Kyle Marlia-Conner

05/18/2025

Assignment 11

Jackson’s JSON

In Java, working with JSON (JavaScript Object Notation) is common but essential for effectively communicating with web services, managing configuration data, and exchanging structured information. JSON’s readability, lightweight nature, and compatibility with various programming environments solidify its status as the standard for data interchange. Among the many available libraries, the Jackson API stands out as one of Java's most widely adopted JSON libraries, known for its exceptional performance, flexibility, and powerful features.

“Jackson has been known as ‘the Java JSON library’ or ‘the best JSON parser for Java’. Or simply as ‘JSON for Java’” (FasterXML, n.d.). The Jackson JSON processor, often referred to simply as Jackson, was initially developed by [Tatu Saloranta in 2007](https://docs.oracle.com/cd/F10086_01/BIGLI/GUID-8CED37DB-F62A-4B98-A30D-68202E9E134B.htm). Since then, it has become a fundamental part of the Java ecosystem and is currently maintained by the FasterXML organization. Jackson is widely used in enterprise and open-source projects due to its modular architecture and comprehensive support for various data formats. It offers a mature and efficient framework for converting Java objects to JSON and vice versa, making it an ideal choice for developers who require fast and scalable JSON processing.

Jackson provides several core modules, including:

* [**jackson-core**](https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core): The low-level streaming API for reading and writing JSON.
* [**jackson-databind**](https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind/2.19.0): Offers high-level data binding using ObjectMapper.
* [**jackson-annotations**](https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-annotations): Provides annotation support for customizing serialization/deserialization.

Baeldung (2023) explains that “the three main approaches for processing JSON with Jackson are the streaming API, the tree model, and data binding.” The Streaming API efficiently reads and writes JSON as a stream of tokens. This method is particularly advantageous for handling large files or operating in low-memory environments. Next is the Data Binding feature, which automates the conversion of JSON into Java objects and vice versa through the ObjectMapper. This capability is the most commonly utilized aspect of Jackson. Lastly, the Tree Model offers a way to parse JSON into a JsonNode tree, allowing for easy traversal and modification of the data programmatically. Together, these modes provide a comprehensive framework for working with JSON.

The Jackson API offers advanced features, including polymorphic type handling, filtering, and integration with various data formats such as XML and YAML. Developers can utilize annotations like @JsonProperty, @JsonIgnore, and @JsonInclude to manage objects' mapping. For instance, @JsonProperty("firstName") enables developers to rename fields in the resulting JSON, while @JsonIgnore can exclude specific fields from serialization.

Jackson's flexibility comes from its ability to manage deeply nested JSON structures, work with generic types, and serialize custom types using tailored serializers and deserializers. Its high performance and reliability make it an outstanding choice for microservices and other distributed systems.

In the end, Jackson’s JSON API is a powerful and flexible tool that provides everything needed to handle JSON in Java applications efficiently. Its broad feature set, modular architecture, and excellent performance make it the go-to solution for developers across many industries.

References:

baeldung. (2023, September 28). *Jackson Json series*. Baeldung. https://www.baeldung.com/jackson

*JsonAnyGetter annotation in Jackson*. Tutorialspoint. (n.d.-c). https://www.tutorialspoint.com/jackson\_annotations/jackson\_annotations\_jsonanygetter.htm

FasterXML. (n.d.). *FasterXML/jackson: Main portal page for the jackson project*. GitHub. https://github.com/FasterXML/jackson

*Explore Jackson annotations*. Tutorialspoint. (n.d.-a). https://www.tutorialspoint.com/jackson\_annotations/index.htm

baeldung. (2025, March 26). *Jackson Streaming API*. Baeldung. https://www.baeldung.com/jackson-streaming-api#bd-extracting-json-parts

Jenkov, J. (2019, February 2). *Jackson installation*. Jenkov.com Tech & Media Labs - Resources for Developers, IT Architects and Technopreneurs. https://jenkov.com/tutorials/java-json/jackson-installation.html