

## Ejecución de WebServices desde Android

Vamos a utilizar el protocolo SOAP para enviar y recuperar información con el fin de llamar a los métodos web.

Arquitectura de una aplicación usando WebServices:



## ¿Qué es SOAP?

SOAP es una especificación de protocolo para el intercambio de información estructurada en la aplicación de servicios web en redes informáticas. Se basa en Extensible Markup Language (XML) para su formato de mensaje, y por lo general se basa en otros protocolos de capa de aplicación, sobre todo de transferencia de hipertexto (HTTP) y Protocolo simple de transferencia de correo (SMTP), por mensaje de negociación y transmisión.

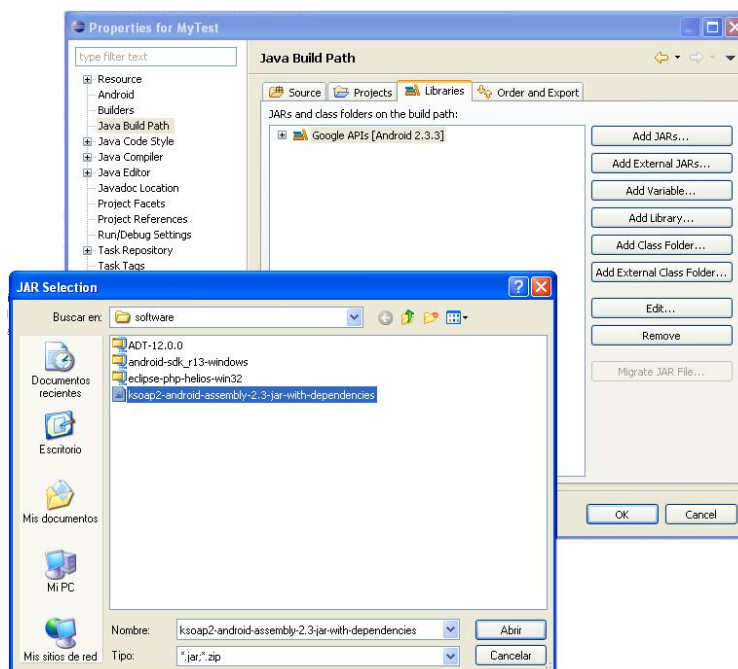
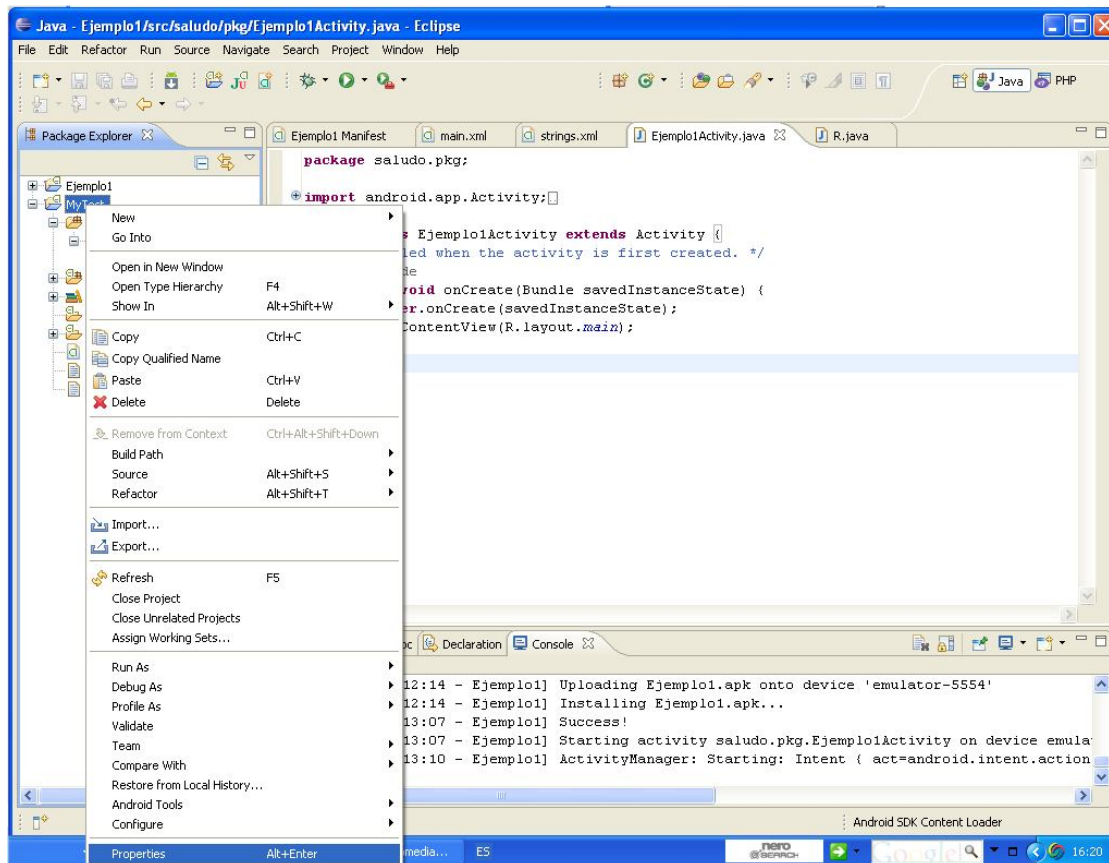
## KSOAP

