

University Parking System: Serialization for the Parking System Application- Client/Server

for

Master of Science

Information Technology

Kassandra Vega Lucero

University of Denver College of Professional Studies

May 26, 2025

Faculty: Nathan Braun, MBA

Dean: Michael J. McGuire, MLS

Table of Contents

Background	3
Implementation Challenges	3
Existing Implementation	6
Unit Tests.....	6
Conclusion.....	7
References.....	8

Background

Over the last several weeks, I have been developing a Parking System for the University. This system has made use of classes to register students and other community members in order for them to use the parking lots and be charged based on the parking lot's individual charge guidelines. The focus for this week is to incorporate a more modern approach to serialization. The current system uses a basic form of serialization which sends the data line by line. Modern serialization protocol includes using systems such as JSON, Protobufs, or Java serialization. Below is a recap of my implementation.

Implementation Challenges

Implementing any form of serialization for the parking system has been a great challenge. Upon looking for the line-by-line serialization, I could not find how my system was implementing it. When I looked up examples of line-by-line serialization to use as a reference of what I should be looking for, they indicated it is usually in the form of writing into a file (GeeksforGeeks 2025). I do recall having done this in a prior project and used my notes to reference what was done. I recall having notes due to issues with getting the system to compile and run the code as intended. The simple serialization was nowhere to be found. I attempted to create the simple serialization and had issues, so I figured it might be in my best interest to try and get the modern implementation going instead since it would need to be changed regardless.

As I began attempting to implement the modern serialization, I encountered a roadblock due to an earlier discovery in a prior assignment. A couple of weeks ago when I was having issues with my code compiling, I learned that my project did not have a pom.xml file. This file is

linked to having Maven in the system. While researching how to read and write JSON files, I saw dependencies would need to be added to the pom.xml file (GeeksforGeeks 2024). Since the pom.xml file was missing from my system, I needed to find a way to get the file back in the system. I found an article that would create a Maven project in Eclipse from scratch (GeeksforGeeks 2022). When I first created a parking system for part I of this course, I had issues creating a Maven project and was instructed to create it without it. After several instances of trial and error with downloading and installing, I was able to convert my existing project into a Maven Project which gave me access to the pom.xml file. When running commands for Maven I got the messages seen in Figure 1. Upon doing light research, I ran into an article (which I can no longer locate) indicating that this would only cause issues for future implementations using newer versions of Java. The next step was adding dependencies. However, when trying to run the commands indicated in the assignment, I get the errors depicted in Figure 2.

```

en-3.9.9/lib/jansi-2.4.1.jar)
WARNING: Use --enable-native-access=ALL-UNNAMED to avoid a warning for callers in this module
WARNING: Restricted methods will be blocked in a future release unless native access is enabled

Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfc97d260186937)
Maven home: C:\Users\kassi\OneDrive\Desktop\apache-maven-3.9.9
Java version: 24.0.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-24
Default locale: en_US, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"

C:\Users\kassi>mvn clean install
WARNING: A restricted method in java.lang.System has been called
WARNING: java.lang.System::load has been called by org.fusesource.jansi.internal.JansiLoader in an unnamed module (file:/C:/Users/kassi/OneDrive/Desktop/apache-mav
en-3.9.9/lib/jansi-2.4.1.jar)
WARNING: Use --enable-native-access=ALL-UNNAMED to avoid a warning for callers in this module
WARNING: Restricted methods will be blocked in a future release unless native access is enabled

WARNING: A terminally deprecated method in sun.misc.Unsafe has been called
WARNING: sun.misc.Unsafe::objectFieldOffset has been called by com.google.common.util.concurrent.AbstractFuture$UnsafeAtomicHelper (file:/C:/Users/kassi/OneDrive/D
esktop/apache-maven-3.9.9/lib/guava-33.2.1-jre.jar)
WARNING: Please consider reporting this to the maintainers of class com.google.common.util.concurrent.AbstractFuture$UnsafeAtomicHelper
WARNING: sun.misc.Unsafe::objectFieldOffset will be removed in a future release
[INFO] Scanning for projects...
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 0.249 s
[INFO] Finished at: 2025-05-26T18:03:56-06:00
[INFO] -----
[ERROR] The goal you specified requires a project to execute but there is no POM in this directory (C:\Users\kassi). Please verify you invoked Maven from the corre
ct directory. -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://wiki.apache.org/confluence/display/MAVEN/MissingProjectException
java -cp bin ict4305 parking_server.Server
Error: Could not find or load main class ict4305.parking_server.Server
Caused by: java.lang.ClassNotFoundException: ict4305.parking_server.Server

```

Figure 1. Messages seen when running commands for Maven.

```
C:\Users\kassi>java -cp bin ict4305.parking.server.Server
Error: Could not find or load main class ict4305.parking.server.Server
Caused by: java.lang.ClassNotFoundException: ict4305.parking.server.Server

C:\Users\kassi>javac -d bin src/ict4305/parking/server/Server.java
error: file not found: src\ict4305\parking\server\Server.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\kassi>java -cp bin ict4305.parking.server.Server
Error: Could not find or load main class ict4305.parking.server.Server
Caused by: java.lang.ClassNotFoundException: ict4305.parking.server.Server

C:\Users\kassi>javac -d bin src/ict4305/parking/server/*.java
error: Invalid filename: src/ict4305/parking/server/*.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\kassi>echo %CLASSPATH%
%CLASSPATH%

C:\Users\kassi>java -cp bin ict4305.parking.server.Server
Error: Could not find or load main class ict4305.parking.server.Server
Caused by: java.lang.ClassNotFoundException: ict4305.parking.server.Server

C:\Users\kassi>
```

Figure 2. Command results when running prompts for verifying the server worked.

Due to the issues I was encountering using JSON, I had my friend Christian Palma that is visiting during summer break guide me through another possible implementation using Gson. While going through and following his guidance for importing and implementing Gson, I ran into issues importing from com.google.gson. The work I found for this was downloading a jar file which was then added to a user library, but it had to be the binary version otherwise the results were reduced. When trying to run the command server code, I also got inconclusive results using the Gson approach.

Existing Implementation

My existing implementation is able to provide some successful unit test. With some guidance of Palma, I was able to get a couple of tests up and running. I do want to preface by indicating my understanding of the server and server client are quite cloudy. I added print statements to attempt to debug some of the issues, but I can't seem to locate where the issue is coming from. While the assignment indicated I should use command prompt, I used code to simulate customer input. Palma walked me through some of the basics, but there are still some questions I have along with the prior issues indicated.

When attempting to locate why adding the License parameter was not being added, I added print statements which show they are being added to the JSON string, but they aren't being converted back (Figure 3). This is the issue with the unsuccessful test in `ParkingRequestTest`.

```
Raw JSON Input: {"commandName":"CAR","parameters":{"license":"R0B4C0","customer":"CUST1"}}
Command: CAR
License: null
Customer: CUST1
```

Figure 3. Null License when converting JSON string to object.

Unit Tests

Creating tests was complicated due to my confusion, but I have included the ones I have attempted (including the unsuccessful tests).

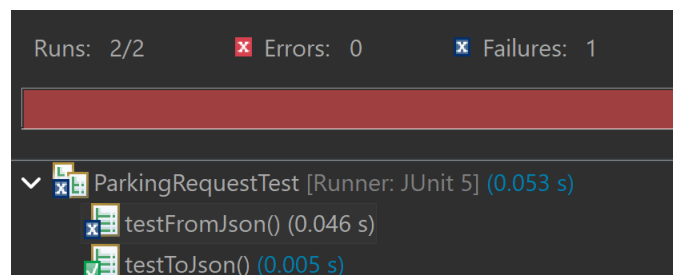


Figure 4. Test for Parking Request including failed test.

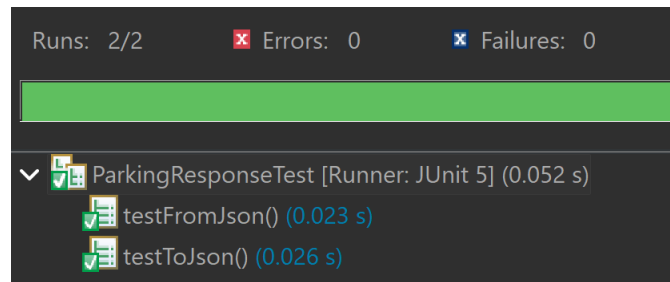


Figure 5. Successful tests for ParkingResponseTest.

Conclusion

After this week, I feel more lost than I have before. I can grasp the basics that I have covered in other class projects, but I am struggling with the command prompts as well as general organization and implementation. Topics have begun blurring together through the troubleshooting and are unsure on how I can separate them again.

References

Baeldung. 2024. "Jackson vs Gson." Baeldung. Updated January 8, 2024.

<https://www.baeldung.com/jackson-vs-gson>.

Catch Error. 2021. "how to import path in java eclipse || how to create json in java using eclipse || import path." YouTube Video. Published August 21, 2021.

https://www.youtube.com/watch?v=pR2SsuE_fuw.

GeeksforGeeks. 2022. "How to Create a Maven Project in Eclipse IDE?" GeeksforGeeks. Updated March 02, 2022. <https://www.geeksforgeeks.org/how-to-create-a-maven-project-in-eclipse-ide/>.

GeeksforGeeks 2024. "How to Read and Write JSON Files in Java?" GeeksforGeeks. Updated October 10, 2024. <https://www.geeksforgeeks.org/how-to-read-and-write-json-files-in-java/>.

GeeksforGeeks. 2025. "Serialization and Deserialization in Java with Example." GeeksforGeeks. Update January 04, 2025. <https://www.geeksforgeeks.org/serialization-in-java/>.

Kass. 2017. "Importing GSON into Eclipse." Medium. Published December 16, 2017. <https://medium.com/programmers-blockchain/importing-gson-into-eclipse-ec8cf678ad52>.

Palma, Christian. Question and Answer Session. May 25, 2025.

Professor Saad. 2017. "Java JSON Kar File Download." YouTube. Published April 5, 2017. <https://www.youtube.com/watch?v=joAxQtSaErE>.

Ram Technical Help. 2024. "How to fix Downloading from external resources is disabled error in eclipse." YouTube Video. Published May 15, 2024.

https://www.youtube.com/watch?v=pR_jO5X-GYw.

Stephens, Garrett. 2023. "Port 8080, Why do I see this a lot? What does it mean? What is the History behind it?" Medium. Published January 1, 2023.

<https://medium.com/@garstep/can-you-explain-a-few-of-the-internet-assigned-numbers-authority-list-and-in-doing-so-explain-what-cc73cc257799>.

Stram. 2016. "How do I download the Gson library?" Stack overflow. Answered June 22, 2016.

<https://stackoverflow.com/questions/37975605/how-do-i-download-the-gson-library>.