DimensionData Technical Assignment:

Contents

[Scope: 1](#_Toc513126737)

[Released Features: 1](#_Toc513126738)

[Release Binary: 1](#_Toc513126739)

[Build and Deployment 2](#_Toc513126740)

[Build Pre-requisite: 2](#_Toc513126741)

[User instructions: 3](#_Toc513126742)

[Design/Architecture 5](#_Toc513126743)

# Scope:

Here is initial scope of an assignment:

Create a simple command line JAVA application with database access;  
Based on an input from the command line provide the following functionality;  
1) Provide help / usage info  
2) Add Server (id, name, description)  
3) Edit Server (name, description)  
4) Delete Server (id)  
5) Count Number of Servers  
6) List Servers  
Additional Requirements (nice to have, but not required);  
1) Ability to Add Server from XML (Senior Only)  
2) Filtering / Paging on List Servers (Senior Only)  
3) Test coverage  
4) Maven or Gradle Build  
5) Executable Jar

# Released Features:

Following items are delivered in this release:

1. Must Requirements

* Provide help / usage info –
* Add Server (id, name, description)
* Edit Server (name, description)
* Delete Server (id)
* Count Number of Servers
* List Servers
* Find Server by ID - additional
* Delete All servers - additional

1. Additional Requirements (nice to have, but not required);

* Maven Build
* Executable Jar

# Release Binary:

* ServerSpringExample-1.0-SNAPSHOT-jar-with-dependencies.jar

# Build and Deployment

## Build Pre-requisite:

* JDK 1.8.0.92 64 bit
* Apache Maven 3.3.1
* Intellij IDEA Community Edition 2018.1
* MySQL 8.0 Server/ Command line Client
* Windows 2010 64 bit machine
* MAVEN and JDK is installed on machine, and required env.varibales and path are set correctly for maven and JAVA

**Deployment Pre-requisite:**

* JRE 1.8.0.92 64 bit
* MySQL 8.0 Server/ Command line Client

**Build Instructions:**

* Copy sources to buid directory for e.g. c:\serverexample
* Run mvn clean install from src directory
  + For e.g. $c:\serverexample\mvn clean install
* Once mvn is executed successfully, following jar file will be put in to mvn repository and will be also available under target directory of src.
  + ServerSpringExample-1.0-SNAPSHOT.jar
  + ServerSpringExample-1.0-SNAPSHOT-jar-with-dependencies.jar

ServerSpringExample-1.0-SNAPSHOT-jar-with-dependencies.jar is Executable jar which contains all the required dependency jar files.

**Deployment Instructions:**

* Make sure mysql is running
* Open MySQL command line tool
* Make sure you have created following username and password to access MYSQL
  + Username: root
  + Password: password

Enter username and password when asked on MySQL command line.

* Execute following statements on the command line for one time only.
  + $create database servers;
  + $create table SERVER (ID CHAR(128) NOT NULL, NAME varchar(255) NOT NULL, PRIMARY KEY (ID)) ENGINE=InnoDB;
* Now execute jar using following command
  + $java –jar ServerSpringExample-1.0-SNAPSHOT-jar-with-dependencies.jar

# User instructions:

When jar is executed following will be displayed to the user:

------------------------------------------------------------

Here are your options

help: To display this message

countServers: To display the current number of servers present

addServer: To add new server with id and name

editServer: To change the name of a server identified by id (requires use to enter - the id and the new name)

deleteServer: To delete a server (requires user to enter the id of the server to delete)

listAll: To display details of all servers

findServer: To display a server details (Requires user to enter the id of the server

deleteAll: To delete all servers

------------------------------------------------------------

Enter your option:

1. To display help page, enter “help”

i.e. Enter your option: help

(2) To display number of servers, enter “countservers”

i.e. Enter your option: countservers

(3) To add new server, enter “addserver”

i.e. Enter your option: addserver

Enter Server ID:

1

Enter Server Name:

Server1

Here is one such output of executions of all operations

------------------------------------------------------------

Here are your options

help: To display this message

countServers: To display the current number of servers present

addServer: To add new server with id and name

editServer: To change the name of a server identified by id (requires use to enter - the id and the new name)

deleteServer: To delete a server (requires user to enter the id of the server to delete)

listAll: To display details of all servers

findServer: To display a server details (Requires user to enter the id of the server

deleteAll: To delete all servers

------------------------------------------------------------

Enter your option: help

------------------------------------------------------------

Here are your options

help: To display this message

countServers: To display the current number of servers present

addServer: To add new server with id and name

editServer: To change the name of a server identified by id (requires use to enter - the id and the new name)

deleteServer: To delete a server (requires user to enter the id of the server to delete)

listAll: To display details of all servers

findServer: To display a server details (Requires user to enter the id of the server

deleteAll: To delete all servers

------------------------------------------------------------

Enter your option: listall

No servers

Enter your option: countservers

Total servers: 0

Enter your option: addserver

Enter Server ID:

1

Enter Server Name:

server1

Enter your option: addserver

Enter Server ID:

1

Enter Server Name:

server2

Server already exists

Enter your option: addserver

Enter Server ID:

2

Enter Server Name:

server2

Enter your option: listall

Server ID: 1 Name: server1

Server ID: 2 Name: server2

Enter your option: countservers

Total servers: 2

Enter your option: findserver

Enter Server ID:

1

Server ID: 1 Name: server1

Enter your option: editserver

Enter Server ID:

2

Enter Server Name:

sever22222222

Enter your option: listall

Server ID: 1 Name: server1

Server ID: 2 Name: sever22222222

Enter your option: delete

No such options:

Enter your option: deleteserver

Enter Server ID:

2

Enter your option: listall

Server ID: 1 Name: server1

Enter your option: deleteall

Enter your option: listall

No servers

Enter your option: quit

# Design/Architecture

This application uses Spring/JDBC for implementation.

1. Sever Model to store server data: Server.java
2. Server DAO interface: ServerDAO.java
3. Server JDBC implementation using JDBCTemplate: ServerJDBCTemplate.java.
4. Spring Configuration: conf/Bean.xml
5. Main class: ServerManager.java

* JDBCTemplate is used instead of direct JDBC API as JDBC template eliminates the problems of JDBC API such as:
* Writing lot code before and after executing the query, such as creating connection, statement, closing resultset, connection etc.
* Writing code to perform exception handling code on the database logic.
* Writing code to handle transaction.

Improvement/Features to be added:

* Junit/code coverage
* Exception handling improvement
* Addition remaining features
  + Addition of server using XML
  + Filtering/paging of list of servers