JavaScript Object Notation (JSON4J) libraries in IBM WebSphere Application Server Web 2.0 and Mobile Toolkit Version 1.1.0

Contents

Getting started with the JSON4J libraries..... 1
Overview of JSON4J libraries..... 1
Limitations of the JSON4J libraries..... 1
Migration..... 2

JSON4J usage examples..... 2
JSON4J Java usage examples..... 2
JSON4J XML usage examples..... 4

JSON4J reference..... 5
JSON4J Libraries..... 5
Overview: JSON4J Libraries..... 6
Package com.ibm.json.java..... 6
Package com.ibm.json.xml..... 7
Overview List (JSON4J Libraries)..... 7
com.ibm.json.xml Contents..... 8
All Classes..... 8
All Classes..... 8
JSON..... 9
JSONArray..... 12
JSONArtifact..... 16
JSONObject..... 18
OrderedJSONObject..... 23
XMLToJSONTransformer..... 26
Hierarchy Overview (JSON4J Libraries)..... 30
Hierarchy for Package com.ibm.json.java..... 31
Hierarchy for Package com.ibm.json.xml..... 31
Deprecated API..... 31
Index..... 32
Organization of the API Library..... 37
Constant Field Values..... 39
Serialized Form..... 39

Index..... 41

Getting started with the JSON4J libraries

Welcome to the Getting Started section of the JavaScript Object Notation (JSON4J) libraries, the starting point for learning about JSON4J.

This `85528_WAS_web2mobile-json-help.pdf` file derives from the 8.5.5 `com.ibm.websphere.web2mobile.json.help` plug-in that was used in WebSphere Application Server 8.5.5.28 and previous versions. The `com.ibm.websphere.web2mobile.json.help` plug-in is available in the `855_WAS_all_product_doc.zip` file at `https://public.dhe.ibm.com/software/webserver/appserv/library/v85/`.

Avoid trouble: JSON4J is deprecated. Do not use JSON4J for new applications. For more information, see statements about the Web 2.0 and Mobile Toolkit deprecated feature in Deprecated features.

This section covers three main topics and six sub topics:

- Getting Started:
  - Overview
  - Limitations
  - Migration
- Usage Examples:
  - Java Usage Examples
  - XML Usage Examples
- Reference:
  - Javadoc

Note: JSON4J is now included with the application server runtime. The JSON4J library is not installed by the Web 2.0 and Mobile Toolkit.

Overview of JSON4J libraries

The JavaScript Object Notation (JSON4J) library is an implementation of a set of JavaScript Object Notation (JSON) handling classes for use within Java environments.

The JSON4J library provides the following functions:

- A simple Java model for constructing and manipulating data to be rendered as the JSON implementation.
- A fast transform for XML to JSON conversion, for situations where conversion from an XML reply from a Web service into a JSON structure is wanted for easy use in Asynchronous JavaScript and XML (Ajax) applications. The advantage of this is that Ajax-patterned applications can handle JSON formatted data without having to rely on ActiveX objects in Microsoft Internet Explorer XML transformations and other platform-specific XML parsers. In addition, JSON-formatted data tends to be more compact and efficient to transfer.
- A JSON string and stream parser that can generate the corresponding JSONObject, which represents that JSON structure in Java. You can then change that JSONObject and serialize the changes back to the JSON implementation.

Limitations of the JSON4J libraries

The JSON (JavaScript Object Notation) string parsing follows the guidelines presented by multiple reference sources on the format of JSON text.

Primarily, according to all sources reviewed, attribute names are always quoted strings.

See the following JSON structure:

```
{
  "attributeName": "foo"
}
```

However, for browsers that support attribute names without quotation marks; attribute names still evaluate correctly. This library follows the definition of the JSON format and is more rigid than most browsers. The library will not evaluate unquoted attribute names as valid JSON. Instead, JSON4J will generate a java.io.IOException exception. Therefore, always use correctly quoted attribute names in JSON text that you pass into the parse functions of this library.

Migration

If you are migrating from IBM WebSphere Application Server V6.1, V7.0, or V8.0 Feature Pack for Web 2.0 and Mobile to WebSphere Application Server Version 8.5, there are a few things to consider.

- JSON4J is not installed with the Web 2.0 and Mobile Toolkit. JSON4J is now part of the application server runtime. You no longer need to created a JSON4J shared library or embed the JSON4J library in your web application to use JSON4J.
- When migrating a previous JSON4J enabled application, you should remove the JSON4J library from your web application's `WEB-INF/lib` utility directory. You should also skip any steps that require creating shared library references when deploying your application.

JSON4J usage examples

The Usage examples section of the Developer's Guide for the JavaScript Object Notation (JSON4J) libraries contains example usage of the application programming interface (API) and expected output results.

The following basic examples are provided:

Example	Description
Java	This example shows two basic usage patterns of the Java API.
XML	This example shows how the XML converter API converts several document element types.

JSON4J Java usage examples

The core object, JSONObject, is an extension of HashMap with capabilities for handling JSON through parsing and construction.

The core object, JSONObject, is an extension of HashMap. This object has two capabilities for handling JSON. The first is a static function that parses a JSON string and returns a JSONObject instance that represents that particular string. The second function is to construct an entirely new JSON string by allocating a new JSONObject instance, then using the get, put, and remove methods to manipulate the attributes. The PUT function of the JSONObject does type checking to ensure that objects in the JSONObject instance have corresponding JSON representations.

Current types that can be put into or fetched through get from the map are as follows:

- java.lang.String
- java.lang.Boolean
- java.lang.Number
- com.ibm.json.java.JSONArray
- com.ibm.json.java.JSONObject

- null

You can then serialize any constructed JSONObject instance to a JSON string by using of one of several serialize methods. With these methods, you can specify an output stream, an output writer, or return a Java string of the JSON text, and indicate whether to use a verbose (formatted and indented for improved readability), or compact (single line, transmission efficient) format.

Demo 1

```
public void demoJson()
{
   String JSON = "{\"attribute\":\"foo\", \"number\":100.959}";
      try
      {
         JSONObject obj = (JSONObject)JSON.parse(JSON);
         Double dbl = (Double)obj.get("number");
         if (dbl == null || dbl.doubleValue() != 100.959)
         {
            throw new Exception("Numeric value was incorrect");
         }

         String str = (String)obj.get("attribute");
         if (dbl == null || !str.equals("foo"))
         {
            throw new Exception("String attribute was incorrect");
         }
         String jsonStr = obj.serialize(true);
         System.out.println(jsonStr);
      }
      catch (Exception ex)
      {
         ex.printStackTrace();
      }
}
```

The output of the previous example is:

```
{
  "attribute": "foo",
  "number": 100.959
}
```

Demo 2

```
public void demoJson()
{
   String JSON = "{\"attribute\":\"foo\", \"number\":100.959}";
      try
      {
         JSONObject obj = JSONObject.parse(JSON);
         Double dbl = (Double)obj.get("number");
         if (dbl == null || dbl.doubleValue() != 100.959)
         {
            throw new Exception("Numeric value was incorrect");
         }

         String str = (String)obj.get("attribute");
         if (dbl == null || !str.equals("foo"))
         {
            throw new Exception("String attribute was incorrect");
         }
         String jsonStr = obj.serialize(true);
         System.out.println(jsonStr);
      }
      catch (Exception ex)
      {
         ex.printStackTrace();
      }
}
```

The output of the previous example is:

```
{
  "attribute": "foo",
```

```
  "number": 100.959
}
```

Demo 3

```
public void demoJson2()
{
   try
   {
      JSONObject obj = new JSONObject();
      obj.put("attribute", "foo");
      obj.put("number", new Double(100.959));
      String jsonStr = obj.serialize(true);
      System.out.println(jsonStr);
   }
   catch (Exception ex)
   {
      ex.printStackTrace();
   }
}
```

The output of the previous example is:

```
{
  "attribute": "foo",
  "number": 100.959
}
```

Refer to specific API documentation for additional details.

