

直播商城系统短视频带货app直播电商app直播购物app

产品名称	直播商城系统短视频带货app直播电商app直播购物app
公司名称	深圳市简点科技有限公司
价格	.00/个
规格参数	
公司地址	深圳市宝安区西乡街道前进二路航城工业区智汇创新中心B栋西509A
联系电话	13682566474

产品详情

直播源码,短视频,直播带货,游戏陪玩,仿比心,猎游,tt语音聊天,美女约玩,陪玩系统源码,游戏电竞开黑,约玩源码后台: laravel-admin 前端: VUE 移动: Android + ios

web演示地址 : <http://www.jinqianlive.com>

后台演示地址 : <http://www.jinqianlive.com/admin>

账号 : test 密码 : test

安卓下载 : <https://baoya.lanzous.com/icsdtxa> (用手机浏览器打开下载, 不要用微信直接下载)

IOS 视频演示 :

链接 : <https://pan.baidu.com/s/18KaHu-39TMQLetb0m7XD0Q> 提取码 : v929

文档地址 : <http://www.jinqianlive.com/appapi/listAllApis.php?type=expand>

系统截图

技术结构前端

集小视频/IM聊天/直播等功能于一体的直播项目。界面可仿制抖音|火山小视频|陌陌直播|比心陪玩等。

后端

系统开发语言

PHP 视频互动系统由 WEB 系统，REDIS 服务，MYSQL 服务，视频服务，聊天服务，后台管理系统和定时监控组成，后台管理采用 PHP 语言开发，所有服务提供横向扩展。

WEB 系统提供页面、接口逻辑。

REDIS 服务提供数据的缓存、存储动态数据。

MYSQL 服务提供静态数据的存储。

视频服务提供视频直播，旁路直播，转码、存储、点播等。

聊天服务提供直播群聊，私聊，消息通知等。

定时监控：监听主播异常掉线情况、直播消息推送等。

视频服务

直播配置

RTMP 服务添加一个 application 这个名字可以任意起，也可以起多个名字，由于是直播我就叫做它 live，如果打算弄多个序列的直播就可以 live_cctv。

```
#user nobody; worker_processes 1; #error_log logs/error.log; #error_log logs/error.log notice;
#error_log logs/error.log info; #pid logs/nginx.pid; events {
worker_connections 1024; } rtmp { #RTMP server server {
listen 1935; #server port chunk_size 4096; #chunk_size
# vod server application vod {
play /mnt/hgfs/dn_class/vod; #media file position }
# live server 1 application live{ #Darren live first add live on;
} # live server 2 application live_cctv{ #Darren live add
live on; } } } ..... 其他配置不需理会
```

在 Ubuntu 端用 ffmpeg 产生一个模拟直播源，向 rtmp 服务器推送

推流

```
ffmpeg -re -i /mnt/hgfs/dn_class/vod/35.mp4 -c copy -f flv rtmp://192.168.100.33/live/35
```

注意，源文件必须是 H.264+AAC 编码的

拉流

```
ffplay rtmp://192.168.100.33/live/35
```

点播配置

建立媒体文件夹 /mnt/hgfs/dn_class/vod 把媒体文件 35.mp4 复制到 /mnt/hgfs/dn_class/vod 目录下。然后我

们就可以开启一个视频点播的服务了。打开配置文件nginx.conf（路径/usr/local/nginx/conf/nginx.conf），添加RTMP的配置。

```
#user nobody; worker_processes 1; #error_log logs/error.log;
#error_log logs/error.log notice; #error_log logs/error.log info;
#pid logs/nginx.pid; events {
worker_connections 1024; } rtmp { #RTMP server
server { listen 1935; #server port
chunk_size 4096; #chunk_size application vod {
play /mnt/hgfs/dn_class/vod; #media file position } }
} ..... 其他配置不需理会
```

聊天服务

特性

轻量级

高性能

纯Golang实现

支持单个、多个、单房间以及广播消息推送

支持单个Key多个订阅者（可限制订阅者最大人数）

心跳支持（应用心跳和tcp、keepalive）

支持安全验证（未授权用户不能订阅）

多协议支持（websocket，tcp）

可拓扑的架构（job、logic模块可动态无限扩展）

基于Kafka做异步消息推送

安装一、安装依赖\$ yum -y install java-1.7.0-openjdk二、安装Kafka消息队列服务

kafka在网站已经描述的非常详细，在这里就不过多说明，安装、启动请查看[这里](#)。

三、搭建golang环境

1. 下载源码(根据自己的系统下载对应的安装包)

```
$ cd /data/programfiles $ wget -c --no-check-
certificate https://storage.googleapis.com/golang/go1.5.2.linux-amd64.tar.gz $ tar -xvf go1.5.2.linux-
amd64.tar.gz -C /usr/local
```

2. 配置GO环境变量 (这里我加在/etc/profile.d/golang.sh)

```
$ vi /etc/profile.d/golang.sh# 将以下环境变量添加到profile最后面export GOROOT=/usr/local/goexport
```

PATH=\$PATH:\$GOROOT/binexport GOPATH=/data/apps/go \$ source /etc/profile四、部署goim

