

JACKSON - OVERVIEW

http://www.tutorialspoint.com/jackson/jackson_overview.htm

Copyright © tutorialspoint.com

Jackson is a simple Java-based library to serialize Java objects to JSON and vice versa.

Features of Jackson

- **Easy to use** – Jackson API provides a high-level facade to simplify commonly used use-cases.
- **No need to create mapping** – Jackson API provides default mapping for most of the objects to be serialized.
- **Performance** – Jackson is quite fast, consumes less memory space, and is suitable for large object graphs or systems.
- **Clean JSON** – Jackson creates clean and compact JSON results which are easy to read.
- **No Dependency** – Jackson library does not require any other library apart from JDK.
- **Open Source** – Jackson library is open source and free to use.

Process JSON using Jackson

Jackson provides three different ways to process JSON –

- **Streaming API** – It reads and writes JSON content as discrete events. `JsonParser` reads the data, whereas `JsonGenerator` writes the data.
 - It is the most powerful approach among the three.
 - It has the lowest overhead and it provides the fastest way to perform read/write operations.
 - It is analogous to **Stax parser** for XML.
- **Tree Model** – It prepares an in-memory tree representation of the JSON document. `ObjectMapper` build tree of `JsonNode` nodes. It is most flexible approach. It is analogous to DOM parser for XML.
- **Data Binding** – It converts JSON to and from Plain Old Java Object *POJO* using property accessor or using annotations. `ObjectMapper` reads/writes JSON for both types of data bindings. Data binding is analogous to **JAXB parser** for XML. Data binding is of two types –
 - **Simple Data Binding** – It converts JSON to and from Java Maps, Lists, Strings, Numbers, Booleans, and null objects.
 - **Full Data Binding** – It converts JSON to and from any Java type.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js