JSON4J XML usage examples

The XML converter provides a simple static helper class for converting XML documents to JSON format with selectable output methods and verbosity levels.

The XML converter provides a simple static helper class where you can provide an XML document as a stream or as a file on disk. The method of output is also selectable; you can stream the resulting JavaScript Object Notation (JSON) to an output stream or obtain it as a Java String. You can also specify how verbose to make the generated JSON. The default is a compact form with no indentations or new lines, which is the most efficient format for transmission. Verbose form is indented and spaced, which is a more readable format and is the mode to enable when you want to debug and examine the JSON structure generated.

See the following examples that demonstrate how some basic XML element formats are converted to JSON formats:

XML Input Example

```
<getValuesReturn return="true">
   <attribute attrValue="value"/>
   <String>First item</String>
   <String>Second item</String>
   <String>Third item</String>
   <TextTag>Text!</TextTag>
   <EmptyTag/>
   <TagWithAttrs attr1="value1" attr2="value2" attr3="value3"/>
   <TagWithAttrsAndText attr1="value1" attr2="value2" attr3="value3">Text!</TagWithAttrsAndText>
</getValuesReturn>
```

Conversion to JSON (verbose form)

```
{
   "getValuesReturn" : {
      "return" : "true",
      "TextTag" : "Text!",
      "String" : [
         "First item",
         "Second item",
```

```
            "Third item"
         ],
         "TagWithAttrsAndText" : {
            "content" : "Text!",
            "attr3" : "value3",
            "attr2" : "value2",
            "attr1" : "value1"
         }
         ,
         "EmptyTag" : true,
         "attribute" : {
            "attrValue" : "value"
         }
         ,
         "TagWithAttrs" : {
            "attr3" : "value3",
            "attr2" : "value2",
            "attr1" : "value1"
         }
      }
   }
```

JSON4J reference

The Reference section of the Developer's Guide for the JavaScript Object Notation (JSON4J) libraries contains the documentation for the Application Programming Interface (API).

For detailed API documentation, see the JSON4J API documentation.

JSON4J Libraries

The JSON4J Libraries provide Java APIs for working with JSON data structures.

Overview

The JSON4J Libraries documentation is organized using a frame-based structure for easy navigation. The documentation includes:

- Package hierarchies and class listings
- Detailed package, class, and interface descriptions
- Complete API reference documentation

Documentation Structure

The JavaDoc documentation is divided into three main frames:

Package List Frame
Displays all hierarchies in the JSON4J Libraries (30% of left column)

Package Frame
Shows all classes and interfaces, excluding non-static nested types (70% of left column)

Class Frame
Contains detailed package, class, and interface descriptions (80% of right column)

Accessing the Documentation

The documentation is designed to be viewed using frames. If your browser does not support frames, you can access the non-frame version of the documentation through the overview summary page.

Tip: For the best viewing experience, use a frame-capable web browser to navigate the complete API documentation.

Related concepts

Non-frame version
The JSON4J Libraries provide Java APIs for working with JSON data structures and XML to JSON transformations.

Package hierarchies
Navigation frame providing access to all classes and packages in the JSON4J Libraries.

Related reference

All classes and interfaces
Complete list of all classes and interfaces in the JSON4J Libraries.

Overview: JSON4J Libraries

The JSON4J Libraries provide Java APIs for working with JSON data structures and XML to JSON transformations.

About JSON4J Libraries

The JSON4J Libraries are a collection of Java packages that enable developers to work with JSON (JavaScript Object Notation) data in Java applications. These libraries provide comprehensive support for creating, parsing, and manipulating JSON objects and arrays, as well as transforming XML data to JSON format.

Packages

The JSON4J Libraries consist of the following packages:

Package	Description
com.ibm.json.java	Core JSON processing classes including JSON, JSONObject, JSONArray, JSONArtifact, and OrderedJSONObject for creating and manipulating JSON data structures.
com.ibm.json.xml	XML to JSON transformation utilities, including the XMLToJSONTransformer class for converting XML documents to JSON format.

Documentation Information

This documentation was generated on March 8, 2012 using javadoc (IBM Internal Version 1.7.1).

Related concepts

Class Hierarchy Tree
Complete class and interface hierarchy for all packages in the JSON4J Libraries.

Deprecated APIs
This page lists deprecated APIs in the JSON4J Libraries.

Related reference

Index
Alphabetical index of all classes, interfaces, constructors, methods, and packages in the JSON4J Libraries.

Help Documentation
Organization of the API Library - This API (Application Programming Interface) library has pages corresponding to the items in the navigation bar.

Package com.ibm.json.java

Package com.ibm.json.java

Interface Summary

Interface	Description
JSONArtifact	Interface class to define a set of generic apis both JSONObject and JSON array implement.

Class Summary

Class	Description
"JSON" on page 9	Helper class that does generic parsing of a JSON stream and returns the appropriate JSON structure (JSONArray or JSONObject).
"JSONArray" on page 12	Extension of ArrayList that only allows values which are JSON-able.
"JSONObject" on page 18	Models a JSON Object.
"OrderedJSONObject" on page 23	Extension of the basic JSONObject.

Package com.ibm.json.xml

Package com.ibm.json.xml

Package com.ibm.json.xml

Class Summary

Class	Description
XMLToJSONTransformer	This class is a static helper for various ways of converting an XML document/InputStream into a JSON stream or String.

Overview List (JSON4J Libraries)

Navigation frame providing access to all classes and packages in the JSON4J Libraries.

All Classes

View a complete list of all classes in the JSON4J Libraries.

- All Classes

Packages

The JSON4J Libraries contain the following packages:

- com.ibm.json.java - Core JSON processing classes (same as `package-frame.html`)
- com.ibm.json.xml - XML to JSON transformation utilities

Metadata

This documentation was generated on March 8, 2012 using javadoc (IBM Internal Version 1.7.1).

com.ibm.json.xml Contents

com.ibm.json.xml Contents

com.ibm.json.xml

com.ibm.json.xml

Classes

- XMLToJSONTransformer

All Classes

Complete list of all classes and interfaces in the JSON4J Libraries.

Classes and Interfaces

The following table lists all classes and interfaces in the JSON4J Libraries:

Class/Interface	Package
JSON	com.ibm.json.java
JSONArray	com.ibm.json.java
JSONArtifact (interface)	com.ibm.json.java
JSONObject	com.ibm.json.java
OrderedJSONObject	com.ibm.json.java
XMLToJSONTransformer	com.ibm.json.xml

All Classes

Alphabetical list of all classes and interfaces in the JSON4J Libraries.

Overview

This page provides an alphabetical listing of all classes and interfaces in the JSON4J Libraries. Interfaces are shown in italics.

J

JSON
Class in com.ibm.json.java

JSONArray
Class in com.ibm.json.java

JSONArtifact
Interface in com.ibm.json.java

JSONObject
Class in com.ibm.json.java

O

OrderedJSONObject
Class in com.ibm.json.java

X

XMLToJSONTransformer
Class in com.ibm.json.xml

JSON

Helper class that does generic parsing of a JSON stream and returns the appropriate JSON structure (JSONArray or JSONObject).

Class hierarchy

```
java.lang.Object
  extended by com.ibm.json.java.JSON
```

```
public class JSON
extends java.lang.Object
```

Helper class that does generic parsing of a JSON stream and returns the appropriate JSON structure (JSONArray or JSONObject). Note that it is slightly more efficient to directly parse with the appropriate object than to use this class to do a generalized parse.

Constructor Summary

Constructor and Description
`JSON()`

Method Summary

Modifier and Type	Method and Description
static JSONArtifact	`parse(java.io.InputStream is)` Parse an InputStream of JSON text into a JSONArtifact.
static JSONArtifact	`parse(java.io.InputStream is, boolean order)` Parse a InputStream of JSON text into a JSONArtifact.
static JSONArtifact	`parse(java.io.Reader reader)` Parse a Reader of JSON text into a JSONArtifact.
static JSONArtifact	`parse(java.io.Reader reader, boolean order)` Parse a Reader of JSON text into a JSONArtifact.
static JSONArtifact	`parse(java.lang.String str)` Parse a string of JSON text into a JSONArtifact.
static JSONArtifact	`parse(java.lang.String str, boolean order)` Parse a string of JSON text into a JSONArtifact.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

JSON

```
public JSON()
```

Method Detail

parse

```
public static JSONArtifact parse(java.io.Reader reader,
                                 boolean order)
                          throws java.io.IOException
                                 java.lang.NullPointerException
```

Parse a Reader of JSON text into a JSONArtifact.

Parameters:

- `reader` - The character reader to read the JSON data from.
- `order` - Boolean flag indicating if the order of the JSON data should be preserved. This parameter only has an effect if the stream is JSON Object { ... } formatted data. Note that the provided reader is not closed on completion of read; that is left to the caller.

Returns: Returns an instance of JSONArtifact (JSONObject, OrderedJSONObject, or JSONArray), corrisponding to if the input stream was Object or Array notation.

Throws:

- `java.io.IOException` - Thrown on IO errors during parse.
- `java.lang.NullPointerException` - Thrown if reader is null

parse

```
public static JSONArtifact parse(java.io.Reader reader)
                          throws java.io.IOException
                                 java.lang.NullPointerException
```

Parse a Reader of JSON text into a JSONArtifact. This call is the same as JSON.parse(reader, false). Note that the provided reader is not closed on completion of read; that is left to the caller.

Parameters:

- `reader` - The character reader to read the JSON data from.

Returns: Returns an instance of JSONArtifact (JSONObject, OrderedJSONObject, or JSONArray), corrisponding to if the input stream was Object or Array notation.

Throws:

- `java.io.IOException` - Thrown on IO errors during parse.
- `java.lang.NullPointerException` - Thrown if reader is null

parse

```
public static JSONArtifact parse(java.io.InputStream is,
                                 boolean order)
                          throws java.io.IOException
                                 java.lang.NullPointerException
```

Parse a InputStream of JSON text into a JSONArtifact. Note that the provided InputStream is not closed on completion of read; that is left to the caller.

Parameters:

- `is` - The input stream to read from. The content is assumed to be UTF-8 encoded and handled as such.
- `order` - Boolean flag indicating if the order of the JSON data should be preserved. This parameter only has an effect if the stream is JSON Object { ... } formatted data.

Returns: Returns an instance of JSONArtifact (JSONObject or JSONArray), corrisponding to if the input stream was Object or Array notation.

Throws:

- `java.io.IOException` - Thrown on IO errors during parse.
- `java.lang.NullPointerException` - Thrown if reader is null

parse

```
public static JSONArtifact parse(java.io.InputStream is)
                           throws java.io.IOException
                                  java.lang.NullPointerException
```

Parse an InputStream of JSON text into a JSONArtifact. This call is the same as JSON.parse(is, false). Note that the provided InputStream is not closed on completion of read; that is left to the caller.

Parameters:

- `is` - The input stream to read from. The content is assumed to be UTF-8 encoded and handled as such.

Returns: Returns an instance of JSONArtifact (JSONObject, OrderedJSONObject, or JSONArray), corrisponding to if the input stream was Object or Array notation.

Throws:

- `java.io.IOException` - Thrown on IO errors during parse.
- `java.lang.NullPointerException` - Thrown if reader is null

parse

```
public static JSONArtifact parse(java.lang.String str,
                                 boolean order)
                           throws java.io.IOException
                                  java.lang.NullPointerException
```

Parse a string of JSON text into a JSONArtifact.

Parameters:

- `str` - The String to read from.
- `order` - Boolean flag indicating if the order of the JSON data should be preserved. This parameter only has an effect if the stream is JSON Object { ... } formatted data.

Returns: Returns an instance of JSONArtifact (JSONObject or JSONArray), corrisponding to if the input stream was Object or Array notation.

Throws:

- `java.io.IOException` - Thrown on IO errors during parse.
- `java.lang.NullPointerException` - Thrown if str is null

parse

```
public static JSONArtifact parse(java.lang.String str)
                           throws java.io.IOException
                                  java.lang.NullPointerException
```

Parse a string of JSON text into a JSONArtifact. This call is the same as JSON.parse(str, false).

Parameters:

- `str` - The String to read from.

Returns: Returns an instance of JSONArtifact (JSONObject, OrderedJSONObject, or JSONArray), corrisponding to if the input stream was Object or Array notation.

Throws:

- `java.io.IOException` - Thrown on IO errors during parse.
- `java.lang.NullPointerException` - Thrown if str is null

JSONArray

Extension of ArrayList that only allows values which are JSON-able.

Class hierarchy

```
java.lang.Object
  extended by java.util.AbstractCollection
    extended by java.util.AbstractList
      extended by java.util.ArrayList
        extended by com.ibm.json.java.JSONArray
```

All implemented interfaces:

JSONArtifact, java.io.Serializable, java.lang.Cloneable, java.lang.Iterable, java.util.Collection, java.util.List, java.util.RandomAccess

```
public class JSONArray
extends java.util.ArrayList
implements JSONArtifact
```

Extension of ArrayList that only allows values which are JSON-able. See JSONObject for a list of valid values. Instances of this class are not thread-safe.

See Also: Serialized Form

Field Summary

Fields inherited from class java.util.AbstractList

modCount

Constructor Summary

Constructor and Description
`JSONArray()` Create a new instance of this class.
`JSONArray(int initialCapacity)` Create a new instance of this class with the specified initial capacity.

Method Summary

Modifier and Type	Method and Description
void	`add(int index, java.lang.Object element)`
boolean	`add(java.lang.Object element)`

Modifier and Type	Method and Description
boolean	`addAll(java.util.Collection collection)`
boolean	`addAll(int index, java.util.Collection collection)`
static JSONArray	`parse(java.io.InputStream is)` Convert a stream of JSONArray text into JSONArray form.
static JSONArray	`parse(java.io.Reader reader)` Convert a stream (in reader form) of JSONArray text into object form.
static JSONArray	`parse(java.lang.String str)` Convert a String of JSONArray text into object form.
java.lang.String	`serialize()` Convert this object into a String of JSON text.
java.lang.String	`serialize(boolean verbose)` Convert this object into a String of JSON text, specifying verbosity.
void	`serialize(java.io.OutputStream os)` Convert this object into a stream of JSON text.
void	`serialize(java.io.OutputStream os, boolean verbose)` Convert this object into a stream of JSON text.
void	`serialize(java.io.Writer writer)` Convert this object into a stream of JSON text.
void	`serialize(java.io.Writer writer, boolean verbose)` Convert this object into a stream of JSON text, specifying verbosity.
java.lang.Object	`set(int index, java.lang.Object element)`

Methods inherited from class java.util.ArrayList

clear, clone, contains, ensureCapacity, get, indexOf, isEmpty, lastIndexOf, remove, remove, removeRange, size, toArray, toArray, trimToSize

Methods inherited from class java.util.AbstractList

equals, hashCode, iterator, listIterator, listIterator, subList

Methods inherited from class java.util.AbstractCollection

containsAll, removeAll, retainAll, toString

Methods inherited from class java.lang.Object

finalize, getClass, notify, notifyAll, wait, wait, wait

Methods inherited from interface java.util.List

containsAll, equals, hashCode, iterator, listIterator, listIterator, removeAll, retainAll, subList

Constructor Detail

JSONArray

```
public JSONArray()
```

Create a new instance of this class.

