

KSOAP API

Introduction:

In order to understand fully the complexities of KSoap I will take a detailed look at the API. Both it's packages and classes.

1. Package org.kobjects.serialization

Interface Summary

KvmSerializable

provides get and set methods for properties.

org.kobjects.serialization

All known Implementing Classes:

SoapObject

Overview:

public interface KvmSerializable

provides get and set methods for properties. Can be used to replace reflection (to some extent) for "serialization-aware" classes. Currently used in kSOAP and the RMS based kobjects object repository.

Method Summary

java.lang.object

getProperty(int index)

Returns the property at a specified index.

int

getPropertyCount()

returns the number of serializable properties.

void

getPropertyInfo(int index, **propertyInfo** info)

Fills the given property info record.

void

setProperty(int index, java.lang.object value)

sets the property with the given index to the given value.

Method Detail

getProperty

public java.lang.Object getProperty(int index)
Returns the property at a specified index (for serialization)

Parameters:

index - the specified index

Returns:

the serialized property

getPropertyCount

public int getPropertyCount()
returns the number of serializable properties

setProperty:

public void setProperty(int index, java.lang.Object value)
sets the property with the given index to the given value.

Parameters:

index - the index to be set

value - the value of the property

getPropertyInfo

public void getPropertyInfo(int index, PropertyInfo info)
Fills the given property info record.

Class Summary

ElementType

This class encapsulates type information.

PropertyInfo

This class is used to store information about each property an implementation of KvmSerializable exposes.

Class Element type

public class **ElementType** extends java.lang.Object

This class encapsulates type information.

Field Summary

-static java.lang.class	BOOLEAN_CLASS
ElementType	elementType Element type for array properties, null if not array prop.
static java.lang.class	INTEGER_CLASS
static java.lang.class	LONG_CLASS
boolean	multiRef if a property is multi-referenced, set this flag to true.
static java.lang.class	OBJECT_CLASS
static ElementType	OBJECT_TYPE
static java.lang.class	STRING_CLASS
java.lang.object	type Type of the property/elements.
static java.lang.class	VECTOR_CLASS

CONSTRUCTOR SUMMARY

ElementType()

ElementType(java.lang.Object type)

ElementType(java.lang.Object type, boolean multiRef, **ElementType** elementType)

Method Summary

void **clear()**

void **copy**(**ElementType** t2)

Methods inherited from class java.lang.Object

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

FIELD DETAIL

OBJECT_CLASS

public static final java.lang.class **OBJECT_CLASS**

STRING_CLASS

public static final java.lang.class STRING_CLASS

INTEGER_CLASS

public static final java.lang.Class **INTEGER_CLASS**

LONG_CLASS

public static final java.lang.class LONG_CLASS

BOOLEAN_CLASS

public ststic final java.lang.class BOOLEAN_CLASS

VECTOR_CLASS

public static final java.lang.Class VECTOR_CLASS

OBJECT_TYPE

public static final ElementType OBJECT_TYPE

type

public java.lang.Object type

Type of the property/elements. Should usually be an instance of class,

multiRef

public boolean multiRef

if a property is multit-referenced, set this flag to true.

elementType

public **ElementType** elementType

Element type for array properties, null if not array prop.

Constructor Detail.

ElementType

public ElementType()

ElementType

public ElementType(java.lang.Object type)

ElementType

public ElementType(java.lang.Object type,
boolean multiRef,
ElementType elementType)

Method Detail

clear

```
public void clear()
```

copy

```
public void copy(ElementType t2)
```

Class PropertyInfo

Field Summary:

java.lang.string name Name of the property

boolean nonpermanent The equivalent to transient, but named differently because transient is a reserved keyword.

Fields inherited from class org.kobjects.serialization.ElementType

BOOLEAN_CLASS, elementType, INTEGER_CLASS, LONG_CLASS, multiRef, OBJECT_CLASS, OBJECT_TYPE, STRING_CLASS, type, VECTOR_CLASS

Constructor Summary

PropertyInfo()

PropertyInfo(java.lang.String name, java.lang.Object type)

PropertyInfo(java.lang.String name, java.lang.Object type, boolean multiRef, **ElementType** elementType)

Method Summary

void clear()

Methods inherited from class org.kobjects.serialization.**ElementType**
copy

Methods inherited from class java.lang.Object

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

Field Detail

name

public java.lang.String name
Name of the property

nonpermanent

public PropertyInfo()

PropertyInfo

public PropertyInfo(java.lang.String name,
 java.lang.Object type,
 Element elementType)

Method Detail

clear

public void clear()

Overrides:

clear in class **ElementType**

Package org.Ksoap

Package org.ksoap Description

This package contains the basic set of classes required for SOAP (de)serialization.

This package contains the basic set of classes required for SOAP (de)serialization

Interface Summary

Marshal Abstract class for custom (de)serialization.

Class Summary

ClassMap This class provides various soap properties relevant for (de)serialization, including a method for defining mappings between java classes and XML element names.

Soap Contains some constant definitions only.

SoapEnvelope The SOAP envelope.

SoapObject A simple dynamic object that can be used to build soap calls without implenting KvmSerializable.

SoapParser A SOAP parser.

SoapPrimitive A class that is used to encapsulate primitive types (represented by a string in XML serialization).

SoapWriter A writer that is able to write objects wrt.

Exception Summary

SoapFault

Interface Marshal

org.ksoap Interface Marshal

All Known Implementing Classes:

MarshalDate, MarshalBase64, MarshalFloat, MarshalHashtable

public interface **Marshal**

Abstract class for custom (de)serialization.

Method Summary.

java.lang.Object **readInstance**(**SoapParser** parser, java.lang.String namespace, java.lang.String name, **ElementType** expected)

This methods read an instance from the given parser.

void **register**(**ClassMap** cm)

Register this Marshal with the given classMap.

void **writeInstance**(**SoapWriter** writer, java.lang.Object instance)

Write the instance to the given SoapWriter.

Method Detail

readInstance

public java.lang.Object readInstance(**SoapParser** parser,
 java.lang.String namespace,
 java.lang.String name,
 ElementType expected)
 throws java.io.IOException

This methods reads an instance from the given parser. For implementation, please note that the start and end tag must be consumed. This is not symmetric to writeInstance, but otherwise it would not be possible to access the attributes of the start tag here. The underlying XML parser is accessible using parser.parser.

writeInstance

public void writeInstance(**SoapWriter** writer,
 java.lang.Object instance)
 throws java.io.IOException

Write the instance to the given SoapWriter. In contrast to readInstance, it is not necessary to care about the surrounding start and end tags. Additional attributes must be written before anything else is written. The underlying xml writer is accessible using writer.writer.

register

public void register(**ClassMap** cm)

Register this Marshal with the given classMap

Class ClassMap

public class **ClassMap**
extends java.lang.Object

This class provides various SOAP properties relevant for (de)serialization, including a method for defining mappings between Java classes and XML element names.

Field Summary

protected java.util.Hashtable	classToQName Map from Java class names to XML name and namespace pairs
protected int	cnt
java.lang.String	enc
java.lang.String	env
boolean	implicitTypes Determines if type attributes are included for all objects written.
org.kxml.prefixMap	prefixMap
protected java.util.Hashtable	qNameToClass Map from XML qualified names to Java classes
int	version
java.lang.String	xsd

Constructor Summary

ClassMap()

deprecated Create a new class map using the 2001 version of the XML schema namespace

ClassMap(boolean legacy)

deprecated Creates a new Class map.

ClassMap(int version)

Method Summary

void **addMapping**(java.lang.String namespace, java.lang.String name, java.lang.Class clazz)

Defines a direct mapping from a namespace and name to a Java class (and vice versa)

void **addMapping**(java.lang.String namespace, java.lang.String namespace, java.lang.String name, **ElementType** expected)

void **addTemplate(SoapObject so)**

Adds a SoapObject to the class map.

java.lang.Object[] **getInfo**(java.lang.Object type, java.lang.Object instance)
Returns a string array containing the namespace, name, id and Marshal object for the given java object.

java.lang.Object **readInstance**(SoapParser parser, java.lang.String namespace, java.lang.String name, **ElementType** expected)
Returns a string array containing the namespace, name, id and Marshal object for the given java object.

java.lang.Object **readInstance**(SoapParser parser, java.lang.String namespace, java.lang.String name, **ElementType** expected)
Returns a new object read from the given parser.

Methods inherited from class java.lang.Object

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

Field Detail

implicitTypes

public boolean implicitTypes

Determines if type attributes are included for all objects written. If true, the type attribute is only written if the actual type differs from the declared type.

Version

public int version

prefixMap

public org.kxml.PrefixMap prefixMap

xsi

public java.lang.String xsi

xsd

public java.lang.String xsd

env

public java.lang.String env

enc

public java.lang.String enc

cnt

protected int cnt

qNameToClass

protected java.util.Hashtable qNameToClass
Map from xml qualified names to java classes

classToQName

protected java.util.Hashtable classToQName

Map from Java class names to XML name and namespace pairs

Constructor Detail

ClassMap

public ClassMap()

deprecated Create a new class map using the 2001 version of the XML schemanamespace.

ClassMap

public ClassMap(int version)

ClassMap

public ClassMap(boolean legacy)

deprecated Creates a new Class map. If the legacy flag is set to true, the 1999 version of the XML Schema namespace is used, otherwise the 2001 version.

Method Detail

readInstance

public java.lang.Object readInstance(**SoapParser** parser,
java.lang.String namespace,
java.lang.String name,
ElementType expected)
throws java.io.IOException

Returns a new object read from the given parser. If no mapping is found, null is returned. This method is used by the SoapParser in order to convert the XML code to Java objects.

getInfo

public java.lang.Object[] getInfo(java.lang.Object type, java.lang.Object instance)

Returns a string array containing the namespace, name, id and Marshal object for the given java object. This method is used by the SoapWriter in order to map Java objects to the corresponding SOAP section five XML code.

addMapping

public void addMapping(java.lang.String namespace,
java.lang.String name,
java.lang.Class clazz)

Defines a direct mapping from a namespace and name to a java class (and vice versa)

addTemplate

public void addTemplate(**SoapObject** so)

Adds a SoapObject to the class map. During parsing, objects of the given type (namespace/name) will be mapped to corresponding copies of the given SoapObject, maintaining the structure of the template.

