Address Book - PWC Australia Code Challenge Project

The intent of this project is to develop an address book that allows a user to store (between successive runs of the program) the name and phone numbers of their friends, with the following functionality:

1. To be able to display the list of friends and their corresponding phone numbers sorted by their name.
2. Given another address book that may or may not contain the same friends, display the list of friends that are unique to each address book (the union of all the relative complements).

What is included?

The project includes the source code, `book.json`, behaviour driven tests based on the specifications provided and build scripts for Maven. It also ships with a compiled version of the project making it makes it easy execute the application from terminal. The compiled version of the project resided in the `dist` directory and can be executed from command-line/terminal

Maven will automatically download all required dependencies, compile the project, and finally run the tests specs.

Prerequisites

- JDK 7 or higher
- Maven use `mvnw` wrapper

Executing the application

Navigate into the dist folder/directory and run the following commands

1. To save a contact and persist to database run this command:

```
java -jar dist/code-challenge-0.0.1-SNAPSHOT.jar --store Joshua 0479109809
```

2. To unique/union two address books run this command:

```
java -jar dist/code-challenge-0.0.1-SNAPSHOT.jar --union "
{Jenny:098765667,Asha:908654}"
```

Executing the above command will produce the following output

```
$ java -jar dist/code-challenge-0.0.1-SNAPSHOT.jar --store Joshua
0479109809
2019-03-15 21:23:14,358 INFO  Running address book app in store mode
2019-03-15 21:23:14,416 INFO  Added new contact Contact{name='Joshua',
phoneNumber='0479109809'}
2019-03-15 21:23:14,416 INFO  Displaying contact numbers ordered by name.
Joshua 0479109809
```

and the union between two books will produce

```
$ java -jar dist/code-challenge-0.0.1-SNAPSHOT.jar --union "
{Jenny:098765667,Asha:908654}"
2019-03-15 22:00:28,783 INFO  Running address book app in union mode
2019-03-15 22:00:28,842 INFO  Displaying unique entries
Book 1/Book 2: {Adam=04897654367, Asha=908654, Joshua=0479109809,
Zoe=09864567}
```

Detailed instruction on how to execute and use this application can be found in User Guide Document.

Building the API docs

Execute the command below to build API javadoc from the source code.

```
mvn javadoc:javadoc
```

The result is put in `/target/site/apidocs`

Building the source code without tests

```
mvn clean install -Dmaven.test.skip=true
```

This will compile, package and install the project. A JAR file called `code-challenge-0.0.1-SNAPSHOT.jar` is generated at the end of this phase and the result is copied to `target` directory.

Building and copying the dependencies

Run this command below to build and copy the dependencies to project `lib` directory.

```
mvn clean install -Dmaven.test.skip dependency:copy-dependencies
```

Running the tests.

The tests are written in groovy using Junit framework. The source code for the tests are located in `src/test/java`. I wrote the tests to be simple and readable. Anyone looking at the tests would find it easy to figure out what's going on.

```
mvn clean install
```

This will run all the tests in `src/test/java` directory and you will see an output like this below.

Test logs

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running unit.RegexUtilsTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.029 sec
Running unit.AddressBookUnionServiceTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.028 sec
Running unit.AddressBookServiceTest
2019-03-15 05:11:53,565 INFO  Added new contact Contact{name='Simeon', phoneNumber='0479109809'}
2019-03-15 05:11:53,567 INFO  Updated existing contact Contact{name='Simeon', phoneNumber='0479109809'}
2019-03-15 05:11:53,568 INFO  Added new contact Contact{name='Mark', phoneNumber='0479109809'}
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.121 sec
Running unit.InputValidationParserTest
2019-03-15 05:11:53,570 ERROR {joshua:,zoe:87655456787} is not a valid JSON formatted string.
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.002 sec

Results :

Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
```