# 服务端框架文档

## 简介

为了方便后端人员快速上手，提高开发效率，统一编码规范，制作了下面统一的后端框架。

## 快速上手

#### 准备

##### 安装lua5.3.4

see <http://www.lua.org/download.html>

curl -R -O http://www.lua.org/ftp/lua-5.3.0.tar.gz

tar zxf lua-5.3.0.tar.gz

cd lua-5.3.0

make linux test

make install

##### 安装ruby

// many system already install ruby

see https://www.ruby-lang.org/en/documentation/installation

sudo gem install redis //安装ruby's redis library

##### 安装python

// many system already isntall python

see https://www.python.org/downloads/source

##### 安装redis4.0.9

see <https://redis.io/download>

step2：安装tcl

tar －zvxf tcl8.4.4.tar.gz  
cd tcl8.4..4/  
cd unix  
./configure  
make  
make install

##### ubuntu

sudo apt-get install libreadline-dev

sudo apt-get install autoconf

sudo apt-get install git

sudo apt-get install subversion

//install openresty

//see https://openresty.org/cn/installation.html

apt-get install libpcre3-dev libssl-dev perl make build-essential curl

wget https://openresty.org/download/openresty-1.13.6.2.tar.gz

tar -zxvf openresty-1.13.6.2.tar.gz

cd openresty-1.13.6.2

./configure

make

sudo make install

//add /usr/local/openresty/bin to $PATH

//sudo apt-get install luarocks

//source install luarocks for lua5.3

git clone https://github.com/luarocks/luarocks

cd luarocks

./configure --lua-version=5.3 --versioned-rocks-dir

make build

sudo make install

//we will get luarocks-5.3

sudo luarocks install luasocket

sudo luarocks install lpeg

//install protoc

sudo apt-get install protobuf-compiler

#### 部署

##### 创建工作目录

mkdir ~/app

##### 检出安装脚本

cd ~/app

svn checkout

http://172.16.100.8/svn/core/PublicModule/PubSkynet/BaseServer/example/tools/script

##### 部署服务器

cd ~/app

// 执行下面语句查看如何部署服务器

python script/install\_gamesrv.py –h

python script/install\_accountcenter.py -h

#### 编译

具体看各目录下的README.md

#### 运行

具体看各目录下的README.md

#### 简易客户端

看client/README.md

## 服务器结构

#### 节点结构

账号中心

玩家

游戏1服

游戏2服

DB集群

游戏框架最简单结构构造如上。

账号中心

负责账密验证、充值回调认证等

游戏服

负责游戏玩法

DB集群

负责数据落地，示例服务器暂时是用redis做的

如果想分离战斗玩法，可以独立出战斗服，如果做世界服/特殊玩法时可能会独立出部分功能中心服

#### 游戏服内部结构

Gate服务

Logger服务

主服

Db服务

玩家

其他自定义服务

根据业务需求，其他自定义服也可能直接连gate服务

内部主要自定义服务大致有

Gate服务 [可选] 管理所有客户端连接（如果无客户端连接，则可以无gate服）

Db服务 [可选] Db操作代理服务

Logger服务 [常驻] 负责日志操作

主服务 [常驻] 管理大部分玩法

其他服务 [可选] 根据业务需求自行创建的服务（比如MMO游戏中场景服）

#### 登录流程

备注：约定C代表客户端,AC代表账号中心,GS代表游戏服，PS代表第三方平台。C2AC:客户端->账号中心 AC2C:账号中心->客户端 C2PS: 客户端-> 第三方平台 C2GS: 客户端->游戏逻辑服

1. 注册

内部账号:

C2AC: 注册账号

第三方平台:

C2PS: 注册账号

2. 登陆

内部账号:

C2AC: 账密验证

AC2C: 验证成功返回客户端一个认证token

第三方平台账号:

C2PS:到第三方平台进行账密验证，验证通过会获得一个<平台认证token>,可能还会返回一个uid(第三方平台的账号唯一标识)

C2AC:利用<平台认证token>经过账号中心再次验证,做这步是为了确保账号中心确认平台账密验证通过了,并且在本地自动注册平台的账号

AC2C: 验证成功返回客户端一个认证token

3. 登陆成功后，获取服务器列表(C2AC)

4. 获取角色列表(C2AC)

5. 选择服务器

6. 连接选择的服务器，进行token认证登陆(C2GS)

7. 角色列表为空:

新建角色(C2GS)

不为空:

新建角色(C2GS)

或者选择角色

8. 用选择的角色进入游戏(C2GS)

#### Svn目录说明

Svn路径: <http://172.16.100.8/svn/core/PublicModule/PubSkynet>

BaseServer

+base 公共模块代码

+example 示例服务器代码

+accountcenter 账号中心代码

+dev 账号中心开发分支代码

+test 账号中心测试分支代码

+release 账号中心正式分支代码

+gamesrv 游戏服代码

+dev 游戏服开发分支代码

+test 游戏服测试分支代码

+release 游戏服正式分支代码

+shell 管理shell脚本代码（如启服脚本）

+client 简易客户端(可以在linux下链接服务器，发送协议)

+tools 工具代码

+dbdata db配置脚本

+script 部署服务器的脚本代码

CoreDoc

#### 运行环境目录构成

假定工作目录在$workdir下，则$workdir下目录大致构成如下

+accountcenter 账号中心服务器名

+app 代码目录

+3rd 第三方库目录

+lua lua脚本目录

+conf 配置文件目录

+README.md 必读文档

+doc 文档目录

+tools 工具目录

+gamesrv\_1 游戏服务器名(这里是游戏1服)

+log 游戏日志目录

+shell 管理脚本目录

+skynet skynet引擎目录

+src 游戏代码目录

+base 框架共有代码目录

+gamelogic 游戏逻辑目录

+proto 游戏协议目录

+dbdata db数据及配置文件所在目录

+client 简易客户端（自行考虑是否检出）

$workdir目录详细构成建《目录说明.txt》

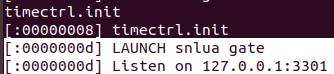
## API文档

# 学习笔记

【占用端口】

猎豹截图20181011152641

【3301】



gamesrv\_1 /src/gamelogic/config

gamesrv\_1 = "127.0.0.1:3301"

猎豹截图20181011153105

【重要文件】

src/base/service/gate/tcp\_protobuf.lua

tcp类型用protobuf编解码的gate服务

src/gamelogic/client/client.lua

管理所有客户端连接，客户端收发包

src/gamelogic/client/linkobj.lua

连线对象（对应单个套接字，登录后绑定玩家）

src/gamelogic/global/globalmgr.lua

全局管理器

src/gamelogic/net/login.lua

登录相关协议

src/gamelogic/game.lua

游戏入口文件

src/gamelogic/main.lua

游戏入口文件（使用game.lua模块)