Class Soap

public class Soap extends java.lang.Object

Constructor Summary

Soap()

Methods inherited from class java.lang.Object

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

Field Detail

VER10

public static final int VER10

VER11

public static final int VER11

VER12

public static final int VER12

ENC2001

public static final java.lang.String ENV2001

ENC2001

public static final java.lang.String ENC2001

ENV

public static final java.lang.String ENV

Namespace constant: <http://schemas.xmlsoap.org/soap/envelope/>

ENC

public static final java.lang.String ENC

Namespace constant: <http://schemas.xmlsoap.org/soap/encoding/>

XSD

public static final java.lang.String XSD

Namespace constant: <http://www.w3.org/2001/XMLSchema>

XSI

public static final java.lang.String XSI

Namespace constant: <http://www.w3.org/2001/XMLSchema>

XSD1999

public static final java.lang.String XSD1999

Namespace constant: <http://www.w3.org/1999/XMLSchema>

prefixMap

public static final org.kxml.PrefixMap[] prefixMap

Constructor Detail

Soap

public Soap()

Class SoapEnvelope

public class **SoapEnvelope** extends java.lang.Object

The SOAP envelope

Constructor Summary

SoapEnvelope()

deprecated

SoapEnvelope(ClassMap classMap)

Method Summary

java.lang.object **getBody()**
Returns the body object of the envelope.

java.lang.object **getResult()**

void **parse**(org.kxml.parser.AbstractionXmlParser parser)
Parses the SOAP envelope from the given parser

void **parserBody**(org.kxml.parser.AbstractionXmlParser parser)

void **parseHead**(org.Kxml.parser.AbstractXmlParser parser)

void **parseTail**(org.Kxml.parser.AbstractionXmlParser parser)

void **setBody**(java.lang.Object body)

void **setClassMap**(*classmap* classMap)

void **setEncodingStyle**(java.lang.String encodingStyle)
Sets the encoding style.

void **write**(org.kxml.io.AbstractionXmlWriter writer)
Writes the envelope and body to the given XML writer.

void **writeBody**(org.Kxml.io.AbstractionXmlWriter writer)
Overwrite this method for customised writing of the soa

void **writeHead**(org.Kxml.io.AbstractionXmlWriter writer)
Writes the head including the encoding style attribute and the body start tag

void **writeTail**(org.Kxml.io.AbstractXmlWriter writer)

Methods inherited from class java.lang.Object

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

Constructor Detail

SoapEnvelope

public SoapEnvelope()
deprecated

SoapEnvelope

public SoapEnvelope(**classMap** classMap)

Method Detail

getBody

public java.lang.Object getBody()

Returns the body object of the envelope.

getResult

public java.lang.object getResult()

Returns the first property of the body object

parse

public void parse(org.kxml.parser.AbstractXmlParser parser)

throws java.io.IOException

Parses the SOAP envelope from the given parser

parseHead

public void parseHead(org.kxml.parser.AbstractXmlParser parser)

throws java.io.IOException.

parseBody

public void parseBody(org.kxml.parser.AbstractionXmlParser parser)

throws java.io.IOException

setEncodingStyle

public void setEncodingStyle(java.lang.String encodingStyle)

Sets the encoding style.

write

public void write(org.kxml.io.AbstractionXmlWriter writer)

throws java.io.IOException

Writes the head including the encoding style attribute and the body start tag.

writeHead

public void writeHead(org.kxml.io.AbstractionXmlWriter writer)

throws java.io.IOException

Writes the head including the encoding style attribute and the body start tag

writeBody

public void writeBody(org.kxml.io.AbstractXmlWriter writer)

throws java.io.IOException

Overwrite this method for customised writing of the soap message body.

writeTail

public void writeTail(org.kxml.io.AbstractionXmlWriter writer)

throws java.io.IOException

setBody

```
public void setBody(java.lang.Object body)
```

setClassMap

```
public void setClassMap(classMap classMap)
```


Class SoapObject

public class **SoapObject** extends java.lang.Object implements KvmSerializable

A simple dynamic object that can be used to build soap calls without implementing KvmSerializable. Essentially, this is what goes inside the body of a soap envelope - it is the direct subelement of the body and all further subelements. Instead of this class, custom classes can be used if they implement the KvmSerializable interface.

Constructor Summary

SoapObject **addProperty**(java.lang.String namespace, java.lang.Object value)
Adds a property(parameter)to the object.

SoapObject

addProperty(java.lang.String name, **ElementType**, java.lang.Object value)
Deprecated. *Adds a property (parameter) to the object. This is essentially a sub element.*

SoapObject

addProperty(java.lang.String name, java.lang.Object value)
Adds a property (parameter) to the object.

boolean **equals**(java.lang.Object o)

java.lang.String **getName**()

java.lang.String **getNamespace**()

java.lang.Object **getProperty**(int index)
Returns a specific property at a certain index.

java.lang.Object **getPropertyCount**(java.lang.String name)

int **getPropertyCount**()
Returns the number of properties.

void **getPropertyInfo**(int index, **PropertyInfo** pi)
Places PropertyInfo of desired property into a designated PropertyInfo object.

SoapObject **newInstance**()
Creates a new SoapObject based on this, allows usage of SoapObjects as templates.

void **setProperty**(int index, java.lang.Object value)
Sets a specified property to a certain value.

Methods inherited from class java.lang.Object

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

Constructor Detail

SoapObject

public SoapObject(java.lang.String namespace,

java.lang.String name)

Creates a new SoapObject instance.

Parameters:

namespace - the namespace for the soap object

name - the name of the soap object

Method Detail

equals

public boolean equals(java.lang.Object o)

Overrides:

equals in class java.lang.Object

getName

public java.lang.String getName()

getNamespace

public java.lang.String getNamespace()

getProperty

public java.lang.Object getProperty(int index)

Returns a specific property at a certain index.

Specified by:

getProperty in interface **KvmSerializable**

Parameters:

index - the index of the desired property

Returns:

the desired property

getProperty

public java.lang.Object getProperty(java.lang.String name)

getPropertyCount

public int getPropertyCount()

Returns the number of properties

Specified by:

getPropertyCount in interface KvmSerializable

Returns:

the number of properties

getPropertyInfo

```
public void getPropertyInfo(int index, PropertyInfo pi)
```

Places PropertyInfo of desired property into a designated PropertyInfo object

Specified by:

getPropertyInfo in interface KvmSerializable

Parameters:

index - index of desired property

info - designated retainer of desired property

newInstance

```
public SoapObject newInstance()
```

Creates a new SoapObject based on this, allows usage of SoapObjects as templates. One application is to set the expected return type of a soap call if the server does not send explicit type information.

Returns:

a copy of this.

setProperty

```
public void setProperty(int index, java.lang.Object value)
```

Sets a specified property to a certain value.

Specified by:

setProperty in interface KvmSerializable

parameters:

index-the index of the specified property

value-the new value of the property

addProperty

```
public SoapObject addProperty(java.lang.String name, java.lang.Object value)
```

Adds a property (parameter) to the object. This is essentially a sub element.

Parameters:

name - The name of the property

value - the value of the property

addProperty

```
public SoapObject addProperty(java.lang.String name, ElementType type, java.lang.Object value)
```

Deprecated. *Adds a property (parameter) to the object. This is essentially a sub element.*

Parameters:

name - the name of the property

type - the type or class of the element

value - the value of the property

addProperty

```
public SoapObject addProperty(PropertyInfo pi, java.lang.Object value)
```

Adds a property (parameter) to the object. This is essentially a sub element.

Parameters:

name - the name of the property

type - the type or class of the element

value - the value of the property

Class SoapParser

public class SoapParser extends java.lang.Object

A SOAP parser. Limitations:

- Partial arrays are not yet supported
- Multi-Dimensional Arrays are not supported
- The hrefs must be local

Field Summary

org.kxml.parser.AbstractXmlParser **parser**

Constructor Summary

SoapParser(org.kxml.parser.AbstractXmlParser parser, classMap classMap)

Method Summary

java.lang.object **read()**
Extracts namespace and name and calls readBody for actual reading

java.lang.oblsct **read**(java.lang.Object owner, int index, java.lang.String namespace, java.lang String name, **ElementType** expected)

Builds an object from XML stream.

protected void **readSerializable**(**KvmSerializable** obj)

java.lang.Object **readUnknown**(java.lang.String namespace, java.lang.String name)

void **readVector**(java.util.Vector v, **ElementType** elementType)

Methods inherited from class java.lang.Object

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

Field Detail

parser

public org.kxml.parser.AbstractXmlParser parser

Constructor Detail

SoapParser

public SoapParser(org.kxml.parser.AbstractXmlParser parser, classMap classMap)

Method Detail

read

public java.lang.object read() throws java.io.IOException

Extracts namespace and name and calls readBody for actual reading

read
public java.lang.Object read(java.lang.Object owner,
int index,
java.lang.String namespace,
java.lang.String name,
ElementType expected)
throws java.io.IOException
Builds an object from the XML stream. This method is public for usage in conjunction with Marshal subclasses.

readSerializable
protected void readSerializable(KvmSerializable obj)
throws java.io.IOException

readUnknown
public java.lang.Object readUnknown(java.lang.String namespace,
java.lang.String name)
throws java.io.IOException

readVector
public void readVector(java.util.Vector v,
ElementType elementType)
throws java.io.IOException

Class SoapPrimitive

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

A class that is used to encapsulate primitive types (represented by a string in XML serialization). Basically, the SoapPrimitive class encapsulates "unknown" primitive types (similar to SoapObject encapsulating unknown complex types). For example, new SoapPrimitive (classMap.xsd, "float", "12.3") allows you to send a float from a MIDP device to a server although MIDP does not support floats. In the other direction, kSOAP will deserialize any primitive type (=no subelements) that are not recognized by the ClassMap to SoapPrimitive, preserving the namespace, name and string value (this is how the stockquote example works).

Constructor Summary

```
SoapPrimitive(java.lang.String namespace, java.lang.String name, java.lang.String value)
```

Method Summary

boolean	equals(java.lang.Object o)
java.lang.String	getName()
java.lang.String	getNamespace()
int	hashCode()
java.lang.String	toString()

Methods inherited from class java.lang.Object

clone, finalize, getClass, notify, notifyAll, wait

Constructor Detail

SoapPrimitive

```
public SoapPrimitive(java.lang.String namespace,  
                    java.lang.String name,  
                    java.lang.String value)
```

Method Detail

equals

```
public boolean equals(java.lang.Object o)
```

Overrides:

equals in class java.lang.Object

hashCode

```
public int hashCode()
```

Overrides:

hashCode in class java.lang.Object

toString

public java.lang.String toString()

Overrides:

toString in class java.lang.Object

getNamespace

public java.lang.String getNamespace()

getName

public java.lang.String getName()

class SoapWriter

public class SoapWriter extends java.lang.Object

A writer that is able to write objects wrt. the SOAP section five encoding rules.

Field Summary

org.kxml.io.AbstractXmlWriter **writer**

Constructor Summary

SoapWriter(org.kxml.io.AbstractXmlWriter writer, **classMap** classMap)

The SoapWriter is initialized with an AbstractXmlWriter and a class map.

Method Summary

void write(java.lang.Object obj)

Serializes the given object

void **writeObjectBody**(kvmSerializable obj)

Writes the body of an KvmSerializable object.

protected void **writeProperty**(java.lang.Object obj, **ElementType** type)

protected void **writeVectorBody**(java.util.Vector vector, **ElementType** elementType)

Methods inherited from class java.lang.Object

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

Field Detail

writer

public org.kxml.io.AbstractXmlWriter writer

Constructor Detail

SoapWriter

public SoapWriter(org.kxml.io.AbstractXmlWriter writer, ClassMap classMap)

The SoapWriter is initialized with an AbstractXmlWriter and a class map.

Method Detail

write

public void write(java.lang.Object obj)

throws java.io.IOException

Serializes the given object

writeObjectBody

public void writeObjectBody(KvmSerializable obj)

throws java.io.IOException

Writes the body of an KvmSerializable object. This method is public for access from Marshal subclasses.

writeProperty

protected void writeProperty(java.lang.Object obj, **ElementType** type)
throws java.io.IOException

WriteVectorBody

protected void writeVectorBody(java.util.Vector vector, **ElementType** elementType)
throws java.io.IOException

Exception Summary Class SoapFault

public class **SoapFault** extends java.io.IOException implements org.Kxml.XmlIO

Field Summary

java.util.vector	detail
java.lang.String	faultactor
java.lang.String	faultcode
java.lang.String	faultString

Constructor Summary

SoapFault()

Method Summary

void	parse(org.kxml.parser.AbstractionXmlparser parser)
java.lang.String	toString()
void	write(org.Kxml.io.AbbstractionXmlWriter xw)

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace

Methods inherited from class java.lang.Object

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

Field Detail

faultcode

public java.lang.String faultcode

faultString

public java.lang.String faultstring

faultactor

public java.lang.String faultactor

detail

public java.util.Vector detail

Method Detail

parse

public void parse(org.kxml.parser.AbstractXmlParser parser)
throws java.io.IOException

Specified by:

parse in interface org.kxml.XmlIO

write

```
public void write(org.kxml.io.AbstractXmlWriter xw)
    throws java.io.IOException
```

Specified by:

write in interface org.kxml.XmlIO

toString

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

Overrides:

toString in class java.lang.Throwable

Package org.ksoap.marshal

Class Summary

MarshalBase64	Base64(de)Serializer
MarshalDate	Marshal class for Dates
MarshalFloat	This marshal class is able to handle float, double and decimal
MarshalHashtable	serializes instances of hashtable to and from xm

Package org.Ksopa.marshal Description

This package contains some (optional) implementations of the Marshal interface

Class MarshalBase64

public class MarshalBase64 extends java.lang.Object implements MarshalBase64 (de)serializer

Field Summary

static java.lang.class **Byte_Array_Class**

Constructor Summary

MarshalBase64()

Method Summary

java.lang.object readInstance(SoapParser parser, java.lang.String namespace, java.lang.String name, **ElementType** expected)

This methods reads an instance from the given parser.

void register(ClassMap cm)
Register this Marshal with the given classMap

void writeInstance(SoapWriter writer, java.lang.object obj)
Write the instance to the given SoapWriter.

Methods inherited from class java.lang.Object

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

Field Detail

BYTE_ARRAY_CLASS

public static java.lang.Class BYTE_ARRAY_CLASS

Constructor Detail

MarshalBase64

public MarshalBase64()

Method Detail

readInstance

```
public java.lang.Object readInstance(SoapParser parser,  
                                     java.lang.String namespace,  
                                     java.lang.String name,  
                                     ElementType expected)  
    throws java.io.IOException
```

Description copied from interface: Marshal

This methods reads an instance from the given parser. For implementation, please note that the start and end tag must be consumed. This is not symmetric to writeInstance, but otherwise it would not be possible to access the attributes of the start tag here. The underlying XML parser is accessible using parser.parser.

Specified by:

readInstance in interface **Marshal**

writeInstance

```
public void writeInstance(SoapWriter writer,  
                          java.lang.Object obj)  
    throws java.io.IOException
```

Description copied from interface: Marshal

Write the instance to the given SoapWriter. In contrast to readInstance, it is not necessary to care about the surrounding start and end tags. Additional attributes must be written before anything else is written. The underlying xml writer is accessible using writer.writer.

Specified by:

writeInstance in interface **Marshal**

register

```
public void register(ClassMap cm)
```

Description copied from interface: Marshal

Register this Marshal with the given classMap

Specified by:

register in interface **Marshal**

Class MarshalDate

public class MarshalDate extends java.lang.Object implements Marshal
Marshal class for Dates.

Field Summary

static java.lang.Class DATE_CLASS

Constructor Summary

MarshalDate()

Method Summary

java.lang.Object readInstance(SoapParser, java.lang.String namespace, java.lang.String
name, **ElementType** expected)

This methods reads an instance from the given parser.

void register(ClassMap cm)

void writeInstance(SoapWriter writer, java.lang.Object obj)
Write the instance to the given SoapWriter.

Methods inherited from class java.lang.Object

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

Field Detail

Date_Class

public static java.lang.class DATE_CLASS

Constructor Detail

MarshalDate

public MarshalDate()

Method Detail

readInstance

public java.lang.Object readInstance(**SoapParser** parser,
 java.lang.String namespace,
 java.lang.String name,
 ElementType expected)
 throws java.io.IOException

Description copied from interface: Marshal

This methods reads an instance from the given parser. For implementation, please note that the start and end tag must be consumed. This is not symmetric to writeInstance, but otherwise it would not be possible to access the attributes of the start tag here. The underlying XML parser is accessible using parser.parser.

Specified by:

readInstance in interface Marshal

writeInstance

```
public void writeInstance(SoapWriter writer,  
                        java.lang.Object obj)  
    throws java.io.IOException
```

Description copied from interface: Marshal

Write the instance to the given SoapWriter. In contrast to readInstance, it is not necessary to care about the surrounding start and end tags. Additional attributes must be written before anything else is written. The underlying xml writer is accessible using writer.writer.

Specified by:

writeInstance in interface Marshal

register

```
public void register(ClassMap cm)
```

Description copied from interface: Marshal

Register this Marshal with the given classMap

Specified by:

register in interface Marshal

Class MarshalFloat

public class MarshalFloat extends java.lang.Object implements Marshal

This marshal class is able to handle float, double and decimal. Since CLDC does not contain support for floating point arithmetics, it will work only with J2SE

Constructor Summary

MarshalFloat()

Method Summary

java.lang.Object readInstance(SoapParser parser, java.lang.String namespace, java.lang.String name, **ElementType** expected)

This methods reads an instance from the given parser.

void **register(ClassMap cm)**
Register this Marshal with the given classMap

void **writeInstance(SoapWriter writer, java.lang.Object instance)**
Write the instance to the given SoapWriter.

Methods inherited from class java.lang.Object

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

Constructor Detail

MarshalFloat

public MarshalFloat()

Method Detail

readInstance

public java.lang.Object readInstance(**SoapParser** parser,
java.lang.String namespace,
java.lang.String name,
ElementType expected)
throws java.io.IOException

Description copied from interface: Marshal

This methods reads an instance from the given parser. For implementation, please note that the start and end tag must be consumed. This is not symmetric to writeInstance, but otherwise it would not be possible to access the attributes of the start tag here. The underlying XML parser is accessible using parser.parser.

Specified by:

readInstance in interface **Marshal**

register

public void register(Class cm)

Description copied from interface: Marshal
Register this Maeshal with the given classMap

Specified by:

register in interface **Marshal**

Class MarshalHashtable

public class MarshalHashtable extends java.lang.Object implements Marshal

serializes instances of hashtable to and from xml. this implementation is based on the xml schema from apache-soap, namely the type 'map' in the namespace 'http://xml.apache.org/xml-soap'. other soap implementations including apache (obviously) and glue are also interoperable with the schema.

Field Summary

static java.lang.class HASHTABLE_CLASS
CLDC does not support .class, so this helper is needed

static java.lang.String NAME
use then during registraration

static java.lang.String NAMESPACE
use then during registration

Constructor Summary

MarshalHastable()

Method Summary

java.lang.object readInstance(SoapParser parser, java.lang.String namespace,
java.lang.String name, **ElementType** expected)

This methods reads an instance from the given parser.

void **register(ClassMap cm)**
Register this Marshal with the given classMap

void **writeInstance(SoapWriter writer, java.lang.Object instance)**
Write the instance to the given SoapWriter.

Methods inherited from class java.lang.Object

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

Field Detail

NAMESPACE

public static final java.lang.String NAMESPACE

NAME

public static final java.lang.String NAME
use then during registration

HASHTABLE_CLASS

public static final java.lang.Class HASHTABLE_CLASS

CLDC does not support .class, so this helper is needed.

Constructor Detail

MarshalHashtable

```
public MarshalHashtable()
```

Method Detail

readInstance

```
public java.lang.Object readInstance(SoapParser parser,  
                                     java.lang.String namespace,  
                                     java.lang.String name,  
                                     ElementType expected)  
    throws java.io.IOException
```

Description copied from interface: Marshal

This methods reads an instance from the given parser. For implementation, please note that the start and end tag must be consumed. This is not symmetric to writeInstance, but otherwise it would not be possible to access the attributes of the start tag here. The underlying XML parser is accessible using parser.parser.

Specified by:

readInstance in interface **Marshal**

writeInstance

```
public void writeInstance(SoapWriter writer,  
                          java.lang.Object instance)  
    throws java.io.IOException
```

Description copied from interface: Marshal

Write the instance to the given SoapWriter. In contrast to readInstance, it is not necessary to care about the surrounding start and end tags. Additional attributes must be written before anything else is written. The underlying xml writer is accessible using writer.writer.

Specified by:

writeInstance in interface **Marshal**

register

```
public void register(ClassMap cm)
```

Description copied from interface: Marshal

Register this Marshal with the given classMap

Specified by:

register in interface **Marshal**

Package org.Ksoap.servlet

Class Summary

SoapServlet

copy-paste seans interop server orb here as needed....

Class SoapServlet

public class SoapServlet extends javax.servlet.http.HttpServlet

copy-paste seans interop server orb here as needed.... does not run compile yet. checked in just for simplified access some design issues: - path and soapaction are not considered. soapaction is deprecated; for multiple paths, please use multiple servlets.

Constructor Summary

SoapServlet()

Method Summary

void **doPost**(javax.servlet.http.HttpServletRequest req, javax.servlet.http.HttpServletResponse res)

classMap **getClassMap**()

protected **getInstance**(javax.servlet.http.HttpServletRequest request)
java.lang.Object the default operation is to map request.getPathInfo to an instance using the information given by buildinstance.

void **publishClass**(java.lang.class service, java.lang.String namespace)
Publish all public methods of the given class.

void **publishInstance**(java.lang.String path, java.lang.Object instance)
publish an instance by associating the instance with the given local path.

void **publishMethod**(java.lang.Class service, java.lang.String namespace, java.lang.String name, java.lang.String[] parameterNames)
Convenience method; use this method if the parameter types can be obtained via reflection.

void **setClassMap**(ClassMap classMap)
Please note: The classMap should not be set after publishing methods, because parameter type information may get lost!

Methods inherited from class javax.servlet.http.HttpServlet

doDelete, doGet, doHead, doOptions, doPut, doTrace, getLastModified, service

Methods inherited from class javax.servlet.GenericServlet

destroy, getInitParameter, getInitParameterNames, getServletConfig, getServletContext, getServletInfo, getServletName, init, log

Methods inherited from class java.lang.Object

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

Constructor Detail

SoapServlet

Method Detail

getInstance

protected java.lang.Object getInstance(javax.servlet.http.HttpServletRequest request)
the default operation is to map request.getPathInfo to an instance using the information given by buildInstance. The returned instance is used as target object for the method invocation. Please overwrite this method in order to define your own (generic) mapping. If no mapping is found, the servlet itself is returned.

publishClass

public void publishClass(java.lang.Class service, java.lang.String namespace)
Publish all public methods of the given class

publishInstance

public void publishInstance(java.lang.String path, java.lang.Object instance)

publish an instance by associating the instance with the given local path. Please note that (currently) also the methods need to be published separately. Alternatively to this call, it is also possible to overwrite the getObject (HttpServletRequest request) method

publishMethod

public void publishMethod(java.lang.Class service,
java.lang.String namespace,
java.lang.String name,
PropertyInfo[] parameters)

publish a method. Please note that also a corresponding instance needs to be published, either calling publishInstance or by overwriting getInstance (), except when the method is a method of the servlet itself.

publishMethod

public void publishMethod(java.lang.Class service,
java.lang.String namespace,
java.lang.String name,
java.lang.String[] parameterNames)

convenience method; use this method if the parameter types can be obtained via reflection

getClassMap

```
public ClassMap getClassMap()  
setClassMap
```

```
public void setClassMap(ClassMap classMap)
```

Please note: The classMap should not be set after publishing methods, because parameter type information may get lost!

doPost

```
public void doPost(javax.servlet.http.HttpServletRequest req,  
                  javax.servlet.http.HttpServletResponse res)  
    throws javax.servlet.ServletException,  
           java.io.IOException
```

Overrides:

doPost in class javax.servlet.http.HttpServlet

Package org.Ksoap.transport

This package org.Ksoap.transport classes for Ksoap

Class Summary

HttpTransport	Methods to facilitate SOAP calls over HTTP using the J2ME generic connexion framework.
HttpTransportSE	An optional HTTP transport class for the desktop version of Java, suitable

Package org.Ksoap.transport Description

This package contains transport classes for kSOAP

Class HttpTransport

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

Methods to facilitate SOAP calls over HTTP using the J2ME generic connection framework.

Instances of HttpTransport can be in one of two states: connected and not connected. When an invocation on call is made the instance is in a connected state until call returns or throws an IOException. in any case once control is returned to the caller the instance is again in the not connected state. HttpTransport is not thread safe and applications should ensure that only one thread is inside the call method at any given time. It is designed in such a way that applications can reuse a single instance for all soap calls to one, or multiple, target endpoints.

The underlying HttpURLConnection is opened with the timeout flag set. In the MIDP API this flag is only a hint to the underlying protocol handler to throw an InterruptedException, however, there are no guarantees that it will be handled. So rather than support a timeout mechanism internally the design is such that applications can manage timeouts in an environment dependent way.

For example some environments may allow for a timeout parameter that can be externally specified in perhaps a system property (which? I don't know. it's in the api). Others like the emulator (ok, who cares) and the Motorola i85s can use a simple and effective timeout mechanism that closes the connection and associated streams in an asynchronous fashion. Calling the close() method inside of a separate thread can provide for this timeout handling by releasing threads that maybe stuck inside of call() performing network io.

Here is some sample code to demonstrate how such a timeout mechanism may look:

```
private HttpTransport soap;  
...  
TimerTask task =  
new TimerTask( ) { public void run( ) { soap.close( ); } };
```

```

try {
new Timer( ).schedule( task, TIMEOUT );
soap.call( soapobject ); // invoke method
task.cancel( );          // cancel the timeout

} catch ( InterruptedException e ) {
// handle timeout here...

} catch ( IOException e ) {
// some other io problem...
}

```

The call() method will throw an InterruptedException if the instance is no longer in the connected state before control is returned to the caller. The call to soap.close() inside the TimerTask transitions the HttpConnection into a not connected state.

Note: The InterruptedException will be caught by a thread waiting on network io, however, it may not be immediate. It is assumed that the protocol handler will gracefully handle the lifecycle of the outputstream and therefore it is not closed inside the close method. IOW the waiting thread will be interrupted after the outputstream has been flushed. If the waiting thread is hung up waiting for input a call to close from a separate thread the exception is observed right away and will return before the thread calling close. At least this is what has been observed on the i85s handset. On this device, if a call to outputstream.close() is made while the outputstream is being flushed it seems to cause a deadlock, ie outputstream will never return.

Field Summary

boolean	debug	set to true if debugging
java.lang.String	requestDump	String dump of request for debugging.
java.lang.String	responseDump	String dump of response for debugging.

Constructor Summary

HttpTransport()

HttpTransport(java.lang.String url, java.lang.String soapAction)
Creates instance of HttpTransport with set url and SoapAction

Method Summary

void call()
Sends the requestEnvelope and fills the responseEnvelope

java.lang.Object call(SoapObject method)
Executes a SOAP Method and returns a response

void call(org.kxml.XmlIO request, org.Kxml.XmlIO result)

void reset()
Closes the connexion and associated streams.

void setClassMap(ClassMap classMap)
set the desired ClassMap for the SOAP Envelopes

void setSoapAction(java.lang.String soapAction)
set the desired soapAction header field

void setUrl(java.lang.String url)
Set the target url

Methods inherited from class java.lang.Object

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

Field Detail

debug

public boolean debug

set to true if debugging

requestDump

public java.lang.String requestDump
String dump of request for debugging.

responseDump

public java.lang.String requestDump
String dump of request for debugging.

responseDump

public java.lang.String responseDump
String dump for debugging

Constructor Detail

HttpTransport

public HttpTransport()
default constructor

HttpTransport

public HttpTransport(java.lang.String url, java.lang.String soapAction)
Creates instance of HttpTransport with set url and SoapAction

Parameters:

url - the destination to POST SOAP data

soapAction - the desired SOAP action (for HTTP headers)

Method Detail

setUrl

public void setUrl(java.lang.String url)

Set the target url.

Parameters:

url - the target url.

setClassMap

public void setClassMap(ClassMap classMap)

set the desired ClassMap for the SOAP Envelopes

Parameters:

classMap - the desired ClassMap

Call

public void call()

throws java.io.IOException

Sends the requestEnvelope and fills the responseEnvelope

Throws:

InterruptedException - if transport was closed async.

java.io.IOException - if an error occurs

call

public void call(org.kxml.XmlIO request,

org.kxml.XmlIO result)

throws java.io.IOException

reset

public void reset()

Closes the connection and associated streams. This method does not need to be explicitly called since the underlying connections and streams are only opened and valid inside of the call method. Close can be called asynchronously, from another thread to potentially release another thread that is hung up doing network io inside of call. Caution should be taken, however when using this as a pseudo timeout mechanism. it is a valid and suggested approach for the motorola handsets. oh, and it works in the emulator...

Class HttpTransportSE

public class HttpTransportSE extends java.lang.Object

An optional HTTP transport class for the desktop version of jacva, suitable eg. for applets.

Field Summary

boolean	debug
java.lang.String	requestDump
java.lang.String	responseDump

Constructor Summary

HttpTransportSE()

HttpTransportSE(java.lang.String url, java.lang.String soapAction)

Method Summary

void	call()	Sends the requestEnvelope and fills the responseEnvelope
java.lang.Object	call(SoapObject method)	
void	set(ClassMap classMap)	
void	setSoapAction(java.lang.String soapAction)	
void	setUrl(java.lang.String url)	

Methods inherited from class java.lang.Object

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

Field Detail

debug

public boolean debug

requestDump

public java.lang.String requestDump

responseDump

public java.lang.String responseDump

Constructor Detail

HttpTransportSE

public HttpTransportSE()

HttpTransportSE

public HttpTransportSE()

HttpTransportSE

public HttpTransportSE(java.lang.String url, java.lang.String soapAction)

Method Detail

setUrl

public void setSoapAction(java.lang.String soapAction)

setClassMap

public void setClassMap(ClassMap classMap)

call

public void call()
throws java.io.IOException
Sends the requestEnvelope and fills the responseEnvelope

call

public java.lang.Object call(SoapObject method)
throws java.io.IOException

