**服务端是webservice的方式, 统一的接口地址：**

[**http://47.97.207.208/apiService/RFService.asmx**](http://47.97.207.208/apiService/RFService.asmx)

**调用接口需要用post方式 ，注意 需要设置 Content-Type: text/xml**   
  
以【测试服务器通不通接口-- **HelloWorld**方法】为例:

<http://47.97.207.208/apiService/RFService.asmx?op=HelloWorld>

### 这个网址 SOAP 1.1 的请求和响应代码，是.net的方式，下面写的是Android的方式，每一个接口，都是类似的过程。 请求内容内容： <?xml version="1.0" encoding="utf-8"?> <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"> <soap:Body> <HelloWorld xmlns="http://tempuri.org/"> <tryConnectDb>1</tryConnectDb> </HelloWorld> </soap:Body> </soap:Envelope>

*// 服务器返回：返回xml数据格式 -- 原始内容如下：***：(有用的是红色的部分，其他照样子复制粘贴即可)**String xml = **"<?xml version=\"1.0\" encoding=\"utf-8\"?>\n"** +  
 **"<soap:Envelopexmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\"xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\">\n"** +  
 **" <soap:Body>\n"** +  
 **" <HelloWorldResponse xmlns=\"http://tempuri.org/\">\n"** +  
 **" <HelloWorldResult>Ok 2018-05-05 06:56:13</HelloWorldResult>\n"** +  
 **" </HelloWorldResponse>\n"** +  
 **" </soap:Body>\n"** +  
 **"</soap:Envelope>"**;

Android端用xml转json的方式接受和处理数据：  
XmlToJson xmlToJson = **new** XmlToJson.Builder(xml).build();  
String jsonStr = xmlToJson.toJson().toString();  
System.***out***.println(jsonStr);

**转化之后的数据内容如下：(有用的是红色的部分：需要巴拉出来，其他不用看)**  
*/\*  
{"soap:Envelope":{"xmlns:soap":"http:\/\/schemas.xmlsoap.org\/soap\/envelope\/","xmlns:xsd":"http:\/\/www.w3.org\/2001\/XMLSchema",****"soap:Body"****:{"HelloWorldResponse":{"xmlns":"http:\/\/tempuri.org\/",****"HelloWorldResult":"Ok 2018-05-05 06:56:13"}****},"xmlns:xsi":"http:\/\/www.w3.org\/2001\/XMLSchema-instance"}}  
\*/*

*🡪 把这个源码拷贝到项目中，用于 xmlToJson转码*

**

（1）登录界面:



字段说明： 订单客户，从 SysCode 表获取 (看db脚本文件)

帐号：admin

密码：123

登录接口： <http://47.97.207.208/apiService/RFService.asmx?op=RFLogin>

网络：这个是wince时代的产物，没用了去掉即可;

看里面的定义，仿照 HellowWord 调用 ，我这里的是：

POST： **(红色部分需要字符串拼接，其他部分复制粘贴)**

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

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Header>

<Authentication xmlns="http://tempuri.org/">

<AccessKey>string</AccessKey>

<Device>string</Device>

</Authentication>

</soap:Header>

<soap:Body>

**<RFLogin xmlns="http://tempuri.org/">**

**<para>**

**<ClientID>123321</ClientID> -- 随便写**

**<UserID>admin</UserID> -- 登录用户admin**

**<Password>123</Password> -- 密码123**

**<AppVer>1.0</AppVer> -- app本地版本号**

**<DvcID>631008030699</DvcID> -- 设备id，这个需要验证的，你先写死这个ok的**

**<Network>1</Network>**

**</para>**

**</RFLogin>**

</soap:Body>

</soap:Envelope>

返回：

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<soap:Body>

<RFLoginResponse xmlns="http://tempuri.org/">

<RFLoginResult>

**<User>**

**<UserID>admin</UserID>**

**<Password>123</Password>**

**<EngName>administrator</EngName>**

**<ChnName>管理人员</ChnName>**

**<OrgID>1</OrgID>**

**<OrgName>1</OrgName>**

**<DvcID>122222222222312</DvcID>**

**<Rights> -- 操作权限列表，先留着，后面我再给定义怎么控制**

**<string>RF001</string>**

**<string>RF002</string>**

**<string>RF003</string>**

**<string>RF004</string>**

**<string>RF005</string>**

**<string>RF006</string>**

**<string>RF007</string>**

**</Rights>**

**</User>**

<AppVer>1.0</AppVer>

<ForceUpdate>0</ForceUpdate> **-- 是否强制更新**

<ServerTime>2018-05-05T15:46:45.1772332+08:00</ServerTime> **-- 服务器本地时间**

<Rst>

<Flag>1</Flag>

<Msg />

</Rst>

</RFLoginResult>

</RFLoginResponse>

</soap:Body>

</soap:Envelope>

（2）主界面：



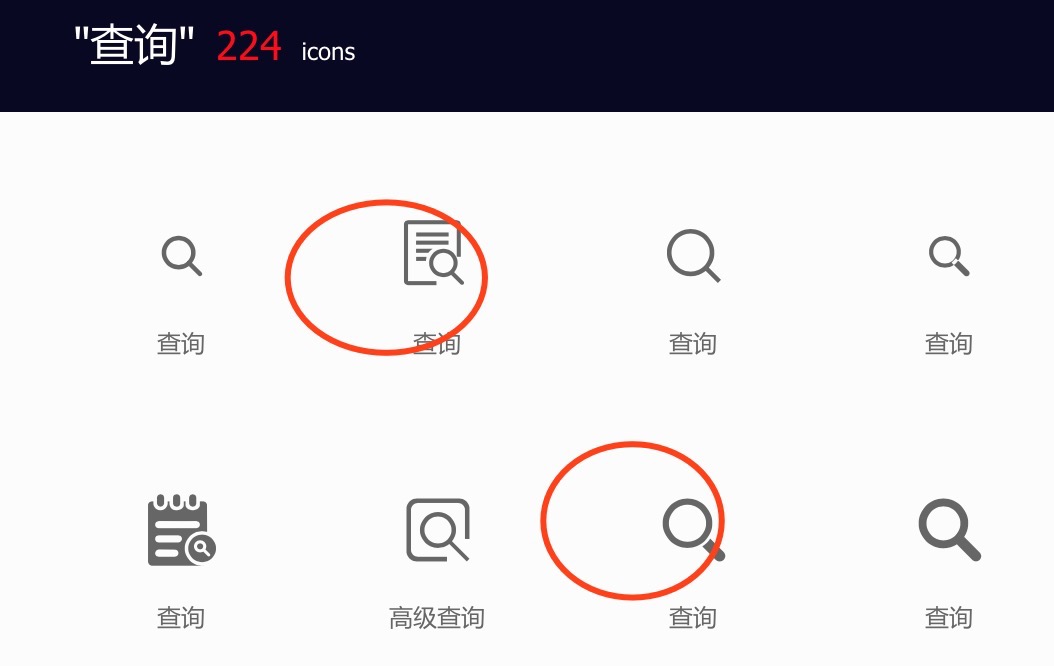
需要重新排一下版本：（底部tab分开即可）

第一个：扫描模块，包括 提货，上车，下车，签收，出库，重打 ： 6个标签按钮;

第二个：查询模块，包括 订单查询，库存查询 两个标签按钮

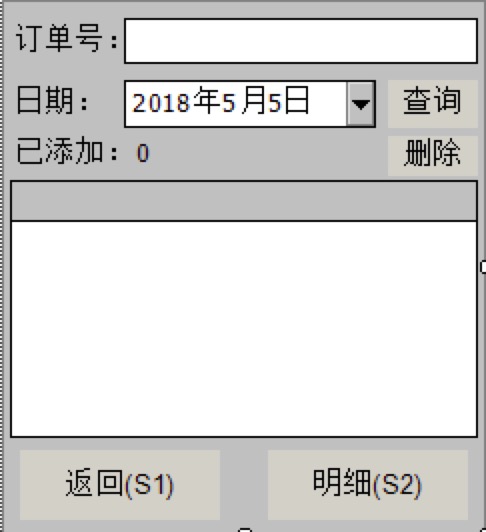
第三个：设置模块, 包括 程序升级检查，其他待定。。

参考： <http://www.iconfont.cn/search/index?q=查询>



各个功能入口的标签图片用阿里的不用找UI设计了。最好是都选择 蓝色 或者 黑色 统一的颜色，以png下载就可以用了;

（3）提货扫描界面-1 : 查询界面:



查询条件，订单号，日期，没什么好说的。。

查询到的结果：中间list列表，包括 OrderID,OrderDate,Status,CrtBillNo （订单号，订单日期，状态，客户单号）四个;

查询接口：

<http://47.97.207.208/apiService/RFService.asmx?op=PupQueryOrderHeader>

查询参数中，只需要填写下面红色部分，其他 复制粘着，post发一通即可;

<soap:Body>

<PupQueryOrderHeader xmlns="http://tempuri.org/">

<para>

<OrderID>11203609</OrderID>

<OrderDate></OrderDate>

<UserID>admin</UserID>

<DvcID>0000</DvcID>

</para>

</PupQueryOrderHeader>

</soap:Body>

得到结果：注意标签名字：PupQueryOrderHeaderResult 🡪 PupQueryOrderHeaderResult 🡪 OrderTitle

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<soap:Body>

<PupQueryOrderHeaderResponse xmlns="http://tempuri.org/">

<PupQueryOrderHeaderResult>

<OrderTitle>

<OrderID>11203609</OrderID>

<OrderDate />

<Status>0</Status>

<CrtBillNo>OT2018050500002</CrtBillNo>

</OrderTitle>

<Rst>

<Flag>1</Flag>

<Msg />

</Rst>

</PupQueryOrderHeaderResult>

</PupQueryOrderHeaderResponse>

</soap:Body>

</soap:Envelope>

查询到的数据，先拿OrderID到表 PupHeader 中查询有没有，有就提示：是否删除本地记录，用户确认后删除本地三个表：（PupHeader, PupDetail, PupScan），

然后重新插入一条记录主表PupHeader:

插入主表PupHeader的SQL，参考::

cmd.CommandText = String.Format("insert into PupHeader (OrderID,OrderDate,Status,CrtBillNo) values ('{0}','{1}','{2}','{3}')",

rst.OrderTitle.OrderID, rst.OrderTitle.OrderDate, rst.OrderTitle.Status, rst.OrderTitle.CrtBillNo);

UI上面的数据显示，从本地表中获取 select OrderID,OrderDate,Status,CrtBillNo from PupHeader where orderid = ‘刚才查询输入的条件id’ - -- 这样做是为了离线缓存;

几个可以测试的提货单号：

11203593,11203595,11203609,11203610,11203614,11203615,11203620

,11203624,11203627,11203630,11203672,11203677,11203678,11203679,11203680

**还是在这个界面中：**

🡪 listview显示上面的记录,用户选择一条以后(应该只有一条)，点击【明细S2】按钮： 需要用得到OrderID ， CrtBillNo ，调用一个订单明细查询接口：

<http://47.97.207.208/apiService/RFService.asmx?op=PodQueryOrderDetail>

POST数据：

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

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Header>

<Authentication xmlns="http://tempuri.org/">

<AccessKey>string</AccessKey>

<Device>string</Device>

</Authentication>

</soap:Header>

<soap:Body>

<PodQueryOrderDetail xmlns="http://tempuri.org/">

<para>

<OrderID>11203593</OrderID> -- 订单号

<UserID>admin</UserID> -- 登录用户名

<DvcID>0000</DvcID> -- 设备id，就是当前手机的IMEI号

</para>

</PodQueryOrderDetail>

</soap:Body>

</soap:Envelope>

**返回结果：**

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<soap:Body>

**<PodQueryOrderDetailResponse xmlns="http://tempuri.org/">**

**<PodQueryOrderDetailResult>**

**<OrderDetail>**

**<OrderDetailEty>**

**<CargoName>100502DRUM01</CargoName>**

**<CargoID>100502DRUM012.01E11</CargoID>**

**<Model />**

**<BatchNo>2.01E11</BatchNo>**

**<Count>3</Count>**

**<Wgt>51.000</Wgt>**

**<Rmk />**

**</OrderDetailEty>**

**</OrderDetail>**

**<Rst>**

**<Flag>1</Flag>**

**<Msg />**

**</Rst>**

**</PodQueryOrderDetailResult>**

**</PodQueryOrderDetailResponse>**

</soap:Body>

</soap:Envelope>

查询到的数据，用

用 "insert into PupDetail (OrderID,CargoID,CargoName,Model,BatchNo,Count,Weight,Remark) values (@OrderID,@CargoID,@CargoName,@Model,@BatchNo,@Count,@Weight,@Remark)"

插入到本地数据库表 PupDetail ; （**插入之前这个表的数据，已经经过用户确认删除了，因此不会有重复的情况**）

下一个提货明细界面（4）上显示时，用 select CargoID,CargoName,Model,BatchNo,Count,Weight,Remark from PupDetail where OrderID='{0}'

这个SQL， 从本地表中提取 显示到（4）：：

（4） 提货明细界面::



