CDT Coding Exercise – Deployment Steps / Testing Guide

Developed a Spring-Boot application which exposes Rest Based Web Services with the help of jetty server and Maven Build and can be used to create , shuffle and delete a deck of cards . This Project also has extensive test coverage for all the layers separately , Testing also covers end-to-end flow . TDD was strictly followed while coding this project by writing the test cases first and then implementing the core logic .

This Document has the Pre-Requisites, Deployment steps, how to Switch between shuffling algorithms, Exposed Rest Web service URI’S and some important points from the project.

1. **Pre-Requisites :**
   1. Have maven Installed.
   2. Make sure no other service is running on Localhost:8080
   3. In-order to test the application have any REST Web-Service Testing Tool ( ex : REST Client , POSTMAN(provided by Google) )
2. **Deployment Steps :**
   1. Check out the code to local.
   2. Open Terminal and move to the folder path.
   3. Run : **mvn clean compile install**
      1. This will give you a “BUILD SUCCESS” message.
   4. Run the below command.

**mvn spring-boot:run -Drun.arguments="--cardShuffler.useComplexShuffling=false"**

* + - * Above command will start the server with jetty and using simple Shuffle logic .
      * Default shuffling logic is hand shuffling i.e. complex shuffling algorithm .

1. **Switching Between Shuffling algorithms :**
   1. Default Shuffling algorithm is to use Complex Shuffling ( Hand shuffling ) .
   2. If you want to turn off Complex Shuffling algorithm use the run-time parameters as given above .
2. **Testing the App :**
   1. Exposed REST Services :
3. GetDeckById (Request Method : GET ): <http://localhost:8080/getDeckById?deckId=1>
4. GetDeckByName ( Request Method : GET ) : <http://localhost:8080/getDeckByName?deckName=Kartheek>
5. GetAllDecks ( Request Method : GET ) : <http://localhost:8080/getAllDecks>
6. Create Deck ( Request Method : PUT) : <http://localhost:8080/createDeck?deckName=Kartheek>
7. Shuffle Deck ( Request Method : POST ) : <http://localhost:8080/shuffleDeck?shuffleCount=1&deckName=Kartheek>
   1. Shuffle Count allows you to shuffle number of times you want to , You can pass the number of shuffles required as parameter .
8. DeleteDeckById ( Request Method : DELETE ) : <http://localhost:8080/deleteDeck?deckId=1>
9. DeleteDeckByName ( Request Method : DELETE ) : <http://localhost:8080/deleteDeckByName?deckName=Kartheek>
10. DB LOG ( In Memory HSQLDB LOG ) can be obtained in the file mem.log in the project . This log is refreshed every time the server is restarted as the DB is refreshed .
11. All the configs in the project are Java spring-boot based .
12. Currently used HSQLDB but it can be upgraded to any durable database just by changing the Spring-Boot Config in POM.XML . Spring-Boot provides default properties for DB and we can override them in application.properties .
13. covered the complete Scope .

FYI ,

Also Added Code which can be used for writing the same functionality in other ways as given below .

1. By Saving cards :
   1. This would need having a Card table and a Metadata Table to map Cards with decks and also some space for saving the order .
   2. Code to be used to get it working this way is also added in below GIT Location .
      1. Class Names : CARD , CardDeckMetadata , CardRepositoryInt

Git Location : <https://github.com/yskartheek/OtherClasses-NikeProj>

1. Having JAVA Based Configuration without Spring-boot:
   1. We can do the same with Java based configuration without spring boot . This will need separate configuration classes .
   2. This would need DB , Session , Session Factory , Transaction Manager configurations to be mentioned in configuration classes .
   3. Code to be used to get it working this way is also added in below GIT Location .
      1. Class Names: Core Configuration, HSQLDBConfiguration.

Git Location : <https://github.com/yskartheek/OtherClasses-NikeProj>

**Rare Error because of running commands in wrong order :**

If you get the below error while running any of the two commands please follow the steps given after the error .

error in the forked process

[ERROR] java.lang.UnsupportedClassVersionError: com/interview/serviceTest/Comple

xShufflerStrategyTest : Unsupported major.minor version 52.0

Run the below items again to fix it :

* 1. **mvn clean compile install**
  2. **mvn spring-boot:run -Drun.arguments="--cardShuffler.useComplexShuffling=false"**