

JACKSON - OBJECTMAPPER CLASS

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

Copyright © tutorialspoint.com

ObjectMapper is the main actor class of Jackson library. ObjectMapper class provides functionalities to convert Java objects to matching JSON constructs and vice versa. It uses instances of JsonParser and JsonGenerator for implementing actual reading/writing of JSON.

ObjectMapper Class Declaration

Following is the declaration for **org.codehaus.jackson.map.ObjectMapper** class –

```
public class ObjectMapper
    extends ObjectCodec
    implements Versioned
```

ObjectMapper Nested Classes

S.No.	Class & Description
1	static class ObjectMapper.DefaultTypeResolverBuilder Customized TypeResolverBuilder that provides type resolver builders used with so-called "default typing" <i>see enableDefaultTyping()</i> for details).
2	static class ObjectMapper.DefaultTyping Enumeration used with <i>enableDefaultTyping</i> to specify what kind of types <i>classes</i> default typing should be used for.

ObjectMapper Class Constructors

S.No.	Constructor & Description
1	ObjectMapper It is the default constructor, which constructs the default JsonFactory as necessary. Use StdSerializerProvider as its SerializerProvider, and BeanSerializerFactory as its SerializerFactory.
2	ObjectMapper(JsonFactoryjf) Construct mapper that uses specified JsonFactory for constructing necessary JsonParsers and/or JsonGenerators.
3	ObjectMapper(JsonFactoryjf, SerializerProvidersp, DeserializerProviderdp)
4	ObjectMapper(JsonFactoryjf, SerializerProvidersp, DeserializerProviderdp, SerializationConfigconfig, DeserializationConfigdconfig)
5	ObjectMapper(SerializerFactorysf) Deprecated. Use other constructors instead; note that you can set the serializer factory with – setSerializerFactory(org.codehaus.jackson.map.SerializerFactory)

ObjectMapper Methods

S.No.	Method & Description
-------	----------------------

- 1 **protected void _configAndWriteValue***JsonGeneratorjgen, Objectvalue*
Method called to configure the generator as necessary and then call write functionality.
- 2 **protected void _configAndWriteValue***JsonGeneratorjgen, Objectvalue, Class < ? > viewClass*
- 3 **protected Object _convertObjectfromValue, JavaTypetoValueType**
- 4 **protected DeserializationContext _createDeserializationContext**
JsonParserjp, DeserializationConfigcfg
- 5 **protected PrettyPrinter _defaultPrettyPrinter**
Helper method that should return default pretty-printer to use for generators constructed by this mapper, when instructed to use default pretty printer.
- 6 **protected JsonSerializer<Object> _findRootDeserializer**
DeserializationConfigcfg, JavaTypevalueType
Method called to locate deserializer for the passed root-level value.
- 7 **protected JsonToken _initForReading***JsonParserjp*
Method called to ensure that given parser is ready for reading content for data binding.
- 8 **protected Object _readMapAndClose***JsonParserjp, JavaTypevalueType*
- 9 **protected Object _readValue***DeserializationConfigcfg, JsonParserjp, JavaTypevalueType*
Actual implementation of value reading + binding operation.
- 10 **protected Object _unwrapAndDeserialize**
JsonParserjp, JavaTyperootType, DeserializationContexttxt, JsonSerializer < Object > deser
- 11 **boolean canDeserialize***JavaTypetype*
Method that can be called to check whether mapper thinks it could deserialize an Object of given type.
- 12 **boolean canSerialize***Class < ? > type*
Method that can be called to check whether mapper thinks it could serialize an instance of given Class.
- 13 **ObjectMapper configure***DeserializationConfig. Featuref, booleanstate*
Method for changing state of an on/off deserialization feature for this object mapper.
- 14 **ObjectMapper configure***JsonGenerator. Featuref, booleanstate*
Method for changing state of an on/off JsonGenerator feature for JsonFactory instance this object mapper uses.
- 15 **ObjectMapper configure***JsonParser. Featuref, booleanstate*
Method for changing state of an on/off JsonParser feature for JsonFactory instance this object mapper uses.
- 16 **ObjectMapper configure***SerializationConfig. Featuref, booleanstate*
Method for changing state of an on/off serialization feature for this object mapper.

