

## msgserver

---

snax.msgserver 是一个基于消息请求和回应模式的网关服务器模板。它基于 snax.gateserver 定制，可以接收客户端发起的请求数据包，并给出对应的回应。

和 service/gate.lua 不同，用户在使用它的时候，一个用户的业务处理不基于连接。即，它不把连接建立作为用户登陆、不在连接断开时让用户登出。用户必须显式的登出系统，或是业务逻辑设计的超时机制导致登出。

和 GateServer 和 LoginServer 一样，snax.msgserver 只是一个模板，你还需要自定义一些业务相关的代码，才是一个完整的服务。与客户端的通信协议使用的是两字节数据长度协议。

### msgserver 服务模板

```
local msgserver = require "snax.msgserver"

local server = {}

msgserver.start(server) --服务初始化函数，要把 server 表传递进去。
```

--在打开端口时，会触发这个 register\_handler 函数参数 name 是在配置信息中配置的当前登陆点的名字  
--你在这个回调要做的事件是通知登录服务器，我这个登录点准备好了

```
function
```

```
server.register_handler(name)
```

```
end
```

--当一个用户登陆后，登陆服务器会转交给你这个用户的 uid 和 serect ，最终会触发 login\_handler 方法。

--在这个函数里，你需要做的是判定这个用户是否真的可以登陆。然后为用户生成一个 subid ，使用 msgserver.username(uid, subid, servername) 可以得到这个用户这次的登陆名。这里 servername 是当前登陆点的名字。

--在这个过程中，如果你发现一些意外情况，不希望用户进入，只需要用 error 抛出异常。

```
function
```

```
server.login_handler(uid, secret)
```

```
end
```

--当一个用户想登出时，这个函数会被调用，你可以在里面做一些状态清除的工作。

```
function
```

```
server.logout_handler(uid, subid)
```

```
end
```

--当外界（通常是登陆服务器）希望让一个用户登出时，会触发这个事件。

--发起一个 logout 消息（最终会触发 logout\_handler）

```
function
```

```
server.kick_handler(uid, subid)
```

```
end
```

--当用户的通讯连接断开后，会触发这个事件。你可以不关心这个事件，也可以利用这个事件做超时管理。

--（比如断开连接后一定时间不重新连回来就主动登出。）

```
function
```

```
server.disconnect_handler(username)
```

```
end
```

--如果用户发起了一个请求，就会被这个 request\_handler 会被调用。这里隐藏了 session 信息，

--等请求处理完后，只需要返回一个字符串，这个字符串会回到框架，加上 session 回应客户端。

--这个函数中允许抛出异常，框架会正确的捕获这个异常，并通过协议通知客户端。

```
function
```

```
server.request_handler(username, msg, sz)
```

```
end
```

## 最简单 msgserver

编写一个最最简单的 simplemsgserver.lua:

```
local msgserver = require "snax.msgserver"
```

```
local crypt = require "skynet.crypt"
```

```
local skynet = require "skynet"
```

```
local subid = 0
```

```
local server = {} --一张表，里面需要实现前面提到的所有回调接口
```

```
local servername
```

--外部发消息来调用，一般用来注册可以登陆的登录名

```
function server.login_handler(uid, secret)
```

```
    skynet.error("login_handler invoke", uid, secret)
```

```
    subid = subid + 1
```

--通过 uid 以及 subid 获得 username

```
    local username = msgserver.username(uid, subid, servername)
```

```
    skynet.error("uid",uid, "login, username", username)
```

```
msgserver.login(username, secret)--正在登录，给登录名注册一个 secret
```

```
return subid
```

```
end
```

--外部发消息来调用，注销掉登陆名

```
function server.logout_handler(uid, subid)
```

```
skynet.error("logout_handler invoke", uid, subid)
```

```
local username = msgserver.username(uid, subid, servername)
```

```
msgserver.logout(username)end
```

--外部发消息来调用，用来关闭连接

```
function server.kick_handler(uid, subid)
```

```
skynet.error("kick_handler invoke", uid, subid)
```

```
end
```

--当客户端断开了连接，这个回调函数会被调用

```
function server.disconnect_handler(username)
```

```
skynet.error(username, "disconnect")
```

```
end
```

--当接收到客户端的请求，这个回调函数会被调用,你需要提供应答。

```
function server.request_handler(username, msg)
```

```
skynet.error("recv", msg, "from", username)
```

```
return string.upper(msg)
```

```
end
```

--监听成功会调用该函数，name 为当前服务别名

```
function server.register_handler(name)
```

```
skynet.error("register_handler invoked name", name)
```

```
servername = name
```

```
end
```

```
msgserver.start(server) --需要配置信息
```

## 发送 lua 消息启动 msgserver

要启动 msgserver，需要给 msgserver 发一个 lua 消息 open（msgserver 框架已经能处理 open 消息）例如

我们编写一个 msgserver 的启动 msgserver 服务，代码 sendtomsgserver.lua：

```
local skynet = require "skynet"

skynet.start(function()

    local gate = skynet.newservice("simplemsgserver")

    --网关服务需要发送 lua open 来打开，open 也是保留的命令

    skynet.call(gate, "lua", "open", {

        port = 8002,

        maxclient = 64,

        servername = "sample", --取名叫 sample，跟使用 skynet.name(".sample")一样

    })

    local uid = "nzhsoft"

    local secret = "11111111"

    local subid = skynet.call(gate, "lua", "login", uid, secret)

    --告诉 msgserver，nzhsoft 这个用户可以登陆

    skynet.error("lua login subid", subid)

    skynet.call(gate, "lua", "logout", uid, subid)

    --告诉 msgserver，nzhsoft 登出
```

```
skynet.call(gate, "lua", "kick", uid, subid)
```

--告诉 msgserver, 剔除 nzhsoft 连接

```
skynet.call(gate, "lua", "close")
```

--关闭 gate, 也就是关掉监听套接字

```
end)
```