**JamesSoft 技术有限公司**

**JamesSoft DaSheng System(JDS)**

**安装手册**

**2020年5月27日**

声 明

Copyright

JamesSoft 技术有限公司

版权所有，保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本书内容的部分或全部，并不得以任何形式传播。

由于产品版本升级或其他原因，本文内容会不定期进行更新。除非另有约定，本文仅作为使用指导，本文中的所有陈述、信息和建议不构成任何明示或暗示的担保。

**前 言**

**版本说明**

本手册对应产品版本为：1.0以及以上版本

**文档简介**

该文档主要描述PIS地面系统的安装过程。

**文档修订历史**

|  |  |  |
| --- | --- | --- |
| **日 期** | **修订说明** | **修订人** |
|  |  |  |
|  |  |  |
|  |  |  |

**目 录**

[1. 系统环境要求 6](#_Toc514253732)

[1.1. 硬件环境 6](#_Toc514253733)

[1.2. 软件环境 6](#_Toc514253734)

[2. 系统软件环境准备 7](#_Toc514253735)

[2.1. 修改root用户密码 7](#_Toc514253736)

[2.2. 静态IP设置 7](#_Toc514253737)

[2.3. 主机名称设置 8](#_Toc514253738)

[2.4. 远程连接 8](#_Toc514253739)

[2.5. Samba服务 9](#_Toc514253740)

[2.6. 安装目录 10](#_Toc514253741)

[2.7. 系统软件安装 11](#_Toc514253742)

[2.7.1. 自动安装 11](#_Toc514253743)

[2.7.2. 安装web服务 11](#_Toc514253744)

[2.7.3. 安装缓存服务 13](#_Toc514253745)

[2.7.4. 安装FTP服务 13](#_Toc514253746)

[2.7.5. 安装NTP服务 13](#_Toc514253747)

[2.7.6. PostgreSQL 数据库 14](#_Toc514253748)

[2.7.7. python3相关模块 15](#_Toc514253749)

[2.7.8. 文件上传/下载 16](#_Toc514253750)

[2.7.8.1. 安装nginx依赖库 16](#_Toc514253751)

[2.7.8.2. 配置nginx 17](#_Toc514253752)

[2.7.8.3. 启动nginx 17](#_Toc514253753)

[2.8. PIS业务 18](#_Toc514253754)

[2.8.1. 用户创建 18](#_Toc514253755)

[2.8.2. 创建文件上传及下载目录 18](#_Toc514253756)

[2.9. IOT业务 19](#_Toc514253757)

[2.9.1. Socket 配置 20](#_Toc514253758)

[2.9.2. IOT配置 20](#_Toc514253759)

[3. YIMS配置及启动 20](#_Toc514253760)

[3.1. 创建数据库 20](#_Toc514253761)

[3.2. 后端参数配置 21](#_Toc514253762)

[3.3. 前端参数配置 23](#_Toc514253763)

[3.4. 系统启动 25](#_Toc514253764)

[4. 附录：离线安装 25](#_Toc514253765)

[4.1. apt-get资源包 26](#_Toc514253766)

[4.1.1. 软件包生成 26](#_Toc514253767)

[4.1.2. 软件包安装 26](#_Toc514253768)

[4.1.3. 软件包的新增或删除 26](#_Toc514253769)

[4.2. Python软件包安装 27](#_Toc514253770)

**JamesSoft 技术有限公司**

**JamesSoft DaSheng System(JDS)**

**安装手册**

# 系统环境要求

## 硬件环境

PIS系统面向地铁运营商使用，最大用户数一般不超过20000用户，并发在线连接数一般不会超过1000，对服务器性能要求不高。

* 服务器：PC 服务器双机（立式或刀片式均可），E5或以上，32G内存，带磁盘阵列，硬盘容量在2T\*2 以上。
* 网络：需要有固定IP地址。
* 防火墙：由现网提供。若现网没有，则需防火墙两台。

如果需要进行双网双平面以确保系统24小时在线，则需要服务器双机，LanSwitch双机，防火墙双机等。

## 软件环境

操作系统：Ubuntu Server 14.04.1

Python版本：python3.4

数据库：PostgreSQL 9.3

Web 服务器：nginx：

# 系统软件环境准备

## 修改root用户密码

在刚刚安装了Ubuntu后， 是不能使用root登录的。在使用安装期间创建的用户名登录后，可以使用命令sudo修改root用户的密码：

sudo passwd

修改密码后，就可以使用root账号登录。

## 静态IP设置

编辑 /etc/network/interfaces，将dhcp 一行屏蔽，

# The primary network interface

auto eth0

#iface eth0 inet dhcp

添加和静态ip有关的参数：

# The primary network interface

iface eth0 inet static

netmask 255.255.255.0

address 192.168.0.10

gateway 192.168.0.1

其中，IP地址、子网掩码、网关地址等根据实际情况修改。

编辑 /etc/resolvconf/resolv.conf.d/head：

nameserver 127.0.0.1

nameserver 192.168.0.1

nameserver 8.8.8.8

重启：

ifdown eth0

ifup eth0

此外，如果是在虚拟机上安装系统，需要注意虚拟机的网络连接方式。在使用vmware workstation pro时，网络连接方式为桥接模式。

## 主机名称设置

编辑 /etc/hostname文件，设置主机名称：

cd /etc

vi hostname

此文件只有一行，内容为主机名称。

编辑 /etc/hosts文件，修改对应的主机名称：

cd /etc

vi hosts

ip\_address hostname

其中，ip\_address为设置的IP地址，hostname为设置的主机名。

## 远程连接

一般而言，服务器安装完成后，不会直接在服务器上操作，而是通过远程连接软件连接到服务器上进行操作，如SecureCRT等，为此，需要安装ssh。

以root用户登录，使用命令：apt-get -y install ssh openssh-server

检查ssh服务是否已经启动：

ps -e |grep ssh

如果有sshd说明ssh服务已经启动。如果没有启动，输入service ssh start命令启动ssh服务。

为了能够使用root用户进行操作，需要对配置文件/etc/ssh/sshd\_config进行更改，将PermitRootLogin值改为yes，然后重新启动服务：

service ssh restart

## 自动挂载远程主机目录到本地目录

有时候在开发时为了方便，需要挂载远程主机的目录到本地。

在本地主机的操作：

1. 安装sshfs： apt-get -y install sshfs
2. 生成密钥，路径在 ./.ssh：ssh-keygen -t rsa
3. 拷贝公钥到远程服务器：ssh-copy-id username@remote\_ip
4. 在需要时进行挂载： sshfs username@remode\_ip:/xxx/xxx /xxx/xxx

卸载时，可以使用 sudo umount /xxx/xxx命令，也可以使用：fusermount -u /xxx/xxx命令。

## Samba服务

Samba服务可以使得Ubuntu将指定目录共享。

**在正式安装时，不需要安装此服务。**

首先下载软件包安装：

apt-get -y install samba samba-common libtalloc2

创建准备共享的目录，在这里为 /JamesSoft，或其他目录需要共享的目录，并修改存取权限：

chmod –R 777 /JamesSoft

打开 /etc/samba/smb.conf文件，

新增一行

security = user

表示需要输入用户名和密码才能访问。

在文件的最后，新增以下内容：

[JamesSoft]

comment = JamesSoft

path = /JamesSoft

public = yes

writable = yes

valid users = root

create mask = 0766

directory mask = 0777

指定共享的目录是 /JamesSoft。

指定用户的使用密码：

smbpasswd –a username

username 是使用smb的用户名称。

最后，重新启动Samba服务：

service smbd restart

验证服务是否正常的方法：在资源管理器中输入服务器地址，双击看到的JamesSoft目录，输入用户名、密码，验证通过即可使用。

## 安装目录

在根目录/下创建目录JamesSoft：

mkdir /JamesSoft

将版本压缩包JDS.tar传到这个目录中，并使用tar命令解开压缩包:

tar –xvf JDS.tar

此时，得到一个JDS目录。在