Kristopher Kuenning

10/12/2025

CSD420

Module 11: Programming Assignment

JSON-P

JSON stands for JavaScript Object Notation, which is a lightweight text-based format used for storing and exchanging data between systems, applications, and services. JSON is also language-independent, meaning it can be used with languages such as Java, Python, C#, PHP, and many others. Typically, JSON is composed of two primary structures: objects and arrays. Objects are key or value pairs, while arrays are ordered lists. JSON is the universal language of data exchange, being simple, readable, and widely supported by developers due to its role as the primary source of communication between modern web services, mobile apps, and APIs. In 2013, JSON-P introduced the initial API under the javax.json namespace, starting with Java EE 7. From 2017 to 2018, JSON-P 1.1 expanded the specifications with updates for JSON and Java 8 under Jakarta JSON processing.

JSON-P, also known as Jakarta JSON Processing, is the official JSON processing API for the Java platform. It is comprised of two different but complementary programming models. A high-level Object model and a low-level but fast streaming API:

* Object Model API – This model builds and manipulates in-memory JSON structures like JsonObject or JsonArray, like XML or DOM. The Object Model API is ideal for developers who need random access, transformations, or to compose new JSON content before writing all the content out.
* Streaming API – This API model is an event-driven, style processor that reads and writes JSON token-by-token, utilizing both JsonPaser or JsonGenerator. The Streaming API enables fast and memory-efficient processing for both large projects and pipelines.

Core parsing and generation JSON-P update added excellent support for key JSON standards like:

* JSON Pointer – Addressing elements with JSON documentation.
* JSON Patch – Expressing sequences of add/remove/replace/move/copy, which exposed JSONPatch or JSONPatchBuilder.
* JSON Merge Patch – A simpler, more object-oriented, different style update.

Altogether, these features enable developers to update, query, and diff JSON without manual string manipulation. API also added Java 8 updates, such as builders or streams, which would provide a better hook for high-volume processing.

Usage patterns of both features and processes support for JSON-P are:

* Reading JSON – The use of JSON object models provides JSONArray navigation with map/collection-like methods. These modifications are suited for moderate-sized documentation developers.
* Reading/Writing JSON – Large or streaming data provides a cursor that yields events like START\_OBJECT and VALUE\_STRING, which minimizes memory footprints and improves throughput.

JSON-P, compared to other processing frameworks such as Jackson, Gson, and Moshi, is ideal for object binding. JSON-P also complements the other processing frameworks by offering a standard API with a general purpose for JSON processing, streaming, and patching that works consistently across Jakarta runtimes and Java SE. Developers seeking a spec-defined patch or a minimal-footprint streaming software with a merge-merge approach should consider JSON-P, which can be combined with JSON-B when needed.

In all, Jakarta JSON processing provides Java developers with a stable toolkit for reading, writing, querying, and updating JSON using both object model and streaming styles, as well as standards-based support for pointers, patching, and merging.

References

* *Developing applications for Oracle Weblogic Server*. Oracle Help Center. (2024, July 16). <https://docs.oracle.com/en/middleware/standalone/weblogic-server/14.1.1.0/wlprg/java-api-for-json-proc.html?com#GUID-53ED16E0-0669-4698-AAAB-BB0C7B9A977B>
* *Open liberty docs*. Open Liberty Docs. (n.d.). <https://openliberty.io/docs/latest/json-p-b.html>
* *W3schools.com*. W3Schools Online Web Tutorials. (n.d.). https://www.w3schools.com/js/js\_json\_jsonp.asp