

# XMPP 协议简单参考

作者: [shengkai](#)

原文链接: <https://ld246.com/article/1468551878448>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



```
</stream:features>
```

说明

服务器会响应给你一个新的`<stream:stream>`会话，大概询问你需要什么方式验证并连接。你需要接着这个会话回应服务器

`<stream:features>`：该节点中列出了可供选择的验证方式

`<starttls xmlns="urn:ietf:params:xml:ns:xmpp-tls"><starttls>`：表示服务器支持开启TLS安全传输协议

`<mechanisms>`：点中有一些选项，表示服务器支持的验证方式。分别代表：

- `PLAIN`：简单验证，直接使用Base64编码
- `ANONYMOUS`：匿名登录（服务器会分配给你一个临时的账户）
- `SCRM-SHA-1`：后面的这些在网上页找不到相关的资料，所以我也不知道具体是什么意思。有时间我得本书看看

`<compression>`：节点表示服务器支持压缩并列出了支持的压缩方案

`<auth>`：表示支持账号密码认证

`<register>`：表示支持注册

## 采用已有账号进行简单登陆

这里我选择PLAIN验证方式并使用账号密码直接登陆。这种验证最简单

假设我的账号是admin密码是admin123的话我将账号和密码这样拼接：`"\0admin\0admin123"`并进行Base64编码得到最终的值

```
String value = Hash.BASE_UTIL.encode("\0admin\0admin123");
```

将编码后的账号密码发用 `auth` 标签发给服务器进行登陆

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl' mechanism='PLAIN'>AGFkbWluAGFkbWluMTIz
```

说明：

`xmlns` 的值表示已 sasl 的方式传输

`mechanism` 的值表示已 PLAIN 方式验证，也是最简单的验证方式

节点中的内容就是账号密码进行 base64 编码后的值

服务器回应：

```
<success xmlns="urn:ietf:params:xml:ns:xmpp-sasl"/>
```

说明：

成功通过验证并确定数据交流方式为 sasl 方式。

## 绑定资源

绑定资源就是声明一下你当前所使用的客户端资源类型。说白了就是你的客户端类型。比如 Windows、Surface、Lumia、WEB、Android 等这事可以自定义的。比如 `asdasdasd` 也是可以的。

```
<iq type='set' id='bind_2'><bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'><resource>Windows</resource></bind>
```

说明：

`<iq>`：这种节点一般用与服务器的交互，待会儿再系统的说，现在说的话一些新手反而会手忙脚乱，比如说我这样的。

`<bind>`：表示我要进行资源绑定操作！

`<resource>`：我要绑定的资源名。因为我是开发 Windows 客户端，所以我把资源名定为 Windows，然后我就可以显示 Windows 在线了

服务器响应：

```
<iq type="result" id="bind_2" to="server.im.321aiyi.com/5tpldmpwov"><bind xmlns="urn:ietf:params:xml:ns:xmpp-bind" id="bind_2" to="server.im.321aiyi.com/5tpldmpwov" jid="admin@server.im.321aiyi.com/Windows" />
```

说明：

`type`：表示服务器响应的状态，值为 `result` 表示绑定成功。

`id="bind_2"`：表示绑定的类型是自定义源类型。

`<bind>`：服务器回应你绑定的资源内容

`<jid>`：绑定后，服务器分配给你的 JID，JID 表示一个用户。（最简单的 JID 由 账号 + @ + 服务器地址三部分构成），样留在后面讲

## 设置用户状态

验证成功 `bind` 后，其实你已经在线了，只不过别人不知道。准确的说，你现在是隐身的。你已经可以和别人对话并收别人给你发的消息了。这个时候，你要告诉大家伙：“喂！我来勒！”所以，你要声明一下你目前状态为在线状态。

当然，除了在线状态之外，还有离线状态、忙碌、什么的一大堆（可选：`online`、`away`、`dnd`）。

```
<presence from='admin@server.im.321aiyi.com/Windows' to='server.im.321aiyi.com'>
```