JSONArray

```
public JSONArray(int initialCapacity)
```

Create a new instance of this class with the specified initial capacity.

Method Detail

add

```
public void add(int index,
                java.lang.Object element)
```

- Specified by: add in interface `java.util.List`
- Overrides: add in class `java.util.ArrayList`

add

```
public boolean add(java.lang.Object element)
```

- Specified by: add in interface `java.util.Collection`
- Specified by: add in interface `java.util.List`
- Overrides: add in class `java.util.ArrayList`

addAll

```
public boolean addAll(java.util.Collection collection)
```

- Specified by: `addAll` in interface `java.util.Collection`
- Specified by: `addAll` in interface `java.util.List`
- Overrides: `addAll` in class `java.util.ArrayList`

addAll

```
public boolean addAll(int index,
                      java.util.Collection collection)
```

- Specified by: `addAll` in interface `java.util.List`
- Overrides: `addAll` in class `java.util.ArrayList`

set

```
public java.lang.Object set(int index,
                            java.lang.Object element)
```

- Specified by: `set` in interface `java.util.List`
- Overrides: `set` in class `java.util.ArrayList`

parse

```
public static JSONArray parse(java.io.InputStream is)
                       throws java.io.IOException
```

Convert a stream of JSONArray text into JSONArray form.

- Parameters: `is` - The inputStream from which to read the JSON. It will assume the input stream is in UTF-8 and read it as such.
- Returns: The contructed JSONArray Object.
- Throws: `IOEXception` - Thrown if an underlying IO error from the stream occurs, or if malformed JSON is read,
- Throws: `java.io.IOException`

parse

```
public static JSONArray parse(java.io.Reader reader)
                       throws java.io.IOException
```

Convert a stream (in reader form) of JSONArray text into object form.

- Parameters: `reader` - The reader from which the JSONArray data is read.
- Returns: The contructed JSONArray Object.
- Throws: `IOEXception` - Thrown if an underlying IO error from the reader occurs, or if malformed JSON is read,
- Throws: `java.io.IOException`

parse

```
public static JSONArray parse(java.lang.String str)
                       throws java.io.IOException
```

Convert a String of JSONArray text into object form.

- Parameters: `str` - The JSONArray string to parse into a Java Object.
- Returns: The contructed JSONArray Object.
- Throws: `IOEXception` - Thrown if malformed JSON is read,
- Throws: `java.io.IOException`

serialize

```
public void serialize(java.io.OutputStream os)
               throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(os,false);

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `os` - The output stream to serialize data to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public void serialize(java.io.OutputStream os,
                      boolean verbose)
```

```
                throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(writer,false);

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `os` - The output stream to serialize data to.
- Parameters: `verbose` - Whether or not to write the JSON text in a verbose format.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public void serialize(java.io.Writer writer)
               throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(writer,false);

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `writer` - The writer which to serialize the JSON text to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public void serialize(java.io.Writer writer,
                      boolean verbose)
               throws java.io.IOException
```

Convert this object into a stream of JSON text, specifying verbosity.

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `writer` - The writer which to serialize the JSON text to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public java.lang.String serialize(boolean verbose)
                           throws java.io.IOException
```

Convert this object into a String of JSON text, specifying verbosity.

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `verbose` - Whether or not to serialize in compressed for formatted Strings.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public java.lang.String serialize()
                           throws java.io.IOException
```

Convert this object into a String of JSON text. Same as serialize(false);

- Specified by: `serialize` in interface `JSONArtifact`
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

JSONArtifact

Interface class to define a set of generic apis both JSONObject and JSON array implement.

Interface JSONArtifact

All known implementing classes:

JSONArray, JSONObject, OrderedJSONObject

```
public interface JSONArtifact
```

Interface class to define a set of generic apis both JSONObject and JSON array implement. This is namely so that functions such as serialize, which are common between the two, can be easily invoked.

Method Summary

Modifier and Type	Method and Description
java.lang.String	`serialize()` Convert this object into a String of JSON text.
java.lang.String	`serialize(boolean verbose)` Convert this object into a String of JSON text, specifying verbosity.
void	`serialize(java.io.OutputStream os)` Convert this object into a stream of JSON text.
void	`serialize(java.io.OutputStream os, boolean verbose)` Convert this object into a stream of JSON text.
void	`serialize(java.io.Writer writer)` Convert this object into a stream of JSON text.
void	`serialize(java.io.Writer writer, boolean verbose)` Convert this object into a stream of JSON text, specifying verbosity.

Method Detail

serialize

```
void serialize(java.io.OutputStream os)
        throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(os,false); Note that encoding is always written as UTF-8, as per JSON spec.

- Parameters: `os` - The output stream to serialize data to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
void serialize(java.io.OutputStream os,
               boolean verbose)
        throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(writer,false); Note that encoding is always written as UTF-8, as per JSON spec.

- Parameters: `os` - The output stream to serialize data to.
- Parameters: `verbose` - Whether or not to write the JSON text in a verbose format.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
void serialize(java.io.Writer writer)
        throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(writer,false);

- Parameters: `writer` - The writer which to serialize the JSON text to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
void serialize(java.io.Writer writer,
               boolean verbose)
        throws java.io.IOException
```

Convert this object into a stream of JSON text, specifying verbosity.

- Parameters: `writer` - The writer which to serialize the JSON text to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
java.lang.String serialize(boolean verbose)
                    throws java.io.IOException
```

Convert this object into a String of JSON text, specifying verbosity.

- Parameters: `verbose` - Whether or not to serialize in compressed for formatted Strings.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
java.lang.String serialize()
                    throws java.io.IOException
```

Convert this object into a String of JSON text. Same as serialize(false);

- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

JSONObject

Models a JSON Object.

Class hierarchy

```
java.lang.Object
  extended by java.util.AbstractMap
    extended by java.util.HashMap
      extended by com.ibm.json.java.JSONObject
```

All implemented interfaces:

JSONArtifact, java.io.Serializable, java.lang.Cloneable, java.util.Map

Direct known subclasses:

OrderedJSONObject

```
public class JSONObject
extends java.util.HashMap
implements JSONArtifact
```

Models a JSON Object. Extension of HashMap that only allows String keys, and values which are JSON-able.

JSON-able values are: null, and instances of String, Boolean, Number, JSONObject and JSONArray.

Instances of this class are not thread-safe.

See Also: Serialized Form

Nested Class Summary

Nested classes/interfaces inherited from class java.util.AbstractMap

java.util.AbstractMap.SimpleEntry, java.util.AbstractMap.SimpleImmutableEntry

Constructor Summary

Constructor and Description
`JSONObject()` Create a new instance of this class.

Method Summary

Modifier and Type	Method and Description
static boolean	`isValidObject(java.lang.Object object)` Return whether the object is a valid value for a property.
static boolean	`isValidType(java.lang.Class clazz)` Return whether the class is a valid type of value for a property.
static JSONObject	`parse(java.io.InputStream is)` Convert a stream of JSON text into object form.
static JSONObject	`parse(java.io.Reader reader)` Convert a stream (in reader form) of JSON text into object form.
static JSONObject	`parse(java.lang.String str)` Convert a String of JSON text into object form.
java.lang.Object	`put(java.lang.Object key, java.lang.Object value)` (non-Javadoc)
java.lang.String	`serialize()` Convert this object into a String of JSON text.
java.lang.String	`serialize(boolean verbose)` Convert this object into a String of JSON text, specifying verbosity.
void	`serialize(java.io.OutputStream os)`

Modifier and Type	Method and Description
	Convert this object into a stream of JSON text.
void	`serialize(java.io.OutputStream os, boolean verbose)` Convert this object into a stream of JSON text.
void	`serialize(java.io.Writer writer)` Convert this object into a stream of JSON text.
void	`serialize(java.io.Writer writer, boolean verbose)` Convert this object into a stream of JSON text, specifying verbosity.
java.lang.String	`toString()` Over-ridden toString() method.

Methods inherited from class java.util.HashMap

clear, clone, containsKey, containsValue, entrySet, get, isEmpty, keySet, putAll, remove, size, values

Methods inherited from class java.util.AbstractMap

equals, hashCode

Methods inherited from class java.lang.Object

finalize, getClass, notify, notifyAll, wait, wait, wait

Methods inherited from interface java.util.Map

equals, hashCode

Constructor Detail

JSONObject

```
public JSONObject()
```

Create a new instance of this class.

Method Detail

isValidObject

```
public static boolean isValidObject(java.lang.Object object)
```

Return whether the object is a valid value for a property.

- Parameters: `object` - The object to check for validity as a JSON property value.

isValidType

```
public static boolean isValidType(java.lang.Class clazz)
```

Return whether the class is a valid type of value for a property.

- Parameters: `clazz` - The class type to check for validity as a JSON object type.

parse

```
public static JSONObject parse(java.io.Reader reader)
                        throws java.io.IOException
```

Convert a stream (in reader form) of JSON text into object form.

- Parameters: `reader` - The reader from which the JSON data is read.
- Returns: The contructed JSON Object.
- Throws: `IOEXception` - Thrown if an underlying IO error from the reader occurs, or if malformed JSON is read,
- Throws: `java.io.IOException`

parse

```
public static JSONObject parse(java.lang.String str)
                        throws java.io.IOException
```

Convert a String of JSON text into object form.

- Parameters: `str` - The JSON string to parse into a Java Object.
- Returns: The contructed JSON Object.
- Throws: `IOEXception` - Thrown if malformed JSON is read,
- Throws: `java.io.IOException`

parse

```
public static JSONObject parse(java.io.InputStream is)
                        throws java.io.IOException
```

Convert a stream of JSON text into object form.

- Parameters: `is` - The inputStream from which to read the JSON. It will assume the input stream is in UTF-8 and read it as such.
- Returns: The contructed JSON Object.
- Throws: `IOEXception` - Thrown if an underlying IO error from the stream occurs, or if malformed JSON is read,
- Throws: `java.io.IOException`

serialize

```
public void serialize(java.io.OutputStream os)
               throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(os,false); Note that encoding is always written as UTF-8, as per JSON spec.

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `os` - The output stream to serialize data to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public void serialize(java.io.OutputStream os,
                      boolean verbose)
               throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(writer,false); Note that encoding is always written as UTF-8, as per JSON spec.

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `os` - The output stream to serialize data to.
- Parameters: `verbose` - Whether or not to write the JSON text in a verbose format.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public void serialize(java.io.Writer writer)
              throws java.io.IOException
```

Convert this object into a stream of JSON text. Same as calling serialize(writer,false);

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `writer` - The writer which to serialize the JSON text to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public void serialize(java.io.Writer writer,
                      boolean verbose)
              throws java.io.IOException
```

Convert this object into a stream of JSON text, specifying verbosity.

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `writer` - The writer which to serialize the JSON text to.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public java.lang.String serialize(boolean verbose)
                           throws java.io.IOException
```

Convert this object into a String of JSON text, specifying verbosity.

- Specified by: `serialize` in interface `JSONArtifact`
- Parameters: `verbose` - Whether or not to serialize in compressed for formatted Strings.
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

serialize

```
public java.lang.String serialize()
                           throws java.io.IOException
```

Convert this object into a String of JSON text. Same as serialize(false);

- Specified by: `serialize` in interface `JSONArtifact`
- Throws: `java.io.IOException` - Thrown on IO errors during serialization.

put

```
public java.lang.Object put(java.lang.Object key,
                            java.lang.Object value)
```

(non-Javadoc)

- Specified by: `put` in interface `java.util.Map`
- Overrides: `put` in class `java.util.HashMap`
- See Also: HashMap.put(java.lang.Object, java.lang.Object)

toString

```
public java.lang.String toString()
```

Over-ridden toString() method. Returns the same value as serialize(), which is a compact JSON String. If an error occurs in the serialization, the return will be of format: JSON Generation Error: [<some error>]

- Overrides: `toString` in class `java.util.AbstractMap`

OrderedJSONObject

Extension of the basic JSONObject.

Class hierarchy

```
java.lang.Object
  extended by java.util.AbstractMap
    extended by java.util.HashMap
      extended by com.ibm.json.java.JSONObject
        extended by com.ibm.json.java.OrderedJSONObject
```

All implemented interfaces:

JSONArtifact, java.io.Serializable, java.lang.Cloneable, java.util.Map

```
public class OrderedJSONObject
extends JSONObject
```

Extension of the basic JSONObject. This class allows control of the serialization order of attributes. The order in which items are put into the instance controls the order in which they are serialized out. For example, the last item put is the last item serialized.

JSON-able values are: null, and instances of String, Boolean, Number, JSONObject and JSONArray.

Instances of this class are not thread-safe.

See Also: Serialized Form

Nested Class Summary

Nested classes/interfaces inherited from class java.util.AbstractMap

java.util.AbstractMap.SimpleEntry, java.util.AbstractMap.SimpleImmutableEntry

Constructor Summary

Constructor and Description
`OrderedJSONObject()` Create a new instance of this class.

Method Summary

Modifier and Type	Method and Description
void	`clear()` (non-Javadoc)
java.lang.Object	`clone()`

Modifier and Type	Method and Description
	Returns a shallow copy of this HashMap instance: the keys and values themselves are not cloned.
java.util.Iterator	`getOrder()` Method to obtain the order in which the items will be serialized.
static JSONObject	`parse(java.io.InputStream is)` Convert a stream of JSON text into object form.
static JSONObject	`parse(java.io.Reader reader)` Convert a stream (in reader form) of JSON text into object form.
static JSONObject	`parse(java.lang.String str)` Convert a String of JSON text into object form.
java.lang.Object	`put(java.lang.Object key, java.lang.Object value)` Method to put a JSON'able object into the instance.
java.lang.Object	`remove(java.lang.Object key)` Method to remove an entry from the OrderedJSONObject instance.

Methods inherited from class com.ibm.json.java.JSONObject

isValidObject, isValidType, serialize, serialize, serialize, serialize, serialize, serialize, toString

Methods inherited from class java.util.HashMap

containsKey, containsValue, entrySet, get, isEmpty, keySet, putAll, size, values

Methods inherited from class java.util.AbstractMap

equals, hashCode

Methods inherited from class java.lang.Object

finalize, getClass, notify, notifyAll, wait, wait, wait

Methods inherited from interface java.util.Map

equals, hashCode

Constructor Detail

OrderedJSONObject

```
public OrderedJSONObject()
```

Create a new instance of this class.

Method Detail

parse

```
public static JSONObject parse(java.io.Reader reader)
                        throws java.io.IOException
```

Convert a stream (in reader form) of JSON text into object form.

- Parameters: `reader` - The reader from which the JSON data is read.
- Returns: The contructed JSON Object. Note that the JSONObject will be an instance of OrderedJSONObject and as such, attribute order is maintained.
- Throws: `IOEXception` - Thrown if an underlying IO error from the reader occurs, or if malformed JSON is read,
- Throws: `java.io.IOException`

parse

```
public static JSONObject parse(java.lang.String str)
                        throws java.io.IOException
```

Convert a String of JSON text into object form.

- Parameters: `str` - The JSON string to parse into a Java Object.
- Returns: The contructed JSON Object. Note that the JSONObject will be an instance of OrderedJSONObject and as such, attribute order is maintained.
- Throws: `IOEXception` - Thrown if malformed JSON is read,
- Throws: `java.io.IOException`

parse

```
public static JSONObject parse(java.io.InputStream is)
                        throws java.io.IOException
```

Convert a stream of JSON text into object form.

- Parameters: `is` - The InputStream from which to read the JSON. It will assume the input stream is in UTF-8 and read it as such.
- Returns: The contructed JSON Object. Note that the JSONObject will be an instance of OrderedJSONObject and as such, attribute order is maintained.
- Throws: `IOEXception` - Thrown if an underlying IO error from the stream occurs, or if malformed JSON is read,
- Throws: `java.io.IOException`

put

```
public java.lang.Object put(java.lang.Object key,
                            java.lang.Object value)
```

Method to put a JSON'able object into the instance. Note that the order of initial puts controls the order of serialization. Meaning that the first time an item is put into the object determines is position of serialization. Subsequent puts with the same key replace the existing entry value and leave serialization position alone. For moving the position, the object must be removed, then re-put.

- Specified by: `put` in interface `java.util.Map`
- Overrides: `put` in class `JSONObject`
- See Also: HashMap.put(java.lang.Object, java.lang.Object)

remove

```
public java.lang.Object remove(java.lang.Object key)
```

Method to remove an entry from the OrderedJSONObject instance.

- Specified by: `remove` in interface `java.util.Map`
- Overrides: `remove` in class `java.util.HashMap`
- See Also: HashMap.remove(java.lang.Object)

clear

```
public void clear()
```

(non-Javadoc)

- Specified by: `clear` in interface `java.util.Map`
- Overrides: `clear` in class `java.util.HashMap`
- See Also: HashMap.clear()

clone

```
public java.lang.Object clone()
```

Returns a shallow copy of this HashMap instance: the keys and values themselves are not cloned.

- Overrides: `clone` in class `java.util.HashMap`

getOrder

```
public java.util.Iterator getOrder()
```

Method to obtain the order in which the items will be serialized.

- Returns: An iterator that represents the attribute names in the order that they will be serialized.

XMLToJSONTransformer

This class is a static helper for various ways of converting an XML document/InputStream into a JSON stream or String.

Class hierarchy

```
java.lang.Object
  extended by com.ibm.json.xml.XMLToJSONTransformer
```

```
public class XMLToJSONTransformer
extends java.lang.Object
```

This class is a static helper for various ways of converting an XML document/InputStream into a JSON stream or String. For example, the XML document:

```
   <getvaluesreturn return="true">
     <attribute attrvalue="value"/>
     <string>First item</string>
     <string>Second item</string>
     <string>Third item</string>
     <texttag>Text!</texttag>
     <emptytag/>
     <tagwithattrs attr1="value1" attr2="value2" attr3="value3"/>
     <tagwithattrsandtext attr1="value1" attr2="value2" attr3="value3">Text!</
tagwithattrsandtext>
   </getvaluesreturn>
```

in JSON (in non-compact form) becomes

```
{
   "getValuesReturn" : {
      "return" : "true",
      "TextTag" : "Text!",
      "String" : [
         "First item",
         "Second item",
         "Third item"
      ],
      "TagWithAttrsAndText" : {
         "content" : "Text!",
         "attr3" : "value3",
         "attr2" : "value2",
         "attr1" : "value1"
      }
      ,
      "EmptyTag" : true,
      "attribute" : {
         "attrValue" : "value"
      }
      ,
      "TagWithAttrs" : {
         "attr3" : "value3",
         "attr2" : "value2",
         "attr1" : "value1"
      }
   }
}
```

Constructor Summary

Constructor and Description
`XMLToJSONTransformer()`

Method Summary

Modifier and Type	Method and Description
static java.lang.String	`transform(java.io.File xmlFile)` Method to take an XML file and return a String of the JSON format.
static java.lang.String	`transform(java.io.File xmlFile, boolean verbose)` Method to take an XML file and return a String of the JSON format.
static java.lang.String	`transform(java.io.InputStream xmlStream)` Method to take an input stream to an XML document and return a String of the JSON format.
static java.lang.String	`transform(java.io.InputStream xmlStream, boolean verbose)` Method to take an input stream to an XML document and return a String of the JSON format.

Modifier and Type	Method and Description
static void	`transform(java.io.InputStream XMLStream, java.io.OutputStream JSONStream)` Method to do the transform from an XML input stream to a JSON stream.
static void	`transform(java.io.InputStream XMLStream, java.io.OutputStream JSONStream, boolean verbose)` Method to do the transform from an XML input stream to a JSON stream.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

XMLToJSONTransformer

```
public XMLToJSONTransformer()
```

Method Detail

transform

```
public static void transform(java.io.InputStream XMLStream,
                             java.io.OutputStream JSONStream)
                      throws org.xml.sax.SAXException
                             java.io.IOException
```

Method to do the transform from an XML input stream to a JSON stream. Neither input nor output streams are closed. Closure is left up to the caller. Same as calling transform(inStream, outStream, false); (Default is compact form)

- Parameters: `XMLStream` - The XML stream to convert to JSON
- Parameters: `JSONStream` - The stream to write out JSON to. The contents written to this stream are always in UTF-8 format.
- Throws: `org.xml.sax.SAXException` - Thrown is a parse error occurs.
- Throws: `java.io.IOException` - Thrown if an IO error occurs.

transform

```
public static void transform(java.io.InputStream XMLStream,
                             java.io.OutputStream JSONStream,
                             boolean verbose)
                      throws org.xml.sax.SAXException
                             java.io.IOException
```

Method to do the transform from an XML input stream to a JSON stream. Neither input nor output streams are closed. Closure is left up to the caller.

- Parameters: `XMLStream` - The XML stream to convert to JSON
- Parameters: `JSONStream` - The stream to write out JSON to. The contents written to this stream are always in UTF-8 format.
- Parameters: `verbose` - Flag to denote whether or not to render the JSON text in verbose (indented easy to read), or compact (not so easy to read, but smaller), format.

- Throws: `org.xml.sax.SAXException` - Thrown if a parse error occurs.
- Throws: `java.io.IOException` - Thrown if an IO error occurs.

transform

```
public static java.lang.String transform(java.io.InputStream xmlStream)
                                  throws org.xml.sax.SAXException
                                         java.io.IOException
```

Method to take an input stream to an XML document and return a String of the JSON format. Note that the xmlStream is not closed when read is complete. This is left up to the caller, who may wish to do more with it. This is the same as transform(xmlStream,false)

- Parameters: `xmlStream` - The InputStream to an XML document to transform to JSON.
- Returns: A string of the JSON representation of the XML file
- Throws: `org.xml.sax.SAXException` - Thrown if an error occurs during parse.
- Throws: `java.io.IOException` - Thrown if an IOError occurs.

transform

```
public static java.lang.String transform(java.io.InputStream xmlStream,
                                         boolean verbose)
                                  throws org.xml.sax.SAXException
                                         java.io.IOException
```

Method to take an input stream to an XML document and return a String of the JSON format. Note that the xmlStream is not closed when read is complete. This is left up to the caller, who may wish to do more with it.

- Parameters: `xmlStream` - The InputStream to an XML document to transform to JSON.
- Parameters: `verbose` - Boolean flag denoting whther or not to write the JSON in verbose (formatted), or compact form (no whitespace)
- Returns: A string of the JSON representation of the XML file
- Throws: `org.xml.sax.SAXException` - Thrown if an error occurs during parse.
- Throws: `java.io.IOException` - Thrown if an IOError occurs.

transform

```
public static java.lang.String transform(java.io.File xmlFile,
                                         boolean verbose)
                                  throws org.xml.sax.SAXException
                                         java.io.IOException
```

Method to take an XML file and return a String of the JSON format.

- Parameters: `xmlFile` - The XML file to transform to JSON.
- Parameters: `verbose` - Boolean flag denoting whther or not to write the JSON in verbose (formatted), or compact form (no whitespace)
- Returns: A string of the JSON representation of the XML file
- Throws: `org.xml.sax.SAXException` - Thrown if an error occurs during parse.
- Throws: `java.io.IOException` - Thrown if an IOError occurs.

transform

```
public static java.lang.String transform(java.io.File xmlFile)
                                  throws org.xml.sax.SAXException
                                         java.io.IOException
```

Method to take an XML file and return a String of the JSON format. This is the same as transform(xmlStream,false)

- Parameters: `xmlFile` - The XML file to convert to JSON.

- Returns: A string of the JSON representation of the XML file
- Throws: `org.xml.sax.SAXException` - Thrown if an error occurs during parse.
- Throws: `java.io.IOException` - Thrown if an IOError occurs.

Hierarchy Overview (JSON4J Libraries)

Complete class and interface hierarchy for all packages in the JSON4J Libraries.

Package Hierarchies

View the class hierarchy for individual packages:

- com.ibm.json.java
- com.ibm.json.xml

Class Hierarchy

The following shows the complete class hierarchy for the JSON4J Libraries:

- java.lang.Object
  - java.util.AbstractCollection (implements java.util.Collection)
    - java.util.AbstractList (implements java.util.List)
      - java.util.ArrayList (implements java.lang.Cloneable, java.util.List, java.util.RandomAccess, java.io.Serializable)
        - com.ibm.json.java.JSONArray (implements com.ibm.json.java.JSONArtifact) - Represents a JSON array structure
  - java.util.AbstractMap (implements java.util.Map)
    - java.util.HashMap (implements java.lang.Cloneable, java.util.Map, java.io.Serializable)
      - com.ibm.json.java.JSONObject (implements com.ibm.json.java.JSONArtifact) - Represents a JSON object structure
        - com.ibm.json.java.OrderedJSONObject - JSON object that maintains insertion order
  - com.ibm.json.java.JSON - Utility class for JSON parsing and serialization
  - com.ibm.json.xml.XMLToJSONTransformer - Transforms XML documents to JSON format

Interface Hierarchy

The following shows the interface hierarchy for the JSON4J Libraries:

- com.ibm.json.java.JSONArtifact - Base interface for JSON data structures

Key Classes

JSONArray
Extends ArrayList to provide JSON array functionality with serialization support.

JSONObject
Extends HashMap to provide JSON object functionality with key-value pair storage.

OrderedJSONObject
Extends JSONObject to maintain the insertion order of properties.

JSON
Provides static utility methods for parsing and serializing JSON data.

XMLToJSONTransformer
Transforms XML documents into JSON format.

Related concepts

Overview Summary
The JSON4J Libraries provide Java APIs for working with JSON data structures and XML to JSON transformations.

Related reference

com.ibm.json.java Package Hierarchy
Hierarchy for Package com.ibm.json.java

com.ibm.json.xml Package Hierarchy
com.ibm.json.xml Hierarchy

All Classes
Alphabetical list of all classes and interfaces in the JSON4J Libraries.

Hierarchy for Package com.ibm.json.java

Hierarchy for Package com.ibm.json.java

Hierarchy Overview

- java.lang.Object
  - java.util.AbstractCollection (implements java.util.Collection)
    - java.util.AbstractList (implements java.util.List)
      - java.util.ArrayList (implements java.lang.Cloneable, java.util.List, java.util.RandomAccess, java.io.Serializable)
        - com.ibm.json.java.JSONArray (implements com.ibm.json.java.JSONArtifact)
  - java.util.AbstractMap (implements java.util.Map)
    - java.util.HashMap (implements java.lang.Cloneable, java.util.Map, java.io.Serializable)
      - com.ibm.json.java.JSONObject (implements com.ibm.json.java.JSONArtifact)
        - com.ibm.json.java.OrderedJSONObject
  - com.ibm.json.java.JSON

Interface Hierarchy

com.ibm.json.java.JSONArtifact

Hierarchy for Package com.ibm.json.xml

com.ibm.json.xml Hierarchy

Hierarchy for Package com.ibm.json.xml

Hierarchy Overview

- java.lang.Object
  - com.ibm.json.xml.XMLToJSONTransformer

Deprecated API

This page lists deprecated APIs in the JSON4J Libraries.

Deprecated API

This page contains information about deprecated APIs in the JSON4J Libraries. Deprecated APIs are those that are no longer recommended for use and may be removed in future versions of the library.

Currently, there are no deprecated APIs in this library.

Related information

For more information about the JSON4J Libraries, see the following resources:

- Overview
- Class Hierarchy
- Index
- Help

Index

Alphabetical index of all classes, interfaces, constructors, methods, and packages in the JSON4J Libraries.

A

add(int, Object)
Method in class com.ibm.json.java.JSONArray

add(Object)
Method in class com.ibm.json.java.JSONArray

addAll(Collection)
Method in class com.ibm.json.java.JSONArray

addAll(int, Collection)
Method in class com.ibm.json.java.JSONArray

C

clear()
Method in class com.ibm.json.java.OrderedJSONObject

clone()
Method in class com.ibm.json.java.OrderedJSONObject

Returns a shallow copy of this HashMap instance: the keys and values themselves are not cloned.

com.ibm.json.java
Package com.ibm.json.java

com.ibm.json.xml
Package com.ibm.json.xml

G

getOrder()
Method in class com.ibm.json.java.OrderedJSONObject

Method to obtain the order in which the items will be serialized.

I

isValidObject(Object)
Static method in class com.ibm.json.java.JSONObject

Return whether the object is a valid value for a property.

isValidType(Class)
Static method in class com.ibm.json.java.JSONObject

Return whether the class is a valid type of value for a property.

J

JSON
Class in com.ibm.json.java

Helper class that does generic parsing of a JSON stream and returns the appropriate JSON structure (JSONArray or JSONObject).

JSON()
Constructor for class com.ibm.json.java.JSON

JSONArray
Class in com.ibm.json.java

Extension of ArrayList that only allows values which are JSON-able.

JSONArray()
Constructor for class com.ibm.json.java.JSONArray

Create a new instance of this class.

JSONArray(int)
Constructor for class com.ibm.json.java.JSONArray

Create a new instance of this class with the specified initial capacity.

JSONArtifact
Interface in com.ibm.json.java

Interface class to define a set of generic apis both JSONObject and JSON array implement.

JSONObject
Class in com.ibm.json.java

Models a JSON Object.

JSONObject()
Constructor for class com.ibm.json.java.JSONObject

Create a new instance of this class.

O

OrderedJSONObject
Class in com.ibm.json.java

Extension of the basic JSONObject.

OrderedJSONObject()
Constructor for class com.ibm.json.java.OrderedJSONObject

Create a new instance of this class.

P

parse(Reader, boolean)
Static method in class com.ibm.json.java.JSON

Parse a Reader of JSON text into a JSONArtifact.

parse(Reader)
Static method in class com.ibm.json.java.JSON

Parse a Reader of JSON text into a JSONArtifact.

parse(InputStream, boolean)
Static method in class com.ibm.json.java.JSON

Parse a InputStream of JSON text into a JSONArtifact.

parse(InputStream)
Static method in class com.ibm.json.java.JSON

Parse an InputStream of JSON text into a JSONArtifact.

parse(String, boolean)
Static method in class com.ibm.json.java.JSON

Parse a string of JSON text into a JSONArtifact.

parse(String)
Static method in class com.ibm.json.java.JSON

Parse a string of JSON text into a JSONArtifact.

parse(InputStream)
Static method in class com.ibm.json.java.JSONArray

Convert a stream of JSONArray text into JSONArray form.

parse(Reader)
Static method in class com.ibm.json.java.JSONArray

Convert a stream (in reader form) of JSONArray text into object form.

parse(String)
Static method in class com.ibm.json.java.JSONArray

Convert a String of JSONArray text into object form.

parse(Reader)
Static method in class com.ibm.json.java.JSONObject

Convert a stream (in reader form) of JSON text into object form.

parse(String)
Static method in class com.ibm.json.java.JSONObject

Convert a String of JSON text into object form.

parse(InputStream)
Static method in class com.ibm.json.java.JSONObject

Convert a stream of JSON text into object form.

parse(Reader)
Static method in class com.ibm.json.java.OrderedJSONObject

Convert a stream (in reader form) of JSON text into object form.

parse(String)
Static method in class com.ibm.json.java.OrderedJSONObject

Convert a String of JSON text into object form.

parse(InputStream)
Static method in class com.ibm.json.java.OrderedJSONObject

Convert a stream of JSON text into object form.

put(Object, Object)
Method in class com.ibm.json.java.JSONObject

put(Object, Object)
Method in class com.ibm.json.java.OrderedJSONObject

Method to put a JSON'able object into the instance.

R

remove(Object)
Method in class com.ibm.json.java.OrderedJSONObject

Method to remove an entry from the OrderedJSONObject instance.

S

serialize(OutputStream)
Method in class com.ibm.json.java.JSONArray

Convert this object into a stream of JSON text.

serialize(OutputStream, boolean)
Method in class com.ibm.json.java.JSONArray

Convert this object into a stream of JSON text.

serialize(Writer)
Method in class com.ibm.json.java.JSONArray

Convert this object into a stream of JSON text.

serialize(Writer, boolean)
Method in class com.ibm.json.java.JSONArray

Convert this object into a stream of JSON text, specifying verbosity.

serialize(boolean)
Method in class com.ibm.json.java.JSONArray

Convert this object into a String of JSON text, specifying verbosity.

serialize()
Method in class com.ibm.json.java.JSONArray

Convert this object into a String of JSON text.

serialize(OutputStream)
Method in interface com.ibm.json.java.JSONArtifact

Convert this object into a stream of JSON text.

serialize(OutputStream, boolean)
Method in interface com.ibm.json.java.JSONArtifact

Convert this object into a stream of JSON text.

serialize(Writer)
Method in interface com.ibm.json.java.JSONArtifact

Convert this object into a stream of JSON text.

serialize(Writer, boolean)
Method in interface com.ibm.json.java.JSONArtifact

Convert this object into a stream of JSON text, specifying verbosity.

serialize(boolean)
Method in interface com.ibm.json.java.JSONArtifact

Convert this object into a String of JSON text, specifying verbosity.

serialize()
Method in interface com.ibm.json.java.JSONArtifact

Convert this object into a String of JSON text.

serialize(OutputStream)
Method in class com.ibm.json.java.JSONObject

Convert this object into a stream of JSON text.

serialize(OutputStream, boolean)
Method in class com.ibm.json.java.JSONObject

Convert this object into a stream of JSON text.

serialize(Writer)
Method in class com.ibm.json.java.JSONObject

Convert this object into a stream of JSON text.

serialize(Writer, boolean)
Method in class com.ibm.json.java.JSONObject

Convert this object into a stream of JSON text, specifying verbosity.

serialize(boolean)
Method in class com.ibm.json.java.JSONObject

Convert this object into a String of JSON text, specifying verbosity.

serialize()
Method in class com.ibm.json.java.JSONObject

Convert this object into a String of JSON text.

set(int, Object)
Method in class com.ibm.json.java.JSONArray

T

toString()
Method in class com.ibm.json.java.JSONObject

Over-ridden toString() method.

transform(InputStream, OutputStream)
Static method in class com.ibm.json.xml.XMLToJSONTransformer

Method to do the transform from an XML input stream to a JSON stream.

transform(InputStream, OutputStream, boolean)
Static method in class com.ibm.json.xml.XMLToJSONTransformer

Method to do the transform from an XML input stream to a JSON stream.

transform(InputStream)
Static method in class com.ibm.json.xml.XMLToJSONTransformer

Method to take an input stream to an XML document and return a String of the JSON format.

transform(InputStream, boolean)
Static method in class com.ibm.json.xml.XMLToJSONTransformer

Method to take an input stream to an XML document and return a String of the JSON format.

transform(File, boolean)
Static method in class com.ibm.json.xml.XMLToJSONTransformer

Method to take an XML file and return a String of the JSON format.

transform(File)
Static method in class com.ibm.json.xml.XMLToJSONTransformer

Method to take an XML file and return a String of the JSON format.

X

XMLToJSONTransformer
Class in com.ibm.json.xml

This class is a static helper for various ways of converting an XML document/InputStream into a JSON stream or String.

XMLToJSONTransformer()
Constructor for class com.ibm.json.xml.XMLToJSONTransformer

Organization of the API Library

Organization of the API Library - This API (Application Programming Interface) library has pages corresponding to the items in the navigation bar.

Overview

The Overview page is the front page of this API library and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain the following categories:

- Interfaces (italic)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class or Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class or interface description, summary tables, and detailed member descriptions:

Description

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class/interface declaration
- Class/interface description

Summary Tables

- Nested Class Summary
- Field Summary
- Constructor Summary
- Method Summary

Detail Descriptions

- Field Detail

- Constructor Detail
- Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

- Annotation Type declaration
- Annotation Type description
- Required Element Summary
- Optional Element Summary
- Required Element Detail
- Optional Element Detail

Enum

Each enum has its own separate page with the following sections:

- Enum declaration
- Enum description
- Enum Constant Summary
- Enum Constant Detail

Hierarchy Overview

There is a Hierarchy Overview page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

Deprecated API

The Deprecated API page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

Index

The Index contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

Previous | Next

These links take you to the next or previous class, interface, package, or related page.

Frames | No Frames

These links show and hide the HTML frames. All pages are available with or without frames.

Serialized Form

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

Constant Field Values

The Constant Field Values page lists the static final fields and their values.

Note: This help file applies to API documentation generated using the standard doclet.

Constant Field Values

This topic provides information about constant field values in the JSON4J Libraries.

Overview

This page lists the constant field values defined in the JSON4J Libraries. Constant fields are static final fields that contain fixed values used throughout the API.

Contents

No constant field values are currently documented for this API.

Related concepts

Overview
The JSON4J Libraries provide Java APIs for working with JSON data structures and XML to JSON transformations.

Tree
Complete class and interface hierarchy for all packages in the JSON4J Libraries.

Deprecated
This page lists deprecated APIs in the JSON4J Libraries.

Related reference

Index
Alphabetical index of all classes, interfaces, constructors, methods, and packages in the JSON4J Libraries.

Help
Organization of the API Library - This API (Application Programming Interface) library has pages corresponding to the items in the navigation bar.

Serialized Form

Serialization information for classes in the JSON4J Libraries.

Package com.ibm.json.java

Class com.ibm.json.java.JSONArray

Extends java.util.ArrayList **implements** Serializable

serialVersionUID: 9076798781015779954L

Class com.ibm.json.java.JSONObject

Extends java.util.HashMap **implements** Serializable

serialVersionUID: -3269263069889337298L

Class com.ibm.json.java.OrderedJSONObject

Extends JSONObject **implements** Serializable

serialVersionUID: -3269263069889337299L

Serialized Fields

order

```
java.util.ArrayList order
```

Index

A

all classes 8, 8
API documentation 5
API examples 2
attribute names 1

C

constant field values 39

G

getting started 1

J

Java APIs 8, 8, 39
Java usage examples 2
JavaScript Object Notation 1
JSON4J 1, 1, 1, 2, 2, 2, 4, 5
JSON4J Libraries 8, 8, 39
JSONObject 2

L

limitations 1

M

migration 2

O

overview 1

Q

quoted strings 1

R

reference 5

U

usage examples 2

W

WebSphere Application Server 2

X

XML converter 4
XML usage examples 4