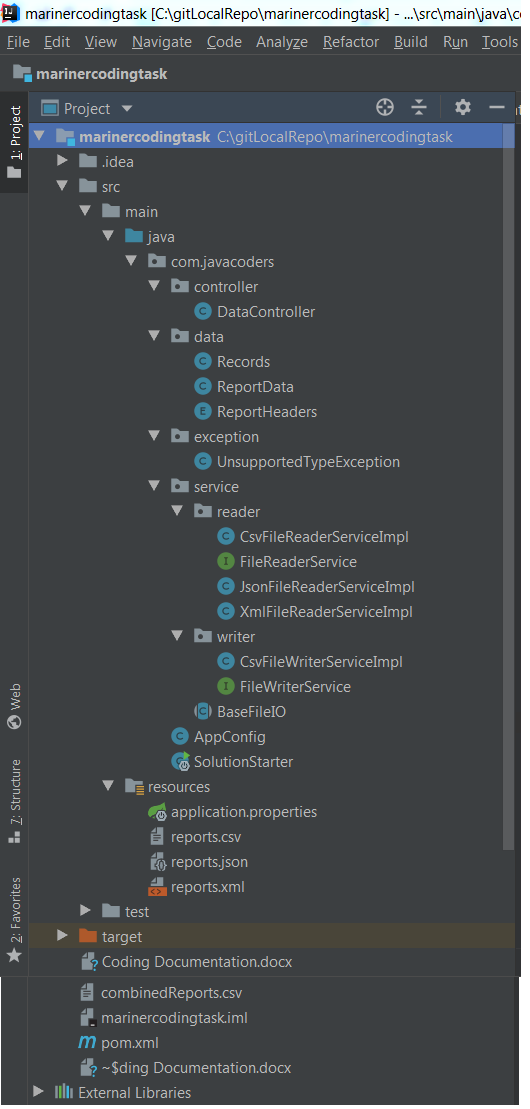
The document highlights following:

* Sample output
* Directory structure
* How to run the program
* Tools/libraries used
* Explanation on why to use them

**SAMPLE OUTPUT:**



**DIRECTORY STRUCTURE:**

****

**TOOLS/LIBRARIES USED:**

* Java 8
* Spring Framework/SpringBoot
* Maven-3.3.9
* Apache Commons CSV library to read/write csv file
* FasterXML/jackson library to read json file
* JAXB library to read xml file
* SLF4J library for logging
* IntelliJ IDE for Development

**How to run the program**:

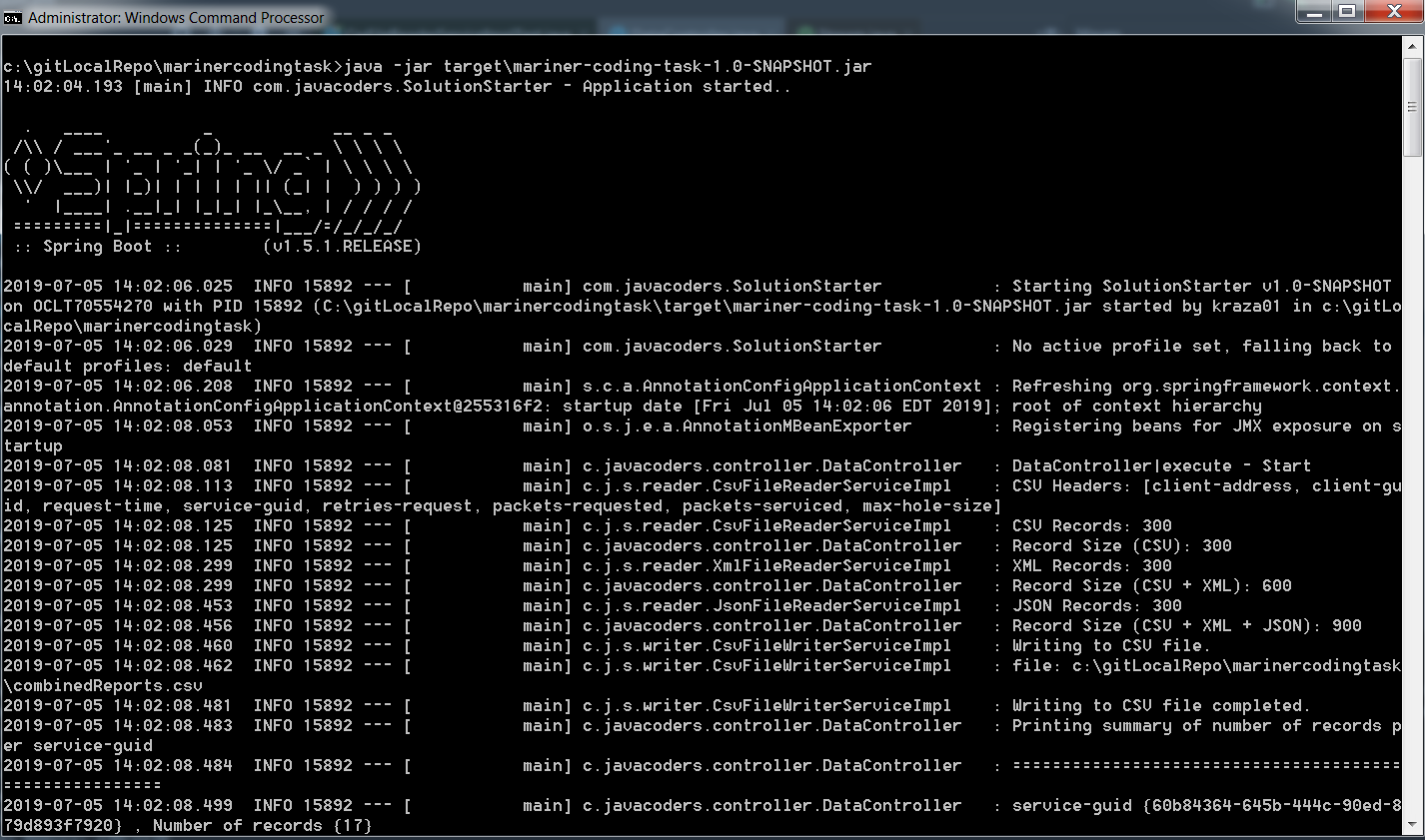
Following can be run from command line under ‘marinercodingtask’ project:

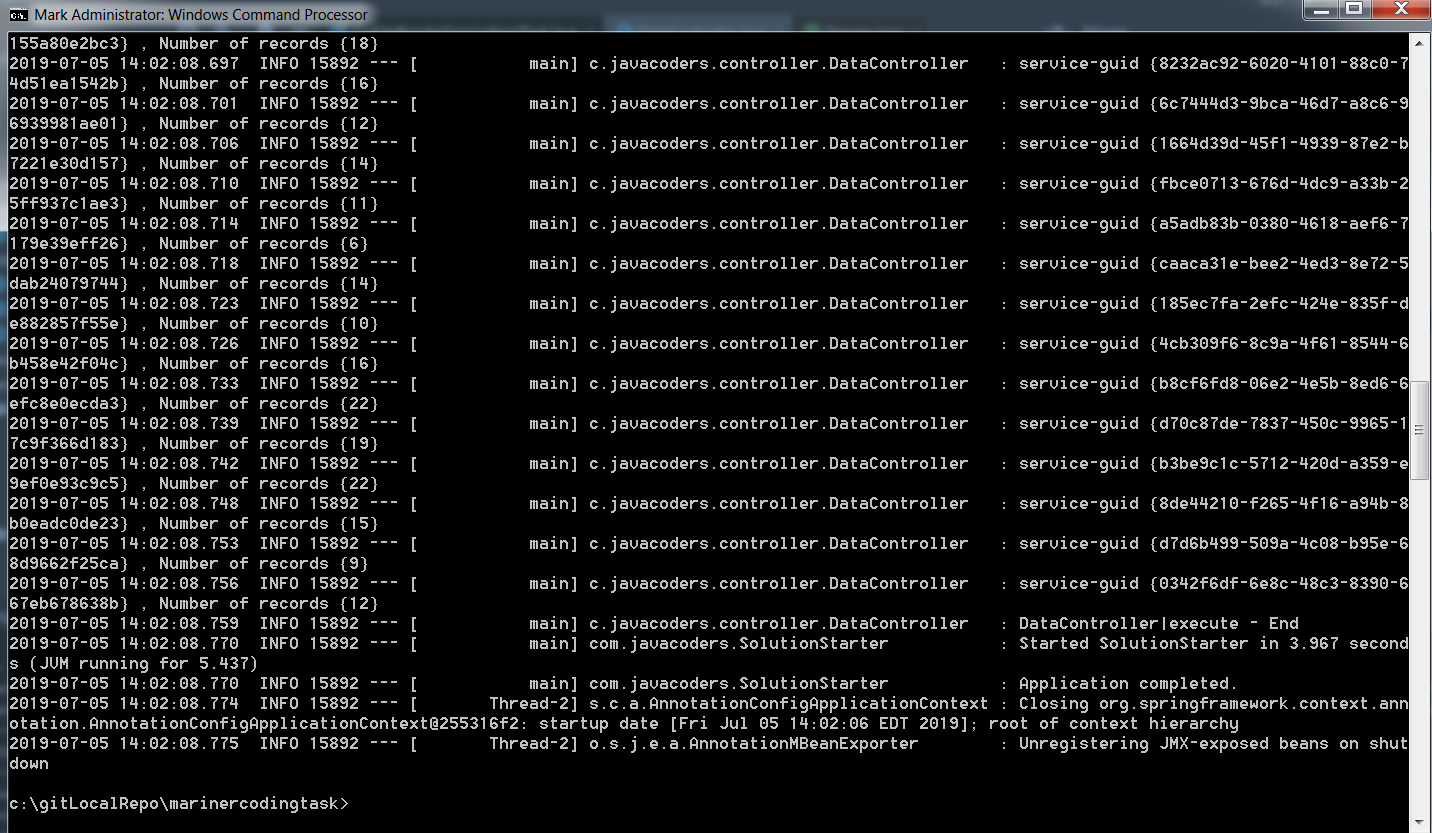
**java -jar target\mariner-coding-task-1.0-SNAPSHOT.jar**

OR

**The project can be imported in an IDE (Eclipse, IntelliJ) and run SolutionStarter.java as Springboot application**

**Output:**





**EXPLANATION ON WHY TO USE THE TOOLS/LIBRARIES:**

In order to complete the coding exercise latest technologies and open source libraries are used such as Java 8, SpringBoot, Maven, etc. The benefit of using Java 8 platform is to avail enrich and extensive features provided by the language along with the benefit of integrating and using enormous open source libraries as Java itself is opensource.

SpringBoot is used here mainly to have extensive features provided by Spring framework instead of writing boiler plate code. More importantly, Springboot reduces development efforts and provides production ready application that can be deployed in an application container directly as there is tomcat container embedded within Springboot.

Maven is used for this exercise for better dependency management as well as build and packaging the application.

Open source libraries such as Apache Commons CSV library, FasterXML/Jackson library, JAXB library are used to read/write csv, xml, json files without writing the low level code required to manipulate with different types of files. These libraries are very popular in Java world and provides benefits such as faster, efficient, improved performance with an additional benefit of using fewer lines of code for the developers so that developers can focus more on business logic and do not worry about how to read/write or manipulate with these types of files. For example, using JAXB library, only few lines of code were required to read data from an xml file:

jaxbContext = JAXBContext.*newInstance*(Records.class);  
Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();  
records = (Records) unmarshaller.unmarshal(getFileResource(*XML\_FILE\_PATH*));

SLF4J is used for logging output to console instead of traditional System.out.println(). It is easy to use in any application and supports all the main logging frameworks and works well with modern frameworks including Spring. In addition, it provides an abstraction of the underlying logging framework which makes it independent of the logging framework being used. In order to use it, all you need is to add relevant dependency in pom.xml and add following line in Java class:

private static final Logger *logger* = LoggerFactory.*getLogger*(CsvFileReaderServiceImpl.class);

Finally IntelliJ IDE from JetBrains is used as it provides rich/extensive features (autocomplete, refactoring, auto build, debugging capabilities, etc. to name a few) and plugins needed for development and thus makes life easier for developers.