1. 下载goim及依赖包

```
$ yum install hg $ go get -u github.com/Terry-Mao/goim $ mv $GOPATH/src/github.com/Terry-Mao/goim $GOPATH/src/goim $ cd $GOPATH/src/goim $ go get ./...
```

2. 安装router、logic、comet、job模块(配置文件请依据实际机器环境配置)

```
$ cd $GOPATH/src/goim/router $ go install $ cp router-example.conf $GOPATH/bin/router.conf $ cp router-log.xml $GOPATH/bin/ $ cd ../logic/ $ go install $ cp logic-example.conf $GOPATH/bin/logic.conf $ cp logic-log.xml $GOPATH/bin/ $ cd ../comet/ $ go install $ cp comet-example.conf $GOPATH/bin/comet.conf $ cp comet-log.xml $GOPATH/bin/ $ cd ../logic/job/ $ go install $ cp job-example.conf $GOPATH/bin/job.conf $ cp job-log.xml $GOPATH/bin/
```

到此所有的环境都搭建完成！

五、启动goim

```
$ cd /$GOPATH/bin $ nohup $GOPATH/bin/router -c $GOPATH/bin/router.conf 2>&1 > /data/logs/goim/panic-router.log & $ nohup $GOPATH/bin/logic -c $GOPATH/bin/logic.conf 2>&1 > /data/logs/goim/panic-logic.log & $ nohup $GOPATH/bin/comet -c $GOPATH/bin/comet.conf 2>&1 > /data/logs/goim/panic-comet.log & $ nohup $GOPATH/bin/job -c $GOPATH/bin/job.conf 2>&1 > /data/logs/goim/panic-job.log &
```

如果启动失败，默认配置可通过查看panic-xxx.log日志文件来排查各个模块问题。

六、测试Arch

Benchmark Server

CPU Memory OS Instance Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz DDR3 32GB Debian GNU/Linux 81 Benchmark Case

Online: 1,000,000

Duration: 15min

Push Speed: 40/s (broadcast room)

Push Message: {"test":1}

Received calc mode: 1s per times, total 30 times

Benchmark Resource

CPU: 2000%~2300%

Memory: 14GB

GC Pause: 504ms

Network: Incoming(450MBit/s), Outgoing(4.39GBit/s)

Benchmark Result

Received: 35,900,000/s

推送协议可查看push http协议文档

配置

TODO

例子

Websocket: Websocket Client Demo

Android: Android

iOS: iOS

文档

push http协议文档推送接口

集群comet

comet 属于接入层，非常容易扩展，直接开启多个comet节点，修改配置文件中的base节点下的server.id改成不同值（注意一定要保证不同的comet进程值唯一），前端接入可以使用LVS 或者 DNS来转发

logic

logic 属于无状态的逻辑层，可以随意增加节点，使用nginx upstream来扩展http接口，内部rpc部分，可以使用LVS四层转发

kafka

kafka 可以使用多broker，或者多partition来扩展队列

router

router 属于有状态节点，logic可以使用一致性hash配置节点，增加多个router节点（目前还不支持动态扩容），提前预估好在线和压力情况

job

job 根据kafka的partition来扩展多job工作方式，具体可以参考下kafka的partition负载

使用PHP+Swoole实现的网页即时聊天工具，

全异步非阻塞Server，可以同时支持数百万TCP连接在线

基于websocket+flash_websocket支持所有浏览器/客户端/移动端

支持单聊/群聊/组聊等功能

支持永久保存聊天记录，使用MySQL存储

基于Server PUSH的即时内容更新，登录/登出/状态变更/消息等会内容即时更新

用户列表和在线信息使用Redis存储

支持发送连接/图片/语音/视频/文件

支持Web端直接管理所有在线用户和群组

|安装|

swoole扩展

```
pecl install swoole
```

swoole框架

```
composer install运行
```

将webroot目录配置到Nginx/Apache的虚拟主机目录中，使webroot/可访问。

详细部署说明

1. 安装composer(PHP依赖包工具)

```
curl -sS https://getcomposer.org/installer | php mv composer.phar /usr/local/bin/composer
```

注意：如果未将php解释器程序设置为环境变量PATH中，需要设置。因为composer文件第一行为#!/usr/bin/env php，并不能修改。更加详细的对composer说明：<http://blog.csdn.net/zzulp/article/details/18981029>

2. composer install

切换到PHPWebIM项目目录，执行指令composer install，如很慢则

```
composer install --prefer-dist
```

3. Nginx配置

这里未使用swoole_framework提供的Web AppServer

Apache请参照Nginx配置，自行修改实现

这里使用了im.swoole.com作为域名，需要配置host或者改成你的域名

```
server { listen 80; server_name im.swoole.com;
index index.html index.php; location / {
```

```
root /path/to/webim/webroot; proxy_set_header X-Real-IP $remote_addr;
if (!-e $request_filename) { rewrite ^/(.*)$ /index.php; }
} location ~ .*\.php|php5)?$ { fastcgi_pass 127.0.0.1:9000;
fastcgi_index index.php; include fastcgi.conf; }
```

****注意：**https下必须采取wss So-有两种方案 1.采用nginx 反向代理4431端口 swoole的端口和4431进行通讯。2.swoole 确认是否启用了openssl，是否在编译时加入了--enable-openssl的支持,然后在set 证书路径即可。两种方案选择其一就好，不过第一种方案有个潜在神坑就是你通过反向代理拿不到真实的IP地址了,这点值得注意，Nginx有办法拿到真实的ip，不懂可以私聊我，光wss的坑太多了就不一一说了。****4. 修改配置**

配置configs/db.php中数据库信息，将聊天记录存储到MySQL中

配置configs/redis.php中的Redis服务器信息，将用户列表和信息存到Redis中

表结构

```
CREATE TABLE `webim_history` ( `id` int(11) NOT NULL AUTO_INCREMENT, `addtime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP, `name` varchar(64) COLLATE utf8mb4_bin NOT NULL, `avatar` varchar(255) COLLATE utf8mb4_bin NOT NULL, `type` varchar(12) COLLATE utf8mb4_bin NOT NULL, `msg` text COLLATE utf8mb4_bin NOT NULL, `send_ip` varchar(20) COLLATE utf8mb4_bin, PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_bin
```

修改configs/webim.php中的选项，设置服务器的URL和端口

```
$config['server'] = array( //监听的HOST 'host' => '0.0.0.0', //监听的端口
'port' => '9503', //WebSocket的URL地址，供浏览器使用的
'url' => 'ws://im.xxx.com:9503', //用于Comet跨域，必须设置为web页面的URL
//比如你的网站静态页面放在 http://im.xxx.com:8888/main.html
//这里就是 http://im.xxx.com:8888 'origin' => 'http://im.xxx.com:8888', );
```

server.host server.port 项为WebIM服务器即WebSocket服务器的IP与端口，其他选择项根据具体情况修改

server.url对应的就是服务器IP或域名以及websocket服务的端口，这个就是提供给浏览器的WebSocket地址

server.origin为Comet跨域设置，必须修改origin才可以支持IE等不支持WebSocket的浏览器

5. 启动WebSocket服务器

```
php server.php start
```

IE浏览器不支持WebSocket，需要使用FlashWebSocket模拟，请修改flash_policy.php中对应的端口，然后启动flash_policy.php。

```
php webim/flash_policy.php
```

6. 绑定host与访问聊天窗口（可选）

如果URL直接使用IP:PORT，这里不需要设置。

vi /etc/hosts快速了解项目架构

1.目录结构

+ webim |- server.php //WebSocket协议服务器 |+ swoole.ini // WebSocket协议实现配置
|+ configs //配置文件目录 |+ webroot |+ static
|- config.js // WebSocket配置 |+ log // swoole日志及WebIM日志
|+ src // WebIM 类文件储存目录 |+ Store |- File.php // 默认用内存tmpfs
文件系统(linux /dev/shm)存放天着数据，如果不是linux请手动修改\$shm_dir
|- Redis.php // 将聊天数据存放到Redis
|- Server.php // 继承实现WebSocket的类，完成某些业务功能 |+ vendor // 依赖包目录

2.Socket Server与Socket Client通信数据格式

如：登录

Client发送数据

```
{"cmd":"login","name":"xdy","avatar":"http://tp3.sinaimg.cn/1586005914/50/5649388281/1"}
```

Server响应登录

```
{"cmd":"login","fd":"31","name":"xdy","avatar":"http://tp3.sinaimg.cn/1586005914/50/5649388281/1"}
```

可以看到cmd属性，client与server发送时数据都有指定，主要是用于client或者server的回调处理函数。

3.需要理清的几种协议或者服务的关系

http协议：超文本传输协议。单工通信，等着客户端请求之后响应。

WebSocket协议：是HTML5一种新的协议，它是实现了浏览器与服务器全双工通信。服务器端口与客户端都可以推拉数据。

Web服务器：此项目中可以用基于Swoole的App
Server充当Web服务器，也可以用传统的nginx/apache作为web服务器

Socket服务器：此项目中浏览器的WebSocket客户端连接的服务器，swoole_framework中有实现WebSocket
协议PHP版本的服务器。

WebSocket

Client：实现html5的浏览器都支持WebSocket对象，如不支持此项目中有提供flash版本的实现。

后台一键安装 直接访问入admin

金钱豹直播QQ2306490558，我们团队成员全部拥有多年视频直播领域项目的开发经验，是从创业团队转型过来，所以深知创业时期对开发的需求，本着靠谱，踏实，专业，用心，快速和超高性价比来从事视频直播领域项目的源代码以及二次开发。金钱豹团队主要开发以下视频直播项目：

直播app直播带货app视频直播源码直播购物app直播商城app

直播商城系统短视频带货app直播电商app直播购物app

直播陪玩app直播陪练源码猎游app皮皮约玩泡泡陪玩游戏陪练约玩

仿比心陪练源码比心陪练app二次开发直播陪玩比心陪玩app游戏陪玩

直播约会直播伴游app二次开发

直播相亲一对一视频交友源码二次开发

视频直播带货商城小程序开发直播带货小程序