OVERVIEW PACKAGE CLASS DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

java.util

Class Json

java.lang.Object
    java.util.Json

```
public class Json
extends Object
```

Constructor Summary

Constructors

Constructor and Description
`Json()` Json
`Json(File file)` Initializes the JSON object for the JSON string stored in the referenced File.
`Json(String json)` Creates the JSON object from the supplied string.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`boolean`	`contains(Object value)` Tests if some key contains the specified value.
`boolean`	`containsKey(String key)` Tests if the specified string is a key in this Json object.
`Enumeration`	`elements()` Returns an enumeration of the values in this Json object.
`protected void`	`finalize()` Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
`Object`	`get(String name)` Obtain the value from a name/value pair.
`boolean`	`getBoolean(String name)` Obtain the `boolean` from a name/value pair.
`double`	`getDouble(String name)` Obtain a `double` value from a name/value pair.
`float`	`getFloat(String name)` Obtain a `float` value from a name/value pair.
`int`	`getInt(String name)` Obtain a `int` value from a name/value pair.
`long`	`getLong(String name)` Obtain a `long` value from a name/value pair.
`String`	`getString(String name)` Obtain the `string` from a name/value pair.
`boolean`	`isEmpty()`

	Determines if the Json object has content.
`String[]`	`keyarray()` Provides a list of the keys or names for all name/value pairs in the Json object.
`Enumeration`	`keys()` Returns an enumeration of the keys in this Json object
`void`	`put(String name, double v)`
`void`	`put(String name, float v)`
`void`	`put(String name, int i)`
`void`	`put(String name, long i)`
`void`	`put(String name, Object value)`
`void`	`remove(String name)`
`void`	`save(File file)` Stores the JSON object in string form.
`void`	`save(String filename)` Stores the JSON object in string form.
`int`	`size()` Returns the number of names or keys in this Json object.
`String`	`toString()` Returns a string representation of the object.

Methods inherited from class java.lang.Object

`clone, equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Detail

Json

```
public Json()
```

Json

Json

```
public Json(String json)
```

Creates the JSON object from the supplied string.

Parameters:

`json` - string containing the JSON text representation

Json

```
public Json(File file)
```

Initializes the JSON object for the JSON string stored in the referenced File.

Parameters:

`file` - File object specifying the initial content.

Method Detail

toString

```
public String toString()
```

Description copied from class: `Object`

Returns a string representation of the object. In general, the `toString` method returns a string that "textually represents" this object. The result should be a concise but informative representation that is easy for a person to read. It is recommended that all subclasses override this method.

The `toString` method for class `Object` returns a string consisting of the name of the class of which the object is an instance, the at-sign character `` `@' ``, and the unsigned hexadecimal representation of the hash code of the object.

Overrides:

`toString` in class `Object`

Returns:

`a string representation of the object.`

get

```
public Object get(String name)
```

Obtain the value from a name/value pair.

Parameters:

`name - defines the key/name for the pair.`

Returns:

`an Object appropriate for the value. Numbers are returned as strings.`

getInt

```
public int getInt(String name)
           throws NumberFormatException
```

Obtain a `int` value from a name/value pair.

Parameters:

`name - defines the key/name for the pair.`

Returns:

`a int value`

Throws:

`NumberFormatException - if the value is not numeric or contains an illegal numeric character.`

getLong

```
public long getLong(String name)
             throws NumberFormatException
```

Obtain a `long` value from a name/value pair.

Parameters:

`name - defines the key/name for the pair.`

Returns:

`a long value`

Throws:

`NumberFormatException - if the value is not numeric or contains an illegal numeric character.`

getFloat

```
public float getFloat(String name)
               throws NumberFormatException
```

Obtain a `float` value from a name/value pair.

Parameters:

`name - defines the key/name for the pair.`

Returns:

`a float value`

Throws:

NumberFormatException - if the value is not numeric or contains an illegal numeric character.

getDouble

```
public double getDouble(String name)
                 throws NumberFormatException
```

Obtain a double value from a name/value pair.

Parameters:

name - defines the key/name for the pair.

Returns:

a double value

Throws:

NumberFormatException - if the value is not numeric or contains an illegal numeric character.

getString

```
public String getString(String name)
```

Obtain the string from a name/value pair.

Parameters:

name - defines the key/name for the pair.

Returns:

a string

getBoolean

```
public boolean getBoolean(String name)
```

Obtain the boolean from a name/value pair.

Parameters:

name - defines the key/name for the pair.

Returns:

a string

put

```
public void put(String name,
                Object value)
```

Parameters:

name - ?

value - ?

put

```
public void put(String name,
                int i)
```

Parameters:

name - ?

i - ?

put

```
public void put(String name,
                long i)
```

Parameters:

name - ?

i - ?

put

```
public void put(String name,
                float v)
```

Parameters:

name - ?

v - ?

put

```
public void put(String name,
                double v)
```

Parameters:

name - ?

v - ?

remove

```
public void remove(String name)
```

Parameters:

name - ?

elements

```
public Enumeration elements()
```

Returns an enumeration of the values in this Json object.

Returns:

an enumeration of the values.

keys

```
public Enumeration keys()
```

Returns an enumeration of the keys in this Json object

Returns:

an enumeration of the keys

containsKey

```
public boolean containsKey(String key)
```

Tests if the specified string is a key in this Json object. This is definitive. If you are just interested in whether or not there is a value for the key use `get()` and check for a `null` value. This is faster and would return `null` even if the key existed but had a JSON_NULL value.

Parameters:

key - possible key

Returns:

true if and only if the specified string is a key in this Json object, as determined by the equals method; false otherwise.

contains

```
public boolean contains(Object value)
```

Tests if some key contains the specified value.

Parameters:

`value - a value to search for`

Returns:

`true if and only if some key maps to the value argument in this Json object as determined by the equals method; false otherwise.`

Throws:

`NullPointerException - if the value is null`

keyarray

```
public String[] keyarray()
```

Provides a list of the keys or names for all name/value pairs in the Json object.

Returns:

`String[] containing name for each pair in the object.`

size

```
public int size()
```

Returns the number of names or keys in this Json object.

Returns:

`the number of keys`

isEmpty

```
public boolean isEmpty()
```

Determines if the Json object has content.

Returns:

`TRUE if the object is empty.`

save

```
public void save(String filename)
```

Stores the JSON object in string form.

Parameters:

`filename - specifying the destination`

save

```
public void save(File file)
```

Stores the JSON object in string form.

Parameters:

`file - File object specifying the destination`

finalize

```
protected void finalize()
                 throws Throwable
```

Description copied from class: `Object`

Called by the garbage collector on an object when garbage collection determines that there are no more references to the object. A subclass overrides the `finalize` method to dispose of system resources or to perform other cleanup.

Any exception thrown by the `finalize` method causes the finalization of this object to be halted, but is otherwise ignored.

The `finalize` method in `Object` does nothing.

Overrides:

`finalize` in class `Object`

Throws:

`Throwable` - [Need description!]

OVERVIEW PACKAGE CLASS DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD