

实验九：网络服务

13331388 庄梓嘉

实验目的

- 1.学习使用 KSOAP2 访问 Web Service ；
- 2.了解 SOAP 协议基础知识 ；

实验内容

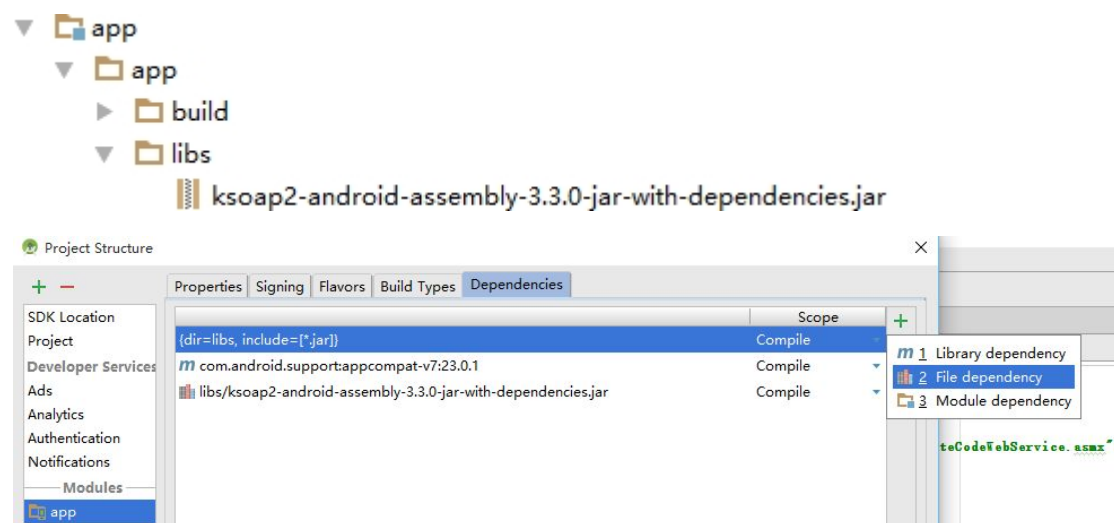
使用 Ksoap2 , 发送文字信息到指定网络 service 获得生成的验证码图片 , 并显示出来。

【拓展项】

打开应用界面随机生成一张验证码 , 输入对应验证码的信息后 , 匹配是否通过。

实验步骤

(1) 将 SOAP 开发包 Ksoap2android 导入到项目 libs 文件夹中 , 将其添加到依赖项 , File->Project Structure->app->Dependencies->File dependency



(2) Download 是实现 Runnable 接口的一个类 ,用来实现访问 webService 获取验证码图片字节流。指定 WebService 的命名空间和调用方法名 , 设置调用 WebService 接口需要传入的参数 ,生成调用 WebService 方法的 SOAP 请求信息 , 并指定 SOAP 版本 ,设置是否调用的是 dtNet 开发的 WebService。

```
public class Download implements Runnable {
    private final String NAMESPACE = "http://WebXml.com.cn/";
    private final String METHODNAME = "enValidateByte";
    private final String SOAPACTION = "http://WebXml.com.cn/enValidateByte";
    private final String URL = "http://webservice.webxml.com.cn/WebServices/ValidateCodeWebService.asmx";
    public Download() {}
    public void run() {
        SoapObject request = new SoapObject(NAMESPACE, METHODNAME);
        String str = createText.getText().toString();
        Log.e("str", str);
        request.addProperty("byString", str);
        SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VERSION10);
        envelope.dotNet = true;
        envelope.setOutputSoapObject(request);
        HttpTransportSE transportSE = new HttpTransportSE(URL);
        try {
            transportSE.call(SOAPACTION, envelope);
        } catch (Exception e) {
            e.printStackTrace();
        }
        Log.e("fault", envelope.bodyIn.toString());
        SoapObject result = (SoapObject)envelope.bodyIn;
        SoapPrimitive detail = (SoapPrimitive)result.getProperty("enValidateByteResult");
        Message msg = new Message();
        msg.obj = detail;
        msg.what = UPDATE;
        handler.sendMessage(msg);
    }
}
```

(3) 使用 handler 来更新 UI 界面。

```
private static final int UPDATE = 0;
private Handler handler = new Handler() {
    public void handleMessage(Message message) {
        switch (message.what) {
            case UPDATE:
                progressDialog.dismiss();
                byte[] data = Base64.decode((message.obj.toString()).getBytes(), Base64.DEFAULT);
                Bitmap bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);
                Matrix matrix = new Matrix();
                matrix.postScale(10, 10);
                Bitmap newbit = Bitmap.createBitmap(bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight(), matrix, true);
                BitmapDrawable drawable = new BitmapDrawable(newbit);
                imageView.setImageDrawable(drawable);
                break;
            default:
                break;
        }
        super.handleMessage(message);
    }
};
```

(4) 在 AndroidManifest 文件中添加访问网络的权限。

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

(5) 在访问 Webservice 之前，显示 ProgressDialog，在返回结果后，隐藏 ProgressDialog。

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        progressDialog = ProgressDialog.show(MainActivity.this, "Requesting...", "Please requesting...", true, false);  
        Download download = new Download();  
        new Thread(download).start();  
    }  
});
```

【拓展项】

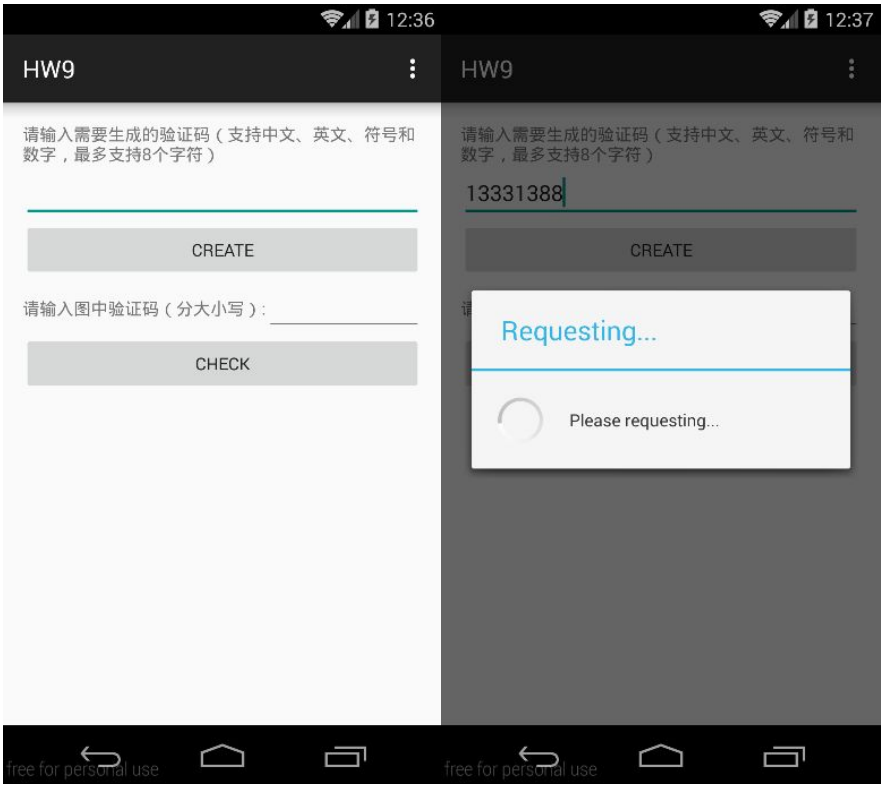
添加检查验证码对错，直接比较输入的字符串是否相等，所以区分大小写。

```
button1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if (checkText.getText().toString().equals(createText.getText().toString())) {  
            Toast.makeText(MainActivity.this, "正确!", Toast.LENGTH_SHORT).show();  
        } else {  
            Toast.makeText(MainActivity.this, "错误! 请重新输入", Toast.LENGTH_SHORT).show();  
            checkText.setText("");  
        }  
    }  
});
```

实验结果

初始页面

查询中

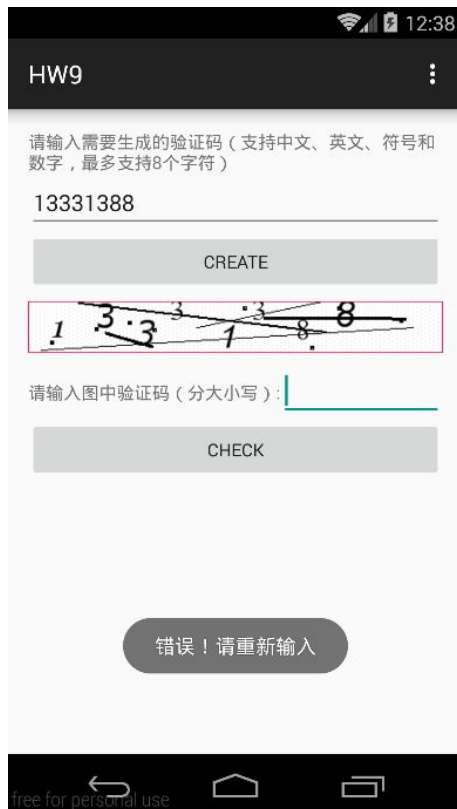


返回结果

检查验证码：正确



检查验证码：错误



参考资料

Android 开发之 WebService 介绍：

<http://blog.csdn.net/lyq8479/article/details/6420398>

Android 平台调用 WebService 详解：

<http://blog.csdn.net/lyq8479/article/details/6428288>

Android 生成验证码 bitmap：

<http://blog.csdn.net/whumr1/article/details/7092013>

Android 验证码的生成与验证：

<http://blog.csdn.net/jdsjlzx/article/details/39481337>

实验总结

这次实验要求使用验证码图的网络服务,学会使用 KSOAP2 以及了解 SOAP 的基本协议。Web Service 这一块内容之前作业做也有使用到,但是这次作业是使用到 SOAP 来进行信息传输。将验证码传输到网络上,再返回验证码图片。只能简单的依据 PDF 中写好的代码来使用,要熟练地自主使用 SOAP 的话,还需要更多地练习。检查验证码有找到相关的博客参考,但是由于时间不够,就直接对比之前的验证码,所以是区分大小写的,可以再加个将字符串全转化成大写再对比,就可以不区分大小写了,简单略过。了解到验证码图的生成过程,以及验证过程并没有那么复杂,生活中到处都看得到验证码的使用,也算是解疑长知识了一回。