Yohannes Ayele

Assignment 11.2

Date 5/18/25

**Introduction**

For this week's assignment, I researched three sources about JSON API. Managing JSON data effectively has become essential in the fast-paced world of Java programming. A reliable and proven method for parsing, generating, processing, and querying JSON data within Java applications is the Java API for JSON Processing (JSON-P), which was defined in JSR 353. This article examines JSON-P's characteristics, background, and real-world uses of JSON-P, which are explored in this paper along with information on its object model and streaming APIs. It also provides instructions on how to get the JAR files required for implementation.

**Historical Background**

In 2013, Java EE 7 introduced the Java API for JSON Processing (JSON-P) to provide a standardized method for handling JSON data in Java applications. Before JSON-P, developers commonly used third-party libraries like Jackson and Gson for JSON processing. A standardized and portable API for parsing, creating, and querying JSON data across several Java platforms was made available by the addition of JSON-P to the Java EE specification. The goal of this specification was to facilitate compatibility within the Java environment and make handling JSON easier.

The core features of JSON processing in Java include two main APIs: the Object Model API and the Streaming API. The Object Model API represents JSON data as a tree-like structure in memory, similar to the Document Object Model (DOM) used in XML processing. With important classes like JsonObject, JsonArray, and JsonValue, this modle is perfect for situations that call for random access and modification of the complete JSON structure. In contrast, the Streaming API processes JSON data in a forward-only, read/write manner using an event-driven approach. It is well-suited for handling large JSON documents or data streams where performance and low memory usage are essential, utilizing key interfaces like JsonParser and JsonGenerator.

JSON-P is a natural fit for enterprise applications due to its seamless integration with Java EE technologies. Because of its consistent methodology, developers who are already familiar with Java EE APIs would experience less of a learning curve and improved portability across a variety of Java contexts.

To implement JSON-P in your Java application, you need to include the appropriate JAR files. The official JAR files for JSON API can be downloaded from the Oracle website: https://download.oracle.com/otndocs/jcp/json-1\_0-fr-eval-spec/index.html

Before downloading the Jar file from Oracle site, you must accept the software license Agreement and download Jar file.

**Conclusion**

Handling JSON data in Java applications is made standardized and effective via the Java API for JSON Processing (JSON-P). Its object model and streaming APIs offer performance and flexibility for a variety of use scenarios. For consistent and portable JSON processing across different Java platforms, incorporate JSON-P into your Java projects.

**Sources**

* Oracle. (2013). Java API for JSON Processing. <https://www.oracle.com/technical-resources/articles/java/json.html>
* Oracle. (n.d.). JSON Processing (Release 7). <https://docs.oracle.com/javaee/7/tutorial/jsonp.htm>
* Oracle. (n.d.). JSR-000353 Java API for JSON Processing 1.0 Final Release for Evaluation. https://download.oracle.com/otndocs/jcp/json-1\_0-fr-eval-spec/index.html