTML JASON Serialized Java Object JDBC Hibernate Service Call Dataset EBCDIC

1.11 The canonical representation, the domain model

Chapter 2

Logging

Chapter 3

Private Methods

Chapter 4

TODO

show cascading inherited beans.