

1. 交互模式

交互模式为：请求/响应交互模式。需要数据一方作为请求方，提供数据一方作为响应方。请求方发送一个带有参数的报文到响应方，响应方根据请求 ip 地址、报文中的参数等判断请求是否合法，如果不合法，返回包含错误信息报文，如果合法，则根据请求报文中的参数获取相应的数据，然后组装成响应报文发送回请求方。

请求方可以是 java 程序、.net 程序、php 程序等，响应方是一个 servlet。为了简单起见，一个 servlet 对应一个接口。

2. 接口

接口的请求和响应采用 XML 作为具体业务报文的载体，请求为 POST 方式，传输时的 charset 需要设置为 utf-8

2.1 巡检报表接口

2.1.1 说明

巡检报表需要显示的数据如下图：

地 市	代 维 公 司	区 县	专 业	人 员	开始时间	结束时间	计划明细数	合格明细数	巡检率

巡检报表响应地址：http://xxx.xxx.xxx.xxx/xxxservlet

2.1.2 请求报文

报文格式

```
<?xml version="1.0" encoding="UTF-8"?>
<report_data>
  <request_req>953947334</request_req>
```

```
<request_time>2012040211394324</request_time>

<request_param>

    <query_month>201203</query_month>

</request_param>

</report_data>
```

报文解释

节点名称	中文名称	类型	说明
request_req	请求流水号	String(32)	每笔请求的流水号必须不同，以示区分
request_time	请求时间	String(14)	发起请求段时间，格式：yyymmddhhnnss
query_month	月份	String(6)	需要获取数据的月份

2.1.3 响应返回报文

报文格式

```
<?xml version="1.0" encoding="UTF-8"?>
<report_data>
    <respon_req>953947334</respon_req>
    <respon_time>20120402113943</respon_time>
    <result>
        <id>0000</id>
        <comment>成功</comment>
    </result>
    <items>
        <item>
            <county>长治县</county>
            <company>铁通</company>
            <speciality>线路</speciality>
            <personnel>王加和</personnel>
            <begin_time>20120301000000</begin_time>
            <end_time>20120331235959</end_time>
            <plan_quantity>50</plan_quantity>
            <checkout_quantity>40</checkout_quantity>
            <patrol_rate>0.80</patrol_rate>
```

```

</item>
<item>
    <county>长治县</county>
    <company>晋通管道维护公司</company>
    <speciality>线路</speciality>
    <personnel>李小林</personnel>
    <begin_time>20120301000000</begin_time>
    <end_time>20120331235959</end_time>
    <plan_quantity>30</plan_quantity>
    <checkout_quantity>10</checkout_quantity>
    <patrol_rate>0.66</patrol_rate>
</item>
<item>
    <county>方山县</county>
    <company>晋通管道维护公司</company>
    <speciality>线路</speciality>
    <personnel>陈强</personnel>
    <begin_time>20120301000000</begin_time>
    <end_time>20120331235959</end_time>
    <plan_quantity>60</plan_quantity>
    <checkout_quantity>30</checkout_quantity>
    <patrol_rate>0.50</patrol_rate>
</item>
.....
</items>
</report_data>

```

报文解释

节点名称	中文名称	类型	说明
respon_req	响应流水号	String(32)	每笔响应的流水号必须不同，以示区分
respon_time	响应时间	String(14)	响应时间，格式：yyymmddhhnnss
id	响应码	String(5)	系统返回的处理结果，详见接口返回错误码表
comment	响应码解释	String(500)	响应码对应中文注解
county	县区	String(10)	县区名称
company	代维公司	String(30)	代维公司名称
speciality	专业	String(10)	专业名称

personnel	人员	String(20)	巡检人员名称
begin_time	开始时间	String(14)	开始时间，格式： yyymmddhhnnss
end_time	结束时间	String(14)	结束时间，格式： yyymmddhhnnss
plan_quantity	计划明细数	Number(7)	计划明细数
checkout_quantity	合格明细数	Number(7)	合格明细数
patrol_rate	巡检率	Number(7)	巡检率

备注：如请求非法或参数错误，则响应报文中无 items 节点

2. 2xxxx 报表接口

2. 3xxxx 报表接口

2. 4xxxx 报表接口

4. 接口错误返回码

响应代码	代码解释
0000	交易成功
0010	请求 IP 地址非法
0020	请求报文解析错误
0030	无法获取请求报文
0040	该接口不支持 GET 方式，请以 POST 方式提交
.....

