**JACKSON API**:

* Very popular and efficient ***java-based library*** to ***serialize*** or ***map java object*** to ***JSON*** and ***vice versa***.

**Features:**

* Provides a high-level façade to simplify commonly used uses cases.
* Provides **default mapping** for most of the **objects** to be **serialized**.
* Low memory footprint and is suitable for large objects graphs or systems.
* Clean and compact JSON results which is easy to read.
* Does not require other libraries except JDK.

Provides ***three ways*** for ***processing JSON*:**

1. **Streaming API**: **JsonParser** ***reads*** the data whereas **JsonGenerator** ***writes*** the data, it is most powerful approach among three and is a lowest overhead and fastest to read and write.
2. **Tree Model**: Prepares in-memory tree representation of the JSON document. **ObjectMapper** **build tree** of **JsonNode *nodes***. It is most flexible approach.
3. **Data Binding**: converts JSON to and from POJO using property accessor or annotations.

* **Simple Data Binding**: ***converts JSON*** to and from **Java Maps**, **Lists**, **Strings**, **Numbers**, **Booleans** and **null objects**.
* **Full Data Binding**: ***converts JSON*** to and from **JAVA type**.

**ObjectMapper** ***reads/writes JSON*** for ***both types*** of ***data bindings***. Data Binding is most convenient way.

**\*Generating Getter and Setter methods Intellij:**

* Code menu > click Generate > Getter and Setter

[ generates for the current values of the class fields].

**\* Overriding toString() for the getter and Setter methods Intellij:**

* Code menu > click Generate > Getter and Setter

**[**overrides toString() method for current values of class fields].