What is the jsonObject :

Json : java script object notation

It is used to convert to java object into json object

Note : json object contain two types of apis

1 jackson api

2 Gson api

Jackson api contains 2 versions

1.x version

2.x version

1.x version is provided one java class

That is the Object Mapper class

ObjectMapper class contains 2 methods()

1 writeValueAsString (); -🡪 javaObject To jsonObect

2 readValue() --🡪 json To java Object

Note: we can download 2 jars files

1 jackson-mapper-asl jarFile

2 jackson-core-ask jarFile

SetJarFile: like this

Project BuildPath-> configuringBuildPath-> labraries-> addExtranalJars

Sample mapper Class:

Public class UtilityMapper{

Private static ObjectMapper mapper;

Static{

mapper = new ObjectMapper();

}

Private static String objectConvertIntoJsonObject(Object obect)

// we can write the model class to convert an object this Object method();

{

String jsonResult =” “,

jsonResult=*mapper*.writeValueAsString(object);

**return** jsonResult;

}

}

// we can use this class and method anywhere in the project to covert java object into jasonObject

Like this

**public** **class** App

{

**public** **static** **void** main( String[] args )

{

Employee emp = **new** Employee();

emp.setId(100);

emp.setName("kiran kumar");

emp.setDesignation("Java Developer");

emp.setSalary(5000);

emp.setEmpId("ind1252");

String jsonEmp= JsonUtility.*convertJavaObjToJsonObj*(emp);

System.***out***.println(jsonEmp);

JsonUtility - > it is the class

convertJavaObjTOJsonObj()-> it is the method();

JsonObject into javaObject to convertions

Hear we need to use this method

readValue();

to convert jsonObject to java Object

json Annotations:

if you want perform ur properties order wise you need to use this annotation

like this:

**import** org.codehaus.jackson.annotate.JsonPropertyOrder;

@JsonPropertyOrder(value = {"name",

"id",

"designation",

"salary",

"empId"})

**public** **class** Employee **implements** Serializable{

**private** **int** id;

**private** String name;

**private** **double** salary;

**private** String designation;

**private** String empId

note :

if you want to ignore some properties to ur jsonObject you can use this annotation like property level

@JsonIgnore

Like this:

**public** **class** Employee **implements** Serializable{

**private** **int** id;

**private** String name;

@JsonIgnore

**private** **double** salary;

**private** String designation;

**private** String empId;