5. 程序示例

5.1 请求程序代码

```
public void sendMessage() throws Exception {
    System.out.println("调用servlet开始=====");
    StringBuffer sendStr = new StringBuffer();
    sendStr.append("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
}
```

```

        sendStr.append("<report_data>");
        sendStr.append("<request_req>953943547334</request_req>");

sendStr.append("<request_time>2012040211394324</request_time>");
        sendStr.append("<request_param>");
        sendStr.append("<query_month>201203</query_month>");
        sendStr.append("</request_param>");
        sendStr.append("</report_data>");

        BufferedReader reader = null;

        try {
            String strMessage = "";
            StringBuffer buffer = new StringBuffer();

            // 接报文的地址
            URL uploadServlet = new URL(
                "http://localhost:9090/TestTransfers");

            HttpURLConnection servletConnection = (HttpURLConnection)
uploadServlet
                .openConnection();
            // 设置连接参数
            servletConnection.setRequestMethod("POST");
            servletConnection.setDoOutput(true);
            servletConnection.setDoInput(true);
            servletConnection.setAllowUserInteraction(true);

            // 开启流，写入XML数据
            OutputStream output = servletConnection.getOutputStream();
            System.out.println("发送的报文: ");
            System.out.println(sendStr.toString());

            output.write(sendStr.toString().getBytes());
            output.flush();
            output.close();

            // 获取返回的数据
            InputStream inputStream =
servletConnection.getInputStream();
            reader = new BufferedReader(new
InputStreamReader(inputStream));
            while ((strMessage = reader.readLine()) != null) {
                buffer.append(strMessage);
            }
        }
    }
}

```



```

    }

    System.out.println("接收返回值:" + buffer);

} catch (java.net.ConnectException e) {
    throw new Exception();
} finally {
    if (reader != null) {
        reader.close();
    }
}
}
}

```

5.2 响应程序代码

```

public class TestTransfers extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {

    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {

        //判断请求报文是否来自代维系统的ip地址
        String ip = request.getRemoteHost();

        // 获取收到的报文
        BufferedReader reader = request.getReader();
        String line = "";
        StringBuffer inputString = new StringBuffer();
        while ((line = reader.readLine()) != null) {
            inputString.append(line);
        }

        //如有必要，可以在报文中增加其他验证和加密的参数
        //解析获取到的报文，根据ip地址、其他验证、加密等等来判断请求报文的服务器是
        否有权限
    }
}

```

```

//如果请求验证合格，则根据请求的参数装配返回的报文

// 要返回的报文
StringBuffer resultBuffer = new StringBuffer();
resultBuffer.append("<?xml version=\"1.0\" 
encoding=\"UTF-8\"?>");
resultBuffer.append("<report_data>");
resultBuffer.append("<respon_req>953947334</respon_req>");

resultBuffer.append("<respon_time>20120402113943</respon_time>");
resultBuffer.append("<result>");
resultBuffer.append("<id>0000</id>");
resultBuffer.append("<comment>成功</comment>");
resultBuffer.append("</result>");
resultBuffer.append("<items>");
resultBuffer.append("<item>");
resultBuffer.append("<county>长治县</county>");
resultBuffer.append("<company>铁通</company>");
resultBuffer.append("<speciality>线路</speciality>");
resultBuffer.append("<personnel>王加和</personnel>");

resultBuffer.append("<begin_time>20120301000000</begin_time>");
resultBuffer.append("<end_time>20120331235959</end_time>");
resultBuffer.append("<plan_quantity>50</plan_quantity>");

resultBuffer.append("<checkout_quantity>40</checkout_quantity>");
resultBuffer.append("<patrol_rate>0.80</patrol_rate>");
resultBuffer.append("</item>");
//.....
//.....
//.....
//循环组装响应的报文

resultBuffer.append("</items>");
resultBuffer.append("</report_data>");

// 设置发送报文的格式
response.setContentType("text/xml");
response.setCharacterEncoding("UTF-8");

PrintWriter out = response.getWriter();
out.println(resultBuffer.toString());
out.flush();

```

```
        out.close();  
    }  
}
```