中间listview的字段对应项目是： (依次对应 CargoID,CargoName,Model,BatchNo,Count,Weight,Remark )

colHeader1 = new ColHeader("物料编码", 80, HorizontalAlignment.Left, true);

colHeader2 = new ColHeader("物料名称", 80, HorizontalAlignment.Left, true);

colHeader3 = new ColHeader("型号", 60, HorizontalAlignment.Left, true);

colHeader4 = new ColHeader("批次", 60, HorizontalAlignment.Left, true);

colHeader5 = new ColHeader("数量", 50, HorizontalAlignment.Left, true);

colHeader6 = new ColHeader("重量", 50, HorizontalAlignment.Left, true);

colHeader7 = new ColHeader("备注", 80, HorizontalAlignment.Left, true);

用户点击 提货S2按钮，转到界面（5）：

(5) 提货明细—扫描界面:(当前界面中可以扫描的数据范围，就是上面界面（4）中的货品范围：**物料编码 就是条码**)



中间listview显示的数据项目是：(4列)

colHeader1 = new ColHeader("物料编码", 100, HorizontalAlignment.Left, true);

colHeader2 = new ColHeader("数量", 50, HorizontalAlignment.Left, true);

colHeader3 = new ColHeader("重量", 50, HorizontalAlignment.Left, true);

colHeader4 = new ColHeader("备注", 80, HorizontalAlignment.Left, true);

🡪 【扫描条码处理过程】：扫描到一个 CargoID 就需要刷新listview中的数据, 扫描几次就有几条记录（不管物料编码是否重复，不用汇总数量）  
(结果存储到专用表PupScan中：**注意，每扫描一条，就插入一条记录，不用汇总数量**,界面看到的就根据扫描时间倒序排列的数据列表):

写代码可以这样做：

1. 这个界面刚打开时，需要从扫描记录表 PupScan 中加载以前扫描存储的数据记录：

用这个 SQL select CargoID,Count,Weight,Remark from PupScan where OrderID='{0}'

2）每次扫描，都往PupScan表中插入一条记录，然后从 PupScan 这里面重新提取数据，刷新UI（上面的SQL）；

**注意 PupDetail 这个表是下载的原始数据表，扫描操作的数据，保存到 PupScan表中;**

**3）如果扫描条数，超过了 原始记录数 PupDetail.** Count ，就提示用户：当前货品XX的数量扫描数（11）已经超过了订单数量（10），是否继续保存，用户确认的话就保存进来；

🡪 【异常拍照】就是调用摄像头，然后把图片转为base64上传：另外一个接口，我在调整。。

🡪 【打印】这个需要调用济强打印机，打印类似于收银小票的格式，格式我找一下：注意的一个地方是：每一个条码，就是一行，重复多少不用管；

🡪 【完成】按钮：调用上传接口：(定义在 http://localhost/RFWebService/RFService.asmx?op=PodUpload)

看定义：

<http://47.97.207.208/apiService/RFService.asmx?op=PodUpload>

**POST内容：（我想了一下：上传时直接拼接xml格式，比先拼接json然后转xml格式，还是省事一点的）**

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

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Header>

<Authentication xmlns="http://tempuri.org/">

<AccessKey>string</AccessKey>

<Device>string</Device>

</Authentication>

</soap:Header>

**🡪 以上内容是固定的，以下部分是需要拼接字符串的**

<soap:Body>

<PodUpload xmlns="http://tempuri.org/">

<para>

<Consignee>string</Consignee>

<SignOrg>string</SignOrg>

<Telephone>string</Telephone>

<Cellphone>string</Cellphone>

<DeliverStatus>int</DeliverStatus>

<CrtBillNo>string</CrtBillNo>

<DvcID>string</DvcID>

<MsgList> **-- 字符串拼接成为数组，**

<PodScanMsgEty> -- 标签字符串固定

<OrderID>string</OrderID>

<CargoID>string</CargoID>

<Count>int</Count>

<Rmk>string</Rmk>

<UserID>string</UserID>

<ScanTime>dateTime</ScanTime>

</PodScanMsgEty>

<PodScanMsgEty>

<OrderID>string</OrderID>

<CargoID>string</CargoID>

<Count>int</Count>

<Rmk>string</Rmk>

<UserID>string</UserID>

<ScanTime>dateTime</ScanTime>

</PodScanMsgEty>

</MsgList>

</para>

</PodUpload>

</soap:Body>

</soap:Envelope>

返回结果: