Central Library Client Software

Tim Storer

1 Dependencies

- A Java Virtual Machine with version > 5.0

2 Setup

1. Unzip the file `psd3<team>.zip` attachment.
2. The database can be initialised with a randomly created collection of books and users by executing the setupdb.sh script. The script executes a SetupDB main class, that can also be invoked programatically. The script can be altered to vary the parameters for setting up the database (database configuration file, number of users, probability that a user is a member of staff and number of books. A book is randomly assigned a description from a set of approximately 9000 stored in config/descriptions.zip (a cached version of this data is stored in config/descriptions.obj to save time reading the XML formatted MARC data. You may need to alter the script to use the correct version of Java, and also to make it executable.
3. Add the psd3.jar class file to your Java project (e.g. in your Eclipse build path), as well as all the libraries in lib/. If you wish to initialise the database from your own code (this is not necessary) then the config/ directory should also be moved to your project's root directory.
4. Set the environment variable `L4J_CONF_FILE` for each of your target runs. This should point to the configuration file for log4j, e.g. the example file in config/l4j.properties. Two examples of setting an environment variable from the command line is shown in setupdb.sh.

3 Usage

The central library can be interacted with via the `CentralLibClient` interface, and its corresponding implementation `CentralLibClientImpl`. The client provides methods for searching for two data types: users and books, as well as as changing the status of a book to either `IN_LIBRARY` (default) or `IN_BRANCH`. The data types are represented by the client by two bean interfaces (Java types that conform to the getProperty/setProperty convention), that are shown below with their properties and types:

```
User(ID:Integer, surname:String, forename:String, isStaff:Boolean)

Book(ID:Integer, status:Status, description:BookDescription);
```

Examples of use are shown in the JUnit test case `CentralLibClientImplTest` in the tests package.
A BookDescription is a further bean type with the following properties:

```
BookDescription(ID:Integer, title:String, authors:String,
                published:String, publisher:String, ISBN:String)
```

User and Book items can either be searched for by specific ID:

```
User findUserByID(Integer ID)

Book findBookByID(Integer ID)
```

of by general search for beans matching properties:

```
List<User> findUsers(String query)

List<Book> findBooksByDescription(String query)
```

The query is a logical expression of field='value' pairs, for example:

```
"Authors='Sommerville,␣I%'␣␣AND␣Title='Software␣Engineering%'"
```

Finally, the status of a book can be set:

```
setStatus(ID:Integer, status:Status)
```

Note that although the properties of bean query results can be altered through setProperty methods, these changes will not be reflected in the database.

4 Trouble Shooting

4.1 Rebuilding the psd3.jar file.

Dependencies: Apache-Ant

The application jar file may need to be rebuilt, particularly if the source code is edited (e.g. to correct a fault - please email tws@dcs.gla.ac.uk with any corrections). To rebuild the jar type:

```
%>ant build-jar
```

in the application directory.

5 Reporting Bugs

1. Prepare a JUnit test that exhibits the bug, include documentation explaining what the test case does and the nature of the failure to be fixed. This test case will be added to the regression test suite for the application.
2. Email tws@dcs.gla.ac.uk with a copy of the test class.