**JACKSON API**

The Jackson API library for Java is a library that is used for mapping java objects to JSON or from JSON to Java. It is a common library that is used whenever one wants to parse data from JSON or vice versa using the programming language Java.

One of the features of JSON is that it stores data in objects and when one is using Java this is no different. When we use this Jackson API, we have the ability to serialize Java Objects to JSON and vice versa. Serialization means converting an object into a string to use and we can also deserialize which is the opposite. Converting from a string to an object.

Some of the best features of the Jackson API are its ease of use, no need for mapping, another sords there is a default mapping of all objects, performance is increased due to the low memory required which is great for larger systems, produces clean JSON that is easy to read, there are no dependencies other than the standard JDK and it is open-source making it easy to access and modify if need be.

If one wants to process JSON with the Jackson API we have three main ways. We can stream the API which involves reading and writing JSON content events. With these events we use the JsonParser which is what reads the data and the JsonGenerator which writes the data.

We can also use the Tree Model which prepares “in memory tree representation of the JSON document.” (tutorialspoint.com) In addition with this we have the ObjectMapper build tree of JSONNodes. This acts similar to a DOM parser for XML.

Lastly, we have Data Binding which is where we covert JSON to and form objects. There are two main ways to do this and this is the Simple Data Binding way of converting Java maps, strings, numbers, Booleans, numbers and null objects and Full Data Binding which will convert JSON to and form any single Java Type.

When it comes to setting up the Jackson API there is one core JAR file and two other files that use the core file. These are the Jackson Core, Jackson Annotaitons, and the Jackson Databind. In addition there are a few other projects that help to parse other data formats. An example of this could be to read and write CBOR you can add the Jackson-dataformat-cbor artifact classpath.

To be able to use any Jackson project you must add the JAR File to the classpath of the application. There are a few ways to do this:

First one needs to decide which version to use of the API. In order to do this go to the Jackson Website to match which versions are available.

Once one has decided which version to use, then one of the ways to connect the API is by putting the JAR file directly into the classpath and this is slightly different with each IDE one may use. One thing to note is that if your project uses Maven to build, “you may need to add Jackson as a dependency to your projects POM File.” (tutorials.jenkov.com)

Jackson API JAR Links: <https://jar-download.com/artifacts/com.fasterxml.jackson.core>

Resources: <http://tutorials.jenkov.com/java-json/jackson-installation.html>

<https://stackoverflow.com/questions/3316762/what-is-deserialize-and-serialize-in-json>