17	JavaType constructType <i>Type</i>	Convenience method for constructing JavaType out of given type <i>typically java. lang. Class</i> , but without explicit context.
18	<T> T convertValueObject <i>fromValue, Class < T > toValueType</i>	Convenience method for doing two-step conversion from given value, into instance of given value type.
19	<T> T convertValueObject <i>fromValue, JavaTypetoValueType</i>	
20	<T> T convertValueObject <i>fromValue, TypeReferencetoValueTypeRef</i>	
21	DeserializationConfig copyDeserializationConfig	Method that creates a copy of the shared default DeserializationConfig object that defines configuration settings for deserialization.
22	SerializationConfig copySerializationConfig	Method that creates a copy of the shared default SerializationConfig object that defines configuration settings for serialization.
23	ArrayNode createArrayNode	Note – return type is co-variant, as basic ObjectCodec abstraction cannot refer to concrete node types <i>asitisapartofcorepackage, whereasimplsarepartofmapperpackage</i> .
24	ObjectNode createObjectNode	Note – return type is co-variant, as basic ObjectCodec abstraction cannot refer to concrete node types <i>asitisapartofcorepackage, whereasimplsarepartofmapperpackage</i>
25	ObjectWriter defaultPrettyPrintingWriter	Deprecated. Since 1.9, use writerWithDefaultPrettyPrinter instead.
26	ObjectMapper disable <i>DeserializationConfig. Feature. . . f</i>	Method for enabling specified DeserializationConfig features.
27	ObjectMapper disable <i>SerializationConfig. Feature. . . f</i>	Method for enabling specified DeserializationConfig features.
28	ObjectMapper disableDefaultTyping	Method for disabling automatic inclusion of type information; if so, only explicitly annotated types <i>oneswithJsonTypeInfo</i> has additional embedded type information.
29	ObjectMapper enable <i>DeserializationConfig. Feature. . . f</i>	Method for enabling specified DeserializationConfig features.
30	ObjectMapper enable <i>SerializationConfig. Feature. . . f</i>	Method for enabling specified DeserializationConfig features.
31	ObjectMapper enableDefaultTyping	Convenience method that is equivalent to calling.

32	ObjectMapper enableDefaultTyping <i>ObjectMapper. DefaultTyping</i> <i>dti</i>
	Convenience method that is equivalent to calling.
33	ObjectMapper enableDefaultTyping <i>ObjectMapper. DefaultTypingapplicability, JsonTypeInfo. AsincludeAs</i>
	Method for enabling automatic inclusion of type information, needed for proper deserialization of polymorphic types <i>unless types are annotated with JsonTypeInfo</i> .
34	ObjectMapper enableDefaultTypingAsProperty <i>ObjectMapper. DefaultTypingapplicability, StringpropertyName</i>
	Method for enabling automatic inclusion of type information -- needed for proper deserialization of polymorphic types <i>unless types have been annotated with JsonTypeInfo</i> -- using "As.PROPERTY" inclusion mechanism and specified property name to use for inclusion default being "@class" since default type information always uses class name as type identifier
35	ObjectWriter filteredWriter <i>FilterProvider filterProvider</i>
	Deprecated. Since 1.9, use <i>writerFilterProvider</i> instead.
36	JsonSchema generateJsonSchemaClass <i><?> t</i>
	Generate Json-schema instance for specified class.
37	JsonSchema generateJsonSchemaClass <i><?> t, SerializationConfig cfg</i>
	Generate Json-schema instance for specified class, using specific serialization configuration
38	DeserializationConfig getDeserializationConfig
	Method that returns the shared default DeserializationConfig object that defines configuration settings for deserialization.
39	DeserializerProvider getDeserializerProvider
40	JsonFactory getJsonFactory
	Method that can be used to get hold of JsonFactory that this mapper uses if it needs to construct JsonParsers and/or JsonGenerators.
41	JsonNodeFactory getNodeFactory
	Method that can be used to get hold of JsonNodeFactory that this mapper will use when directly constructing root JsonNode instances for Trees.
42	SerializationConfig getSerializationConfig
	Method that returns the shared default SerializationConfig object that defines configuration settings for serialization.
43	SerializerProvider getSerializerProvider
44	SubtypeResolver getSubtypeResolver
	Method for accessing subtype resolver in use.
45	TypeFactory getTypeFactory

