**WebServices**

**1. Tworzymy nowy projekt.**

**2. Zmieniamy layout:**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

    android:layout\_width="match\_parent"

    android:layout\_height="wrap\_content"

    android:orientation="vertical" >

    <TextView

        android:id="@+id/textView1"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_marginTop="10dp"

        android:gravity="center"

        android:text="Celsius to Farenheit"

        android:textSize="30dp" />

    <EditText

        android:id="@+id/editText1"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:gravity="center"

        android:numeric="integer"

        android:singleLine="true" />

    <Button

        android:id="@+id/button1"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_gravity="center\_horizontal"

        android:gravity="center"

        android:text="Convert to Farenheit" />

    <TextView

        android:id="@+id/tv\_result"

        android:layout\_width="fill\_parent"

        android:layout\_height="wrap\_content"

        android:gravity="center"

        android:text="" android:textSize="26dp"/>

</LinearLayout>

**3. Dodajemy globalne zmienne w głównej klasie MainActivity**

private final String NAMESPACE = "<http://www.w3schools.com/webservices/>";

private final String URL = "<http://www.w3schools.com/webservices/tempconvert.asmx>";

private final String SOAP\_ACTION = "<http://www.w3schools.com/webservices/CelsiusToFahrenheit>";

private final String METHOD\_NAME = "CelsiusToFahrenheit";

private String TAG = "Przyklad";

private static String celcius;

private static String fahren;

Button b;

TextView tv;

EditText et;

**4. Zmienić funkcję OnCreate:**

**public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **et** = (EditText) findViewById(R.id.***editText1***);  
 **tv** = (TextView) findViewById(R.id.***tv\_result***);  
 **b** = (Button) findViewById(R.id.***button1***);  
 **b**.setOnClickListener(**new** OnClickListener() {  
 **public void** onClick(View v) {  
 **if** (**et**.getText().length() != 0 && **et**.getText().toString() != **""**) {  
  *celcius* = **et**.getText().toString();  
 AsyncCallWS task = **new** AsyncCallWS();  
 task.execute();  
 } **else** {  
 **tv**.setText(**"Please enter Celcius"**);  
 }  
 }  
 });  
}

**5. Stworzyć wewnętrzną klasę w głównej klasie programu:**

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");

        }

    }

**6. Dodać funkcję do głównej klasy:**

public void getFahrenheit(String celsius) {

        SoapObject request = new SoapObject(NAMESPACE, METHOD\_NAME);

        PropertyInfo celsiusPI = new PropertyInfo();

    celsiusPI.setName("Celsius");

    celsiusPI.setValue(celsius);

    celsiusPI.setType(double.class);

        request.addProperty(celsiusPI);

    SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(

            SoapEnvelope.VER11);

    envelope.dotNet = true;

    envelope.setOutputSoapObject(request);

        HttpTransportSE androidHttpTransport = new HttpTransportSE(URL);

    try {

                androidHttpTransport.call(SOAP\_ACTION, envelope);

        SoapPrimitive response = (SoapPrimitive) envelope.getResponse();

        fahren = response.toString();

    } catch (Exception e) {

        e.printStackTrace();

    }

}

**7. Dodać pozwolenie na połączenie w AndroidManifest.xml:**

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

**8. Ewentualnie mogą być potrzebne dodatkowe importy:**

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;

**9. Plik ksoap2-android-assembly-2.4-jar-with-dependencies.jar należy umieścić w katalogu libs, w projekcie należy kliknąć na nim prawym przyciskiem i wybrać opcję Add as library**

**Zadania**

**1. Zmodyfikować program, aby możliwe było przeliczanie w drugą stronę**

**2. Znaleźć inny webservice i spróbować go użyć**