

## **TicketServiceHomework**

A java based implementation providing different functionalities of a Ticket Booking Service namely seat discovery, temporary hold and final reservation of seats.

### **Implementation Details:**

It is developed with the help of spring framework and HSQLDB as the database. Primarily used spring dependency injections, spring JDBC for JdbcTemplate and database transaction Management. Maven is used to build the project.

It can be used entirely with the help of command line. It has two sets of service interfaces one for exposing actual business functionality (like check seat availability) and other for the maintenance perspective for the ease of testing having functionality to re-initialize the database to again provide a fresh instance for usage.

### **Development Tools:**

**IDE:** Eclipse Mars.2 Release (4.5.2)

To Use, code, test or build import the code into workspace.

Jar name with **TicketService-0.0.1-SNAPSHOT-jar-with-dependencies.jar** is a complete jar and is ready for testing.

**Main class name:** `com.walmart.lab.ticketService.Service`

### **List of artifacts in the zip file:**

1. **TicketService-0.0.1-SNAPSHOT-jar-with-dependencies.jar** [Runnable jar ready for testing]
2. **README.pdf** [Containing all the project specific details]
3. **hsqldb-2.3.1.jar** [To be used as portable database]
4. **Java-Docs** [Documentation about the project APIs]
5. **Java Code** [Containing the source code the JAVA implementation]

### **Assumption(s):**

1. In Seat Hold functionality holding the seat Level 1 (**Orchestra**) has been considered as the best possible option followed by other levels in the sequence i.e. **Balcony 2** as the least preferred option.
2. Each seat hold request has an expiry of 120000 milliseconds (120 secs or 2 mins).After which seat hold will be removed and reservation can't be entertained.
3. For every successful reservation, system will provide a confirmation code containing few fixed characters ("**Success**") and couple of random characters.
4. Already confirmed reservation will not work. It will give error.

### **Limitation(s):**

1. In the backend HSQLDB (Hyper SQL Database) is used for the data storage and retrieval's. It is a relational database management system written in Java and has some performance constraints in terms of database transaction, so the response time of all the application functionality will be dependent of it.

### **Testing:**

This solution has been tested on the Windows 7 operating system.

### **Usage Details:**

#### **Prerequisite:**

1. JAVA\_HOME should be set in system's environment variable.
2. Copy the provided hsqldb.jar in one of the folders in the system (e.g. c:\hsqldb\ ) and run the below command using command line to start the local database on the system.

```
java -cp <Jar_Location> org.hsqldb.server.Server
```

e.g. **java -cp c:\hsqldb\hsqldb-2.3.1.jar org.hsqldb.server.Server**

3. After the command below message will be displayed and the window can be left open.

```
java -cp c:\hsqldb\hsqldb-2.3.1.jar org.hsqldb.server.Server
[Thread[main,5,main]]: checkRunning(false) entered
[Thread[main,5,main]]: checkRunning(false) exited
Startup sequence initiated from main() method
Could not load properties from file
Using cli/default properties only
Initiating startup sequence...
Server socket opened successfully in 48 ms.
Database [index=0, id=0, db=file:test, alias=] opened sucessf
Startup sequence completed in 637 ms.
2016-05-04 02:31:49.070 HSQLDB server 2.3.0 is online on port
To close normally, connect and execute SHUTDOWN SQL
From command line, use [Ctrl]+[C] to abort abruptly
```

## Application Functionality:

1. Copy the **TicketService-0.0.1-SNAPSHOT-jar-with-dependencies.jar** in one of the folder (e.g. **C:\TicketService**).Open the cmd and change directory to the copied location.
2. Core functionality of the application can be utilized with the help of below commands:

- a. To find the number of seats available within the venue, optionally by seating level:

**java -jar <jar\_name> availableSeatCount -level <VENUE\_LEVELS | 1- 4 | optional>**

**Note:** For the very first time it will take significant time for setting up the database.

```
C:\TicketService>java -jar TicketService-0.0.1-SNAPSHOT-jar-with-dependencies.jar availableSeatCount -level 1
Work in Progress...
May 04, 2016 2:46:52 AM org.springframework.context.support.AbstractApplicationContext prepareRefresh
INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@65b3120a: startup date [Wed May 04 02:46:52 EDT 2016]; root of context hierarchy
May 04, 2016 2:46:52 AM org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
INFO: Loading XML bean definitions from class path resource [applicationContext.xml]
May 04, 2016 2:46:52 AM org.springframework.beans.factory.support.DefaultListableBeanFactory preInstantiateSingletons
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@21bcffb5: defining beans [org.springframework.aop.config.internalAutoProxyCreator,org.springframework.transaction.annotation.AnnotationTransactionAttributeSource#0,org.springframework.transaction.interceptor.TransactionInterceptor#0,org.springframework.transaction.config.internalTransactionAdvisor,transactionManager,dataSource,ticketDAO,ticketService,ticketServiceMaintenance,service1]; root of factory hierarchy
May 04, 2016 2:46:52 AM org.springframework.jdbc.datasource.DriverManagerDataSource setDriverClassName
INFO: Loaded JDBC driver: org.hsqldb.jdbcDriver
Please wait....Setting up the new database instance...
```

