

K-soap Up-close

The principle aim of this article is to understand the workings of Ksoap i.e what is K-soap, how to use it etc.

So what is K-soap ?

K-soap is basically a subset of SOAP, web services use SOAP to communicate but when you are using web services on smaller devices with a much greater memory limitations it is preferable to use a subset of regular SOAP which can provide the same basic functions but with a small footprint. K-SOAP and k-xml are the result of this.

The Ksoap project is based on k-xml, a lightweight pull parser designed for use with MIDP. While KSoap doesn't support WSDL, it does name calling .Only two objects are required: the SOAPObject and http-transport. In addition, KSoap makes it very easy to capture fault data. The tool kit maps all **SOAP faults to an exception object known as SoapFault**. In this manner, a fault can be caught and handled like any other exception in the MIDlet.

NOTE: Need to look up MIDP and Ksoap MIDlet suite and MIDlet

MIDP

The Mobile Information Device Profile (MIDP), combined with the Connected Limited Device Configuration (CLDC), is the Java runtime environment for today's mobile information devices (MIDs) such as phones and entry level PDAs. What MIDP provides is the core application functionality required by mobile applications - including the user interface, network connectivity, local data storage, and application lifecycle management - packaged as a standardized Java runtime environment and set of Java APIs.

The Ksoap API

The Api is quite small in comparison to other similar SOAP api's and contains just five distinct packages.

Package org.Kobjects.serialization

Package org Ksoap

Package org.Ksoap.servlet

Package org.Ksoap.transport

Package org.ksoap.marshal

Package org.Kobjects.serialization

The first package the Kobjects.serialization package contains just one interface and two classes. The interface KvmSerializable provides get and set methods for properties . The first class in this package is Class Element type it's primary function is to encapsulate type information. The second class is Class propertyInfo this class is used to store information about each property an implementation of KvmSerializable exposes.

Package org.Ksoap

This is probably the most important of the packages it provides all the classes required for SOAP (de)serialization.It contains one interface and seven classes. The Marshal interface is basically an Abstract class for custom (de)serialiazation. The first of the classes is Class ClassMap this class provides various soap properties relevant for (de)serialization, including a method for defining mappings between java classes and XML element names. The next class Class Soap contains come constant definitions only. Class SoapEnvelope provides all the methods needed to create a soap envelope. The next class SoapObject is a simple dynamic object that can be used to build

soap calls without implementing KvmSerializable. Class SoapParser is as it say's a SOAP parser. Class SoapPrimitive is a class that is used to encapsulate primitive types (represented by a string in XML serialization) and finally class SoapWriter is a writer taht is able to write objects wrt.

This package also includes a Class SoapFault, the toolkit maps all SOAP faults to an exception object known as SoapFault. In this manner a fault can be caught and handled like anyother exception in the MIDlet.

The next class is slightly unusual in that it contains some optional implementations of the Marshal interface xtendedClassMap(SE).java this class is deprecated, but still available from CVS. The **MarshalBase64.java** is a Marshal implementation that is able to handle base 64 encoded byte arrays. It relies on org.kobjects.base64.Base64, available from kobjects.org. **The MarshalDate.java** is a Marshal implementation that is able to handle dates. It relies on org.kobjects.isodate.IsoDate, available from kobjects.org. Please note that the classes belong to different packages.The **MarshalHashtable.java** is Marshal implementation that is able to handle hashtables. and the **MarshalFloat** class is able to handle float, double and decimal.

The package org Ksoap.opt is now deprecated the two classes Class ExtendedClass Map has been deprecated by using org.ksoap.marshal.MarshalDate and MarshalBase64 instead and the class ExtendedClassMapSE has been deprecated by org.ksoap.MarshalFloat instead.

The package org.Ksoap.servlet contains just one class SoapServlet.

The final package is one of the most important org.Ksoap.transport it contains just two Classes Class httpTransport and httpTransportSE. The former provides methods to facilitate SOAP calls over HTTP using the J2ME generic connexion framework while the Class Http TransportSE is an optional HTTP transport class for the dextop version of java.

From the above it can be clearly seen that the APi is much more compact and easier to digest than for instance the Apache axis API and the benefits of having such a smaller more compact version of SOAP can be instantly seen.