```

<presence from="sys@server.im.321aiyi.com" status="Online" type="chat" />
</presence>

```

说明：

- `<presence>`：表示出席，也是出场、来到的意思，其实说白了就是上线
- `from`：出席来源，来源是我，所以这里的值写的 JID
- `to`：出席目标，我要去哪里出席，我在整个服务器上出席，服务器的所有人都能看到我在线，所以我给他的值写 `server.im.321aiyi.com`（假如入我只给某个人出席的话，就把服务器地改为对方的 JID，同理，我只对某人隐身的话，就现在服务器出席，然后再对某人单独已隐身状态下席）
- `<status>`：表示出席状态，比如我要以在线状态出席，那么我写 `Online`

```

<presence from="sys@server.im.321aiyi.com/Windows" to="admin@server.im.321aiyi.com/Windows" priority="0" status="chat" type="chat" />

```

说明

- 表示 `sys` 这个用户知道了你的出席，告诉你了 `sys` 这个用户的出席状态

## 发送一个简单的消息

加入我要给 `sys` 这个用户私发一条内容为“你好！”的消息，我可以这么写。

```

<message from="admin@server.im.321aiyi.com/Windows" to="sys@server.im.321aiyi.com" type="chat" body="你好!" />

```

说明：

- `<message>`：节点表示发送或受一条消息，这里简单的说一下，系统的解释后面说。
- `type="chat"`：表示消息的类型，是一条私聊的消息
- `<body>`：该节点中是消息的内容

## IQ 语法

IQ 分别代表 Info 和 Query。IQ 在 XMPP 中是一个节点，该节点通常用于与服务器的信息查询与提交。类似 HTTP 协议中的的 GET 与 POST 的关系。

利用 IQ 节点请求服务器的话，服务器都将必须给予一个响应。所以 IQ 节点中必须通过一个 `id` 属性来保持一个消息的通道。

举个简单的例子，假如服务器向我发送 `<iq id="1" />` 的话，我如果响应 `<iq id="2" />` 的话，那么抱歉，服务器并不会认账。他不会认为你向服务器发的那个 `iq` 是响应之前的请求的。

另外，IQ 的 `type` 属性也是必填的，`type` 属性拥有四种：

- `get`，表示查询请求；
- `set`，表示提交请求；
- `result`，表示成功响应请求的结果；
- `error`，表示处理请求失败。

下一个例子：

## 请求我的花名册

客户端请求数据：

```

<iq from="jane@longbourn.lit/garden" type="get" id="roster1" />

```

说明：

- 客户端发送了一个获取花名册的请求，`xmlns="jabber:iq:roster"` 表示获取花名册。
- 服务器响应：

```

<iq to="jane@longbourn.lit/garden" type="result" id="roster1" />

```

服务器响应了花名册的内容

```

<query xmlns="jabber:iq:roster" />
  <item jid="elizabeth@longbourn.lit" name="Elizabeth" />
  <item jid="bingley@netherfield.lit" name="Bingley" />
</query>

```

说明：

- `type="result"` 表示响应成功，
- `id="roster1"` 表示我响应的是之前发送的 `id` 为 `roster1` 的请求，即名册请求。
- `<query xmlns="jabber:iq:roster">` 是服务器响应的花名册结果集。
- `<item>` 就是花名册中的每个成员。

## 在我的花名册中添加一个成员

```

<iq from="jane@longbourn.lit/garden" type="set" id="roster1" />

```

```
= "jane@longbourn.lit/garden" &nbsp; type="set" &nbsp; id="roster3" &gt; <br>
&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &lt;/query &nbsp; xmlns="jabber:iq:roster" &gt; <br>
&nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &lt;item &nbsp; jid=" <a href="https://ld246.com/forward?goto=
ailto%3Adarcy%40pemberley.lit" target="_blank" rel="nofollow ugc"> darcy@pemberley.lit </
"&nbsp; name="Mr.&nbsp;Darcy"/&gt; <br>
&lt;/query&gt; <br>
&lt;/iq&gt; </pre> <p>说明: </p> <blockquote> <p>这里表示把账号为 darcy 的用户添加到我
花名册中, 并设置备注为 "Mr. Darcy" </p> </blockquote> <p>服务器响应: </p> <pre class="b
ush: xml;toolbar:false"> &lt;iq &nbsp; to="jane@longbourn.lit/garden" &nbsp; type="result" &n
sp;id="roster3"/&gt; </pre> <p>说明: </p> <blockquote> <p>成功添加到化名侧 </p> </block
note> <p> <br> </p> <p> </p>
```