1.sock.write()传数据到服务端

2.sock.read()从服务端读取数据

3.struct.pack()编码

4.struct.unpack()解码

function ServerScene:client()

auto\_reconnect('127.0.0.1', 8000, function(sock)

local tasklet = import('babe/tasklet') --微进程

local struct = import("babe/encoding/struct") -- 结构

-- send test request every second.

tasklet.spawn(function()

while true do

--for i =1,5 do

tasklet.sleep(1)

local file = io.open("C:/BYIDEProjects/newProject/assets/package/ServerTest/test.lua", "r+")

io.input(file)

local str = io.read("\*a")--读取文件所有内容

local len = struct.pack('>I4', #str)--将字符串的长度内容放进格式中

assert(sock:write(len..str)) --将长度和str传过去

--print('client write', #str)

end

end)

-- keep receiving packets.

while true do

--for i =1,5 do

local header = assert(sock:read(4)) --先读取前面4个字节，解码得到str的长度信息

local len = struct.unpack(">I4",header)

local str2 = assert(sock:read(len)) --读取到正真要传输的字符串（不需要解码直接输出）

print('client recv', str2)

end

end)

end