- b. Find and hold the best available seats on behalf of a customer, potentially limited to specific levels

```
java -jar <jar_name> holdSeat -count <NO_OF_SEATS | Mandatory> -minLevel  
<VENUE_LEVELS | 1- 4 | optional> -maxLevel <VENUE_LEVELS | 1- 4 | optional>  
-emailId <VALID_EMAIL_ID | Mandatory>
```

```
C:\TicketService>java -jar TicketService-0.0.1-SNAPSHOT-jar-with-dependencies.jar  
r holdSeat -count 10 -minLevel 1 -maxLevel 4 -emailId test@test.com  
Work in Progress...  
May 04, 2016 3:50:26 AM org.springframework.context.support.AbstractApplicationC  
ontext prepareRefresh  
INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationCont  
ext@65b3120a: startup date [Wed May 04 03:50:26 EDT 2016]; root of context hiera  
rchy  
May 04, 2016 3:50:26 AM org.springframework.beans.factory.xml.XmlBeanDefinitionR  
eader loadBeanDefinitions  
INFO: Loading XML bean definitions from class path resource [applicationContext.  
xml]  
May 04, 2016 3:50:27 AM org.springframework.beans.factory.support.DefaultListabl  
eBeanFactory preInstantiateSingletons  
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.  
DefaultListableBeanFactory@21bcffb5: defining beans [org.springframework.aop.con  
fig.internalAutoProxyCreator,org.springframework.transaction.annotation.Annotation  
onTransactionAttributeSource#0,org.springframework.transaction.interceptor.Trans  
actionInterceptor#0,org.springframework.transaction.config.internalTransactionAd  
visor,transactionManager,dataSource,ticketDAO,ticketService,ticketServiceMainten  
ance,servicel; root of factory hierarchy  
May 04, 2016 3:50:27 AM org.springframework.jdbc.datasource.DriverManagerDataSou  
rce setDriverClassName  
INFO: Loaded JDBC driver: org.hsqldb.jdbcDriver  
Inserting the initial data set into table...  
*****SeatHold Details*****  
SeatHoldID::1  
Venue Level::1      Seats No::1  
Venue Level::1      Seats No::2  
Venue Level::1      Seats No::3  
Venue Level::1      Seats No::4  
Venue Level::1      Seats No::5  
Venue Level::1      Seats No::6  
Venue Level::1      Seats No::7  
Venue Level::1      Seats No::8  
Venue Level::1      Seats No::9  
Venue Level::1      Seats No::10  
*****  
Ticket Service Ends...
```

- c. Reserve and commit a specific group of held seats for a customer

`java -jar <jar_name> bookSeat -bookingId <SEAT_HOLD_ID obtained in previous step |Mandatory> -emailId <EMAIL_ID used for seat hold | Mandatory >`

```
C:\TicketService>java -jar TicketService-0.0.1-SNAPSHOT-jar-with-dependencies.jar
r bookSeat bookingId 1 -emailId test@test.com
Work in Progress...
May 04, 2016 3:53:43 AM org.springframework.context.support.AbstractApplicationContext
ontext prepareRefresh
INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationCont
ext@65b3120a: startup date [Wed May 04 03:53:43 EDT 2016]; root of context hiera
rchy
May 04, 2016 3:53:43 AM org.springframework.beans.factory.xml.XmlBeanDefinitionR
eader loadBeanDefinitions
INFO: Loading XML bean definitions from class path resource [applicationContext.
xml]
May 04, 2016 3:53:43 AM org.springframework.beans.factory.support.DefaultListabl
eBeanFactory preInstantiateSingletons
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.
DefaultListableBeanFactory@21bcffb5: defining beans [org.springframework.aop.con
fig.internalAutoProxyCreator,org.springframework.transaction.annotation.Annotation
onTransactionAttributeSource#0,org.springframework.transaction.interceptor.Trans
actionInterceptor#0,org.springframework.transaction.config.internalTransactionAd
visor,transactionManager,dataSource,ticketDAO,ticketService,ticketServiceMainten
ance,service]; root of factory hierarchy
May 04, 2016 3:53:43 AM org.springframework.jdbc.datasource.DriverManagerDataSou
rce setDriverClassName
INFO: Loaded JDBC driver: org.hsqldb.jdbcDriver
Inserting the initial data set into table...
*****Reservation Details*****
Reservation Confirmation Code::Successyibhu
*****
Ticket Service Ends...
```