Accessor for getting currently configured TypeFactory instance.

46 **VisibilityChecker<?> getVisibilityChecker**

Method for accessing currently configured visibility checker; object used for determining whether given property element `method`, `field`, `constructor` can be auto-detected or not.

47 **boolean isEnabledDeserializationConfig.Feature f**

48 **boolean isEnabledJsonGenerator.Feature f**

49 **boolean isEnabledJsonParser.Feature f**

50 **boolean isEnabledSerializationConfig.Feature f**

51 **ObjectWriter prettyPrintingWriterPrettyPrinter pp**

52 **ObjectReader reader**

Factory method for constructing ObjectReader with default settings.

53 **ObjectReader readerClass<?> type**

Factory method for constructing ObjectReader that reads or update instances of specified type.

54 **ObjectReader readerFormatSchema schema**

Factory method for constructing ObjectReader that will pass specific schema object to JsonParser used for reading content.

55 **ObjectReader readerInjectableValues injectableValues**

Factory method for constructing ObjectReader that will use specified injectable values.

56 **ObjectReader readerJavaType type**

Factory method for constructing ObjectReader that will read or update instances of specified type.

57 **ObjectReader readerJsonNodeFactory f**

Factory method for constructing ObjectReader that will use specified JsonNodeFactory for constructing JSON trees.

58 **ObjectReader readerTypeReference<?> type**

Factory method for constructing ObjectReader that will read or update instances of specified type.

59 **ObjectReader readerForUpdatingObject valueToUpdate**

Factory method for constructing ObjectReader that will update given Object usually Bean, but can be a Collection or Map as well, but NOT an array with JSON data.

60 **JsonNode readTreebyte[] content**

Method to deserialize JSON content as tree expressed using set of JsonNode instances.

61 **JsonNode readTreeFile file**

Method to deserialize JSON content as tree expressed using set of JsonNode instances.

62 **JsonNode readTreeInputStream in**
Method to deserialize JSON content as tree expressed using set of JsonNode instances.

63 **JsonNode readTreeJsonParser jp**
Method to deserialize JSON content as tree expressed using set of JsonNode instances.

64 **JsonNode readTreeJsonParser jp, DeserializationConfig cfg**
Method to deserialize JSON content as tree expressed using set of JsonNode instances.

65 **JsonNode readTreeReader r**
Method to deserialize JSON content as tree expressed using set of JsonNode instances.

66 **JsonNode readTreeString content**
Method to deserialize JSON content as tree expressed using set of JsonNode instances.

67 **JsonNode readTreeURL source**
Method to deserialize JSON content as tree expressed using set of JsonNode instances.

68 **<T> T readValuebyte[] src, Class<T> valueType**

69 **<T> T readValuebyte[] src, int offset, int len, Class<T> valueType**

70 **<T> T readValuebyte[] src, int offset, int len, JavaType valueType**

71 **<T> T readValuebyte[] src, int offset, int len, TypeReference valueTypeRef**

72 **<T> T readValuebyte[] src, JavaType valueType**

73 **<T> T readValuebyte[] src, TypeReference valueTypeRef**

74 **<T> T readValueFile src, Class<T> valueType**

75 **<T> T readValueFile src, JavaType valueType**

76 **<T> T readValueFile src, TypeReference valueTypeRef**

77 **<T> T readValueInputStream src, Class<T> valueType**

78 **<T> T readValueInputStream src, JavaType valueType**

79 **<T> T readValueInputStream src, TypeReference valueTypeRef**

80 **<T> T readValueJsonNode root, Class<T> valueType**
Convenience method for converting results from given JSON tree into given value type.

81 **<T> T readValueJsonNode root, JavaType valueType**
Convenience method for converting results from given JSON tree into given value type.

82 **<T> T readValueJsonNode root, TypeReference valueTypeRef**
Convenience method for converting results from given JSON tree into given value type.

83 **<T> T readValueJsonParser jp, Class<T> valueType**
Method to deserialize JSON content into a non-container type *it can be an array type,*

however – typically a bean, array or a wrapper type like Boolean.

84 **<T> T readValueJsonParser jp, Class<T> valueType, DeserializationConfig cfg**

Method to deserialize JSON content into a non-container type it can be an array type, however – typically a bean, array or a wrapper type like Boolean.

85 **<T> T readValueJsonParser jp, JavaType valueType**

Method to deserialize JSON content into a Java type, reference to which is passed as argument.

86 **<T> T readValueJsonParser jp, JavaType valueType, DeserializationConfig cfg**

Method to deserialize JSON content into a Java type, reference to which is passed as argument.

87 **<T> T readValueJsonParser jp, TypeReference<?> valueTypeRef**

Method to deserialize JSON content into a Java type, reference to which is passed as argument.

88 **<T> T readValueJsonParser jp, TypeReference<?> valueTypeRef, DeserializationConfig cfg**

Method to deserialize JSON content into a Java type, reference to which is passed as argument.

89 **<T> T readValueReader src, Class<T> valueType**

90 **<T> T readValueReader src, JavaType valueType**

91 **<T> T readValueReader src, TypeReference valueTypeRef**

92 **<T> T readValueString content, Class<T> valueType**

93 **<T> T readValueString content, JavaType valueType**

94 **<T> T readValueString content, TypeReference valueTypeRef**

95 **<T> T readValueURL src, Class<T> valueType**

96 **<T> T readValueURL src, JavaType valueType**

97 **<T> T readValueURL src, TypeReference valueTypeRef**

98 **<T> MappingIterator<T> readValuesJsonParser jp, Class<T> valueType**

Method for reading sequence of Objects from parser stream.

99 **<T> MappingIterator<T> readValuesJsonParser jp, JavaType valueType**

Method for reading sequence of Objects from parser stream.

100 **<T> MappingIterator<T> readValuesJsonParser jp, TypeReference<?> valueTypeRef**

Method for reading sequence of Objects from parser stream.

101 **void registerModuleModule module**

Method for registering a module that can extend functionality provided by this mapper; for example, by adding providers for custom serializers and deserializers.

102	void registerSubtypesClass<?>... classes
	Method for registering specified class as a subtype, so that typename-based resolution can link supertypes to subtypes as an alternative to using annotations.
103	void registerSubtypesNamedType... types
	Method for registering specified class as a subtype, so that typename-based resolution can link supertypes to subtypes as an alternative to using annotations.
104	ObjectReader schemaBasedReaderFormatSchema schema
105	ObjectWriter schemaBasedWriterFormatSchema schema
106	ObjectMapper setAnnotationIntrospectorAnnotationIntrospector ai
	Method for changing AnnotationIntrospector used by this mapper instance for both serialization and deserialization.
107	void setDateFormatDateFormat dateFormat
	Method for configuring the default DateFormat to use when serializing time values as Strings, and deserializing from JSON Strings.
108	ObjectMapper setDefaultTypingTypeResolverBuilder<?> typer
	Method for enabling automatic inclusion of type information, using specified handler object for determining which types this affects, as well as details of how information is embedded.
109	ObjectMapper setDeserializationConfigDeserializationConfig cfg
	Method for replacing the shared default deserialization configuration object.
110	ObjectMapper setDeserializerProviderDeserializerProvider p
	Method for setting specific DeserializerProvider to use for handling the caching of JsonDeserializer instances.
111	void setFiltersFilterProvider filterProvider
112	void setHandlerInstantiatorHandlerInstantiator hi
	Method for configuring HandlerInstantiator to use for creating instances of handlers such as serializers, deserializers, type and type id resolvers, given a class.
113	ObjectMapper setInjectableValuesInjectableValues injectableValues
114	ObjectMapper setNodeFactoryJsonNodeFactory f
	Method for specifying JsonNodeFactory to use for constructing root level tree nodes via method createObjectNode()
115	ObjectMapper setPropertyNamingStrategyPropertyNamingStrategy s
	Method for setting custom property naming strategy to use.
116	ObjectMapper setSerializationConfigSerializationConfig cfg
	Method for replacing the shared default serialization configuration object.
117	ObjectMapper setSerializationInclusionJsonSerialize.Inclusion incl

Method for setting default POJO property inclusion strategy for serialization.

118 **ObjectMapper setSerializerFactorySerializerFactory f**

Method for setting specific SerializerFactory to use for constructing bean serializers.

119 **ObjectMapper setSerializerProviderSerializerProvider p**

Method for setting specific SerializerProvider to use for handling caching of JsonSerializer instances.

120 **void setSubtypeResolverSubtypeResolver r**

Method for setting custom subtype resolver to use.

121 **ObjectMapper setTypeFactoryTypeFactory f**

Method that can be used to override TypeFactory instance used by this mapper.

122 **ObjectMapper setVisibilityJsonMethod forMethod, JsonAutoDetect.Visibility visibility**

Convenience method that allows changing configuration for underlying VisibilityCheckers, to change details of what kinds of properties are auto-detected.

123 **void setVisibilityCheckerVisibilityChecker<?> vc**

Method for setting currently configured visibility checker; object used to determine whether given property element method, field, constructor can be auto-detected or not.

125 **JsonParser treeAsTokensJsonNode n**

Method for constructing a JsonParser out of JSON tree representation.

126 **<T> T treeToValueJsonNode n, Class<T> valueType**

Convenience conversion method that binds data provided the JSON tree contains a specific value usually bean type.

127 **ObjectWriter typedWriterClass<?> rootType**

Deprecated. Since 1.9, use writerWithTypeClass instead.

128 **ObjectWriter typedWriterJavaType rootType**

Deprecated. Since 1.9, use writerWithTypeJavaType instead.

129 **ObjectWriter typedWriterTypeReference<?> rootType**

Deprecated. Since 1.9, use writerWithTypeTypeReference instead.

130 **ObjectReader updatingReaderObject valueToUpdate**

Deprecated. Since 1.9, use readerForUpdatingjava.lang.Object instead.

131 **<T extends JsonNode> T valueToTreeObject fromValue**

Reverse of treeToValueorg.codehaus.jackson.JsonNode, java.lang.Class; given a value usually bean, constructs equivalent JSON Tree representation.

132	Version version	Method that will return version information stored and reads from jar that contains this class.
133	ObjectWriter viewWriterClass<?> serializationView	Deprecated. Since 1.9, use writerWithViewClass instead.
134	ObjectMapper withModuleModule module Fluent-style alternative to registerModuleorg.codehaus.jackson.map.Module;	
135	ObjectWriter writer	Convenience method for constructing ObjectWriter with default settings.
136	ObjectWriter writerDateFormat df	Factory method for constructing ObjectWriter that serializes objects using specified DateFormat; or, if null passed, using timestamp (64-bit number).
137	ObjectWriter writerFilterProvider filterProvider	Factory method for constructing ObjectWriter that serializes objects using specified filter provider.
138	ObjectWriter writerFormatSchema schema	Factory method for constructing ObjectWriter that passes specific schema object to JsonGenerator used for writing content.
139	ObjectWriter writerPrettyPrinter pp	Factory method for constructing ObjectWriter that serializes objects using specified pretty printer for indentation or if null, no pretty printer
140	ObjectWriter writerWithDefaultPrettyPrinter	Factory method for constructing ObjectWriter that serializes objects using the default pretty printer for indentation.
141	ObjectWriter writerWithTypeClass<?> rootType	Factory method for constructing ObjectWriter that serializes objects using specified root type, instead of actual runtime type of value.
142	ObjectWriter writerWithTypeJavaType rootType	Factory method for constructing ObjectWriter that serializes objects using specified root type, instead of actual runtime type of value.
143	ObjectWriter writerWithTypeTypeReference<?> rootType	Factory method for constructing ObjectWriter that serializes objects using specified root type, instead of actual runtime type of value.
144	ObjectWriter writerWithViewClass<?> serializationView	Factory method for constructing ObjectWriter that serializes objects using specified JSON View filter.
145	void writeTreeJsonGenerator jgen, JsonNode rootNode	

Method to serialize given JSON Tree, using generator provided.

146 **void writeTree**JsonGenerator jgen, JsonNode rootNode, SerializationConfig cfg

Method to serialize given Json Tree, using generator provided.

147 **void writeValueFile** resultFile, Object value

Method that can be used to serialize any Java value as JSON output, written to File provided.

148 **void writeValue**JsonGenerator jgen, Object value

Method that can be used to serialize any Java value as JSON output, using provided JsonGenerator.

149 **void writeValue**JsonGenerator jgen, Object value, SerializationConfig config

Method that can be used to serialize any Java value as JSON output, using provided JsonGenerator, configured as per passed configuration object.

150 **void writeValue**OutputStream out, Object value

Method that can be used to serialize any Java value as JSON output, using output stream provided using encoding JsonEncoding.UTF8.

151 **void writeValue**Writer w, Object value

Method that can be used to serialize any Java value as JSON output, using Writer provided.

152 **byte[] writeValueAsBytes**Object value

Method that used to serialize any Java value as a byte array.

153 **String writeValueAsString**Object value

Method that can be used to serialize any Java value as a String.

ObjectMapper class inherits methods from java.lang.Object.

ObjectMapper Example

Create the following Java program using any editor of your choice and save it in the folder **C:/> Jackson_WORKSPACE**

File: JacksonTester.java

```
import java.io.IOException;

import org.codehaus.jackson.JsonParseException;
import org.codehaus.jackson.map.JsonMappingException;
import org.codehaus.jackson.map.ObjectMapper;
import org.codehaus.jackson.map.SerializationConfig;

public class JacksonTester {
    public static void main(String args[]){

        ObjectMapper mapper = new ObjectMapper();
        String jsonString = "{\"name\":\"Mahesh\", \"age\":21}";

        //map json to student
```

```

    try {
        Student student = mapper.readValue(jsonString, Student.class);
        System.out.println(student);
        mapper.enable(SerializationConfig.Feature.INDENT_OUTPUT);
        jsonString = mapper.writeValueAsString(student);
        System.out.println(jsonString);
    }
    catch (JsonParseException e) { e.printStackTrace(); }
    catch (JsonMappingException e) { e.printStackTrace(); }
    catch (IOException e) { e.printStackTrace(); }
}

}

class Student {
    private String name;
    private int age;

    public Student(){}

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String toString(){
        return "Student [ name: "+name+", age: "+ age+ " ]";
    }
}

```

Verify the Result

Compile the classes using **javac** compiler as follows –

```
C:\Jackson_WORKSPACE>javac JacksonTester.java
```

Now execute the jacksonTester to see the result.

```
C:\Jackson_WORKSPACE>java JacksonTester
```

Verify the output –

```

Student [ name: Mahesh, age: 21 ]
{
  "name" : "Mahesh",
  "age" : 21
}

```

Processing math: 25%