KSOAP2 es una librería ligera y eficiente, que facilita la interacción con WebServices desde Android



## INSTITUTO PERUANO DE CIENCIAS DE LA INFORMACION

Av Diez Canseco 442 – Piso 9  
Edificio El Ejecutivo – Miraflores  
[www.LibrosDigitales.Net](http://www.LibrosDigitales.Net)





## INSTITUTO PERUANO DE CIENCIAS DE LA INFORMACION

Av Diez Canseco 442 – Piso 9  
Edificio El Ejecutivo – Miraflores  
[www.LibrosDigitales.Net](http://www.LibrosDigitales.Net)

### ¿Qué vamos a desarrollar?

Vamos a desarrollar una sencilla aplicación para demostrar cómo se invoca el servicio web utilizando el protocolo SOAP en la aplicación Android. La Aplicación ha desarrollar hará la conversión de grados celcius a Fahrenheit.

### Ubicación de la WebService:

<http://www.w3schools.com/webservices/tempconvert.asmx>

**TempConvert Web Service - Mozilla Firefox**

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://www.w3schools.com/webservices/tempconvert.asmx

Más visitados Comenzar a usar Firef... Últimas noticias HowToUse - ksoap2-a...

ksoap2-android-assembly-2.3-jar-with-... TempConvert Web Service

## TempConvert

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [CelsiusToFahrenheit](#)
- [FahrenheitToCelsius](#)

**This web service is using <http://tempuri.org/> as its default namespace.**

**Recommendation: Change the default namespace before the XML Web service is made public.**

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other service but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```



## INSTITUTO PERUANO DE CIENCIAS DE LA INFORMACION

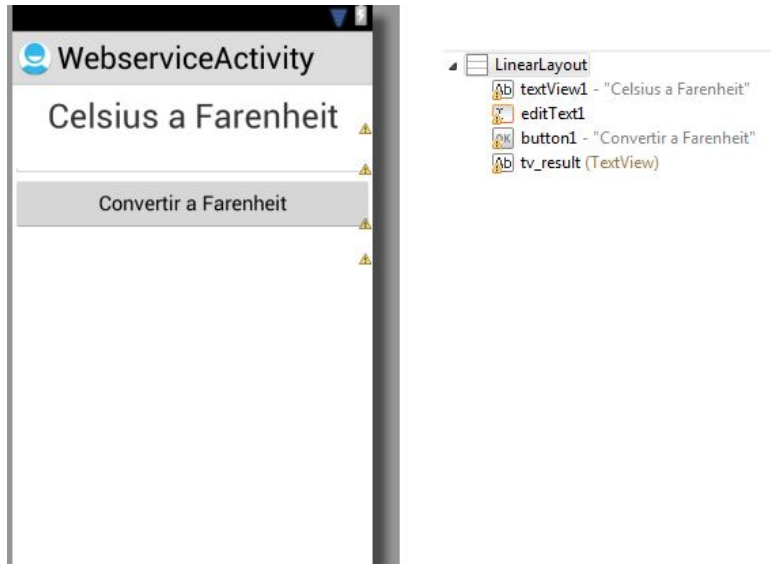
Av Diez Canseco 442 – Piso 9  
Edificio El Ejecutivo – Miraflores  
[www.LibrosDigitales.Net](http://www.LibrosDigitales.Net)

---

### Creación del Proyecto:

Crear un proyecto nuevo android WebserviceTemp

### Diseño del Layout



### En MainActivity.java defina:

```
package com.webservice.android;

import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.PropertyInfo;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapPrimitive;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

import com.example.webserviceactivity.R;

import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
```



## INSTITUTO PERUANO DE CIENCIAS DE LA INFORMACION

Av Diez Canseco 442 – Piso 9  
Edificio El Ejecutivo – Miraflores  
[www.LibrosDigitales.Net](http://www.LibrosDigitales.Net)

---

```
public class MainActivity extends Activity {

    private final String NAMESPACE = "http://tempuri.org/";
    private final String URL =
"http://www.w3schools.com/webservices/tempconvert.asmx";
    private final String SOAP_ACTION = "http://tempuri.org/CelsiusToFahrenheit";
    private final String METHOD_NAME = "CelsiusToFahrenheit";
    private String TAG = "MyWS";

    private static String celcius;
    private static String fahren;
    Button b;
    TextView tv;
    EditText et;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        //Relacionamos los controles con las variables
        et = (EditText) findViewById(R.id.editText1);
        tv = (TextView) findViewById(R.id.tv_result);
        b = (Button) findViewById(R.id.button1);

        //Definimos el evento click
        b.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                //Check if Celcius text control is not empty
                if (et.getText().length() != 0 && et.getText().toString() != "") {
                    //Get the text control value
                    celcius = et.getText().toString();
                    //Create instance for AsyncCallWS
                    AsyncCallWS task = new AsyncCallWS();
                    //Call execute
                    task.execute();
                }
                //If text control is empty
            } else {
                tv.setText("Please enter Celcius");
            }
        });
    }
}
```



## INSTITUTO PERUANO DE CIENCIAS DE LA INFORMACION

Av Diez Canseco 442 – Piso 9  
Edificio El Ejecutivo – Miraflores  
[www.LibrosDigitales.Net](http://www.LibrosDigitales.Net)

---

```
public void getFahrenheit(String celsius) {
    // Creamos el objeto SOAP
    SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);

    // Creamos el objeto que nos permitirá transferir los valores
    PropertyInfo celsiusPI = new PropertyInfo();

    //Set Name
    celsiusPI.setName("Celsius");

    //Set Value
    celsiusPI.setValue(celsius);

    //Set dataType
    celsiusPI.setType(double.class);

    //Asociamos lo anterior al objeto request
    request.addProperty(celsiusPI);

    //Creamos el objeto envelope
    SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(
        SoapEnvelope.VER11);
    envelope.dotNet = true;

    //Definimos la respuesta
    envelope.setOutputSoapObject(request);

    //Creamos el objeto HTTP
    HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);

    try {
        //Invoca a la web service
        androidHttpTransport.call(SOAP_ACTION, envelope);

        //Obtiene la respuesta
        SoapPrimitive response = (SoapPrimitive) envelope.getResponse();

        //Asigna la respuesta a una variable
        fahren = response.toString();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



## INSTITUTO PERUANO DE CIENCIAS DE LA INFORMACION

Av Diez Canseco 442 – Piso 9  
Edificio El Ejecutivo – Miraflores  
[www.LibrosDigitales.Net](http://www.LibrosDigitales.Net)

---

```
private class AsyncCallWS extends AsyncTask<String, Void, Void> {
    @Override
    protected Void doInBackground(String... params) {
        Log.i(TAG, "doInBackground");
        getFahrenheit(celcius);
        return null;
    }

    @Override
    protected void onPostExecute(Void result) {
        Log.i(TAG, "onPostExecute");
        tv.setText(fahren + "° F");
    }

    @Override
    protected void onPreExecute() {
        Log.i(TAG, "onPreExecute");
        tv.setText("Calculating...");
    }

    @Override
    protected void onProgressUpdate(Void... values) {
        Log.i(TAG, "onProgressUpdate");
    }
}
```

### Definimos los permisos de Internet

No se olvide de agregar permisos a Internet en AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET" />
```

### Ejecutamos