Internship Week (1) Day 2

Tripillar Pvt Ltd Internship

Documentation Report for Maven File Handling Project

Submitted By:

Tanusree Paul

1NH23MC158

TABLE OF CONTENTS

Project Overview:

This project takes care of both text and Excel file handling using a Java Maven project. Basic file operations related to reading and writing for the text files are taken care of in this project, along with reading and writing .xlsx files using Apache POI to handle operations related to the Excel file. The project was set up using Maven to take care of dependency management for easy inclusion of libraries when handling Excel file manipulation.

1. Key Learning:

1.1 Maven Project Setup:

Learned how to create a Maven project completely new in IntelliJ IDEA.

Understood how Maven simplifies dependency management directly in the pom.xml file by adding dependencies instead of downloading JAR files whose download has to be requested manually.

Any project's structure with Maven follows the standard conventions (src/main/java, src/test/java, etc.).

1.2 Working with Packages:

Wrote all the Java classes and placed them in packages (com.tripillar.filehandling.text and com.tripillar.filehandling.excel) in order to keep proper coding structure and separation of concerns.

1.3 Text File Handling:

Used learning of reading and writing of text files in Java using java.io package.

FileWriter and BufferedReader has been successfully applied for writing and reading text files.

I understood how to work with exceptions related to file I/O by using a try-with-resources for closing the file resources.

1.4 Excel File Handling with Apache POI:

I Have obtained the practice of working on files in .xlsx format for Excel in Java, using the Apache POI library.

Used classes Workbook, Sheet, and Row for writing data into and reading data from Excel files.

Understanding the nature of Excel files and manipulating rows and columns programmatically.

1.5 Managing Maven Dependencies:

Added and configured the external libraries using Maven (Apache POI) by adding necessary dependencies in pom.xml.

Maven, as such, automatically downloads and manages the required JAR files from the sources that are preferably Maven Central, which is easier to work with large libraries.

2. Challenges Faced and How They Were Resolved:

2.1 Setting Up Maven Dependencies:

Initially, there was some confusion about how to add the Apache POI library for handling Excel files.

With a little research, I identified the Maven dependencies for Apache POI. Added those to the pom.xml. Then refreshed the Maven project to download the JAR files that Maven requires.

2.2 Working with Apache POI:

At first, it was hard to understand how to use Apache POI for working with Excel file handling because of the complexity of structures in Excel data, as well as variations in formats (.xls - HSSF, .xlsx - XSSF).

I have tried to remain focused on the use of class XSSFWorkbook in handling modern files .xlsx and have learned basic operations-how to create workbooks, sheets, rows, and cells-by following instructions in the Apache POI documentation.

2.3 Exception Handling in File I/O:

Handling the exception during file reading/writing operations was a little tricky since I was dealing with both text files and Excel files.

I have used try-with-resources for file operations, so the resources like FileWriter, BufferedReader and FileInputStream are closed automatically.

3. Conclusion:

This project taught me how to use Maven to manage dependency, handle various file formats in Java, and structure code properly by using packages. Though part of the text file was easy using the in-built java.io package, it was much easier for the project to make use of the libraries like Apache POI for handling Excel files with the aid of Maven.