

Homework 6

This is an individual homework assignment. You may receive help from other class mates, but you may NOT copy their work. Copied work will be considered plagiarism and dealt with according to the syllabus and university policy.

Objective: For this assignment, you will demonstrate that you can properly use several 3rd party libraries to connect to a database and create an industry standard compliant JSON web service using the data retrieved from the connection. Your solution is expected to conform to all Java coding conventions and expectations.

Description: Using 3rd party libraries and databases to create web services based upon industry standards is yet another requirement of contemporary object-oriented programming. Combining these new techniques with the standard Java object-oriented techniques allows us to create very complex client/server solutions using industry standards such as HTML, HTTP, and JSON. However, database storage formats are not directly compatible with programmatic data storage formats. We must know how to process and translate database query results into Java standard objects and serialized data formats suitable for use in a web service.

In this assignment, you will create the code using the .jar files provided by your professor to retrieve data from an Oracle database engine and start the Jetty container, define a Java based servlet that will present the results of a database query in JSON format to a web client, re-use a Java class (Textbook from HW5) modified with additional functions (e.g. .toJSON()) to store the database's multiple records into an appropriate data structure. The results of the processing of the web service will appear similar to that shown in Table 1 below.

Assignment:

- Extend the Textbook class file from Homework 5 to include a new .toJSON() method that will return the key/value pairs in standards compliant JSON syntax
- Create your Driver class file that starts the Jetty web container and a single servlet for the path "/textbooks" without quotations
 - This servlet will perform the tasks described in the following requirements
- Using the jetty-all-uber.jar & uca-jetty-wrapper.jar files along with the code exemplified in class, create a servlet with an appropriate name (e.g. TextbookReportJSON) that performs the following:
 - Using the ojdbc7.jar & uca-database-wrapper.jar files along with the code exemplified in class, query the database using the following SQL select query:
 - `SELECT * FROM TEXTBOOKS`
 - You may assume the metadata of the table based upon the table shown below in Figure 1
 - **Note:** Pages & Year are integer data types, all other fields are text
 - Using the results of the database query:
 - **Helpful tip:** the following requirements are unchanged from HW5, you will be able to reuse large amounts of code from HW5 to solve this assignment
 - Instantiate an instance of your Java class for each record in the database (there are 5)
 - Store each individual data element (there are 7) in an appropriate member in your Java class using the appropriate builder's setter method

- **Helpful tip:** Your class file should treat the data according to each field's correct data type, therefore, you'll need to read from the database using the `.getObject(XX)` method, convert that to a String data type using the `String.valueOf()` method, and then convert that to an integer data type using the `Integer.valueOf()` method, something similar to this:
 - `TextbookBuilder.setPages(Integer.valueOf(String.valueOf(rs.getObject(2))));`
 - **Note:** There is an easier mechanic for doing this task, extra credit may be awarded if you find and properly use the easier mechanic
- Store each instance in an appropriate data structure for later processing
 - After querying the database and storing the results into a data structure:
 - Iterate through the data structure outputting the details of each specific object in a proper JSON object
 - When invoked from a client web browser using an appropriate address, the output should be like what is shown below in Table 1
 - There are slightly different JSON compliant structures that you can create using the data stored in the database, the results must pass validation using <http://jsonlint.com>
 - Just because JSON passes validation does not mean it is correct, make sure you verify all of the JSON element (i.e. curly braces) and JSON array (i.e. square brackets) syntax requirements

➤ Upload your .JAVA code file(s) to the appropriate Blackboard assignment

```
{ "textbooks": [
{ "isbn": "0133594955", "pages": 806, "year": 2015, "edition": "8th",
"author": "John Lewis", "title": "Java Software Solutions: Foundations of
Program Design", "course": "MIS-3339"},
{ "isbn": "1439041284", "pages": 608, "year": 2014, "edition": "2nd",
"author": "Joan Casteel", "title": "Oracle 11g: SQL", "course": "MIS-4329"},
{ "isbn": "0133544613", "pages": 600, "year": 2015, "edition": "12th",
"author": "Jeffrey Hoffer", "title": "Modern Database Management",
"course": "MIS-3365"},
{ "isbn": "1305656318", "pages": 648, "year": 2016, "edition": "16th",
"author": "Deborah Morley", "title": "Understanding Computers: Today and
Tomorrow", "course": "MIS-3300"},
{ "isbn": "0134444329", "pages": 744, "year": 2017, "edition": "4th",
"author": "Tony Gaddis", "title": "Starting Out with Python", "course": "MIS-
3300"} ] }
```

Table 1 – Example Output

ISBN	PAGES	YEAR	EDITION	AUTHOR	TITLE	COURSE
0133594955	806	2015	8th	John Lewis	Java Software Solutions: Foundations of Program Design	MIS-3339
1439041284	608	2014	2nd	Joan Casteel	Oracle 11g: SQL	MIS-4329
0133544613	600	2015	12th	Jeffrey Hoffer	Modern Database Management	MIS-3365
1305656318	648	2016	16th	Deborah Morley	Understanding Computers: Today and Tomorrow	MIS-3300
0134444329	744	2017	4th	Tony Gaddis	Starting Out with Python	MIS-3300

Figure 1 – Oracle Table Contents & Metadata