

实验九：网络服务

【实验目的】

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

【实验内容】

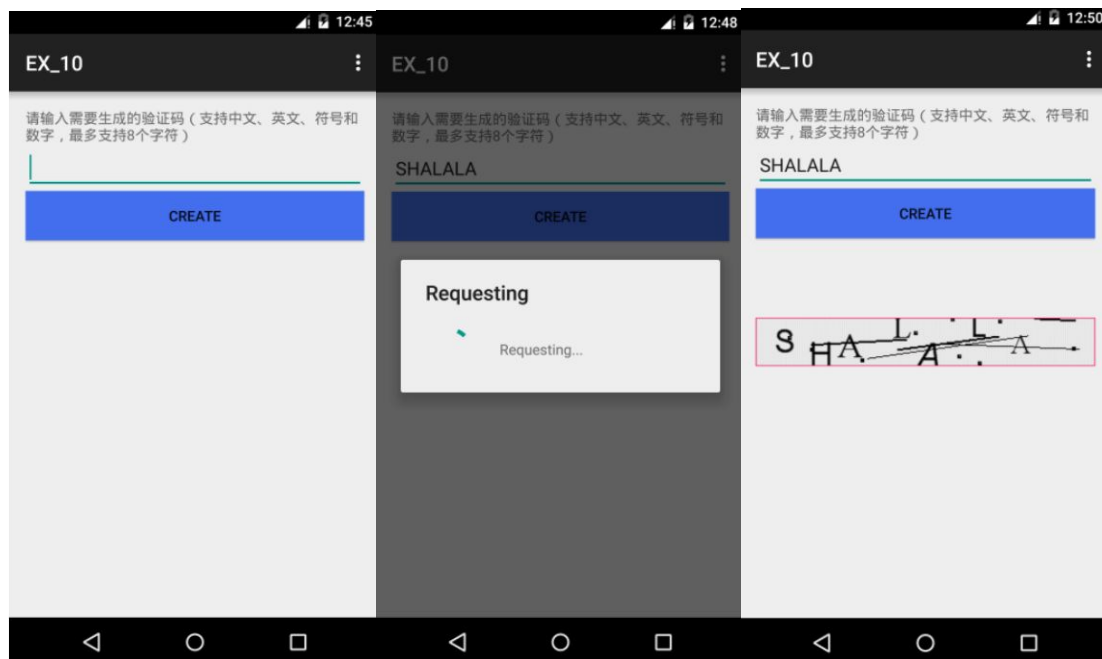
使用 Ksoap2，发送文字信息到指定网络 service 获得生成的验证码图片，并显示出来。

参考界面：

初始界面

查询中

返回结果



【参考内容】

1. 这次使用 webService 是验证码图 web 服务：

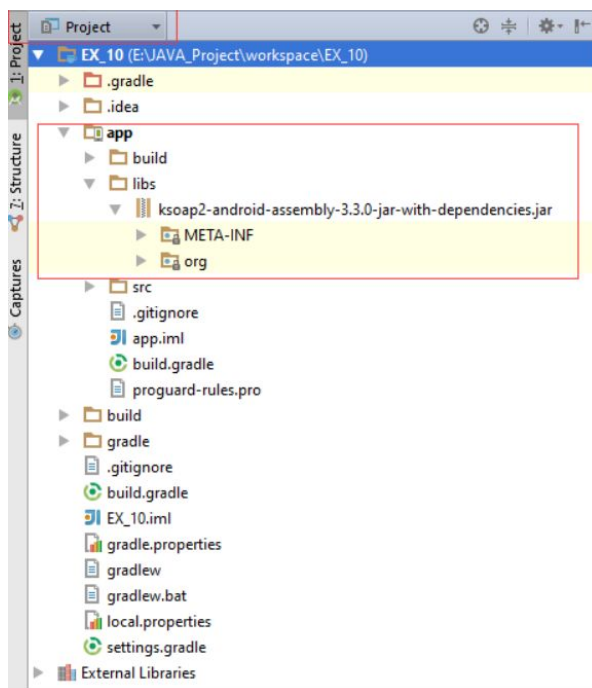
<http://webservice.webxml.com.cn/WebServices/ValidateCodeWebService.aspx>

这次使用的方法是其中的 enValidateByte 方法，该方法支持中文、英文、数字、和符号。

2. 实验中使用了一个 android 平台上轻量级的 SOAP 开发包 Ksoap2android，实验文件夹中有已经下载好的 jar 包，jar 包导入工程的方法为：

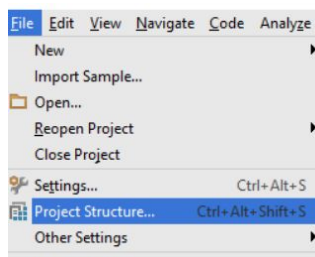
a) 将 jar 包复制到 libs 文件夹里

选择 Project 可以看到在 app 文件夹内的 libs

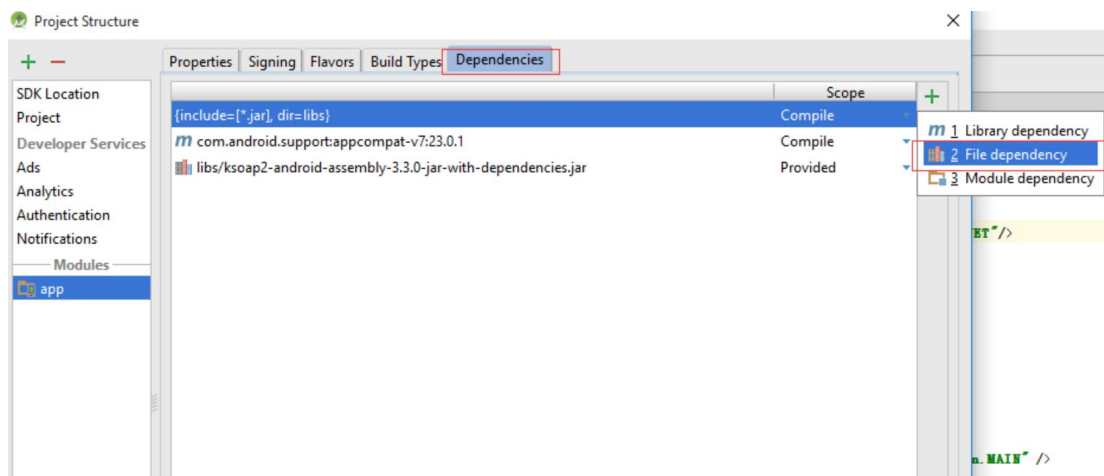


b) 将 jar 添加进依赖项

点击 file 的 project structure:



点击 Dependencies，再点击右边的加号，再选择 File dependency 选到 libs 内的 jar 包添加进去即可。



3. soap 使用方法：

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。

```
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";  
}
```

上面代码段显示的是 soap 需要的命名空间、调用的方法名、soapAction（一般就是命名空间+调用的方法名）和 URL。

【注意】 不要打错字符！！

下面实现：

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

```

@Override
public void run() {
    SoapObject request = new SoapObject(NAMESPACE, METHODNAME);
    Log.e("str", MainActivity.str);
    request.addProperty("byString", MainActivity.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");
}

```

因为 webservice 返回的是验证码图片的字节流，还需要转化成位图才能显示出图片：

```

byte[] data = Base64.decode((msg.obj.toString()).getBytes(), Base64.DEFAULT);
Bitmap bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);

```

得到位图之后使用 handler 来更新 UI 界面。

4. 记得在 Manifest 文件中添加访问网络的权限，注意好位置

```

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

```

5. ProgressDialog 的使用

在访问 Webservice 之前，显示 ProgressDialog;

在返回结果后，隐藏 ProgressDialog;

6. 关于 webservice 以及 soap 更多的知识，推荐两篇不错的博文，有兴趣的同学可以学习一下：

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

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

【提示】

遇到这种错误：

java.lang.ClassCastException: org.ksoap2.SoapFault cannot be
cast to org.ksoap2.serialization.SoapObject

将其打印出来：

```
Log.e("fault", envelope.bodyIn.toString());
```

可以查看是否查询成功还是出错原因

【扩展】

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。（模拟正常应用登陆时需要输入验证码的情况）

界面自拟。