Zac Baker
M11 Programming Assignment
12/21/2025

# Jackson

JavaScript Object Notation (JSON) has become one of the most common structured data sets that various Java applications rely on. The readability for both computers and humans makes it easy to understand the data that is being handled and helps understand what is being parsed. Due to the key-value pair structure, parsing is effective and efficient, which is only increased with frameworks like Jackson, which focus on JSON processing. Jackson is a high-performance JSON processing library that started development in 2007 by Tatu Saloranta and was officially released in 2009. The goal was to create a processing library that was fast, flexible, and lightweight, which has helped it become the standard for default JSON processing. It has evolved into an open-source JSON processing library, which is developed and maintained by FasterXML. It is commonly used in enterprise applications, REST APIs, and frameworks such as Spring Boot, where it is the default JSON processor.

Jackson revolves around three primary modules: streaming, tree model, and databind. The streaming module is the lowest-level and most efficient implementation as it uses the least amount of memory. Since it reads the entire file as tokens, this reduces the overall memory, which handles large JSON files well. The tree model enables the representation of JSON data as a tree of JSON objects without binding it to a Java class. This approach is ideal when it comes to a dynamic or unknown JSON structure since it is more flexible. However, due to the increased complexity, it also required more memory than the streaming module. The data binding module is the most used approach, which automatically converts JSON to Plain Old Java Objects (POJOs) and vice versa. From here, it can be further customized between the default, annotation-based, or custom serialization and deserialization. The default method relies on JavaBean default constructors, getters, and setters to automatically map the JSON fields. Ideal if further customization is not needed and the default options are already sufficient. The annotation-based method allows for Jackson-specific annotations for mapping instructions beyond the standard defaults. This becomes optimal when the default method does not provide the necessary mapping, but custom operations are not needed. It is essentially a middle ground between the two. The custom method is the most complex option, which utilizes custom logic for mapping when neither option is sufficient. Increasing the complexity also increases the memory needed to process the information. However, complexity comes at the cost of simplifying parsing logic by implementing the streaming method similarly to the tree model.

When it comes to JSON processing, Jackson stands out as a sort of benchmark for expected standards. The performance, flexibility, and lightweight nature of Jackson help elevate it to become an efficient framework to increase JSON processing. Whether it is low-level streaming or high-level object mapping, Jackson provides methods for developers to perform various processing needs. The various methods and additional plugins only increase the usefulness that the Jackson framework provides. Since its creation and open-source development, the development community has adopted it as a reliable and powerful tool for processing JSON in Java.

**JAR File Web Address**

https://repo1.maven.org/maven2/com/fasterxml/jackson/core/

**Sources**


Evans, B. (2023, January 2). *Efficient JSON serialization with Jackson and Java*. javamagazine. https://blogs.oracle.com/javamagazine/java-json-serialization-jackson/

Gilliard, M. (2025, June 13). *Three ways to use jackson for JSON in Java*. Twilio. https://www.twilio.com/en-us/blog/developers/tutorials/building-blocks/java-json-with-jackson#Example-data-and-questions

*Jackson Library in Java. Jackson is a popular java JSON library... | by Ramesh Fadatare | Medium*. Medium. (2025, March 10). https://rameshfadatare.medium.com/jackson-library-in-java-eef28ab9fb40

Kumar, P. (2022, August 3). *Jackson Json Java parser api example tutorial*. DigitalOcean. https://www.digitalocean.com/community/tutorials/jackson-json-java-parser-api-example-tutorial

*Oracle Goldengate Licensing Information*. Oracle Help Center. (2025, October 31). https://docs.oracle.com/en/middleware/goldengate/core/23/ogglc/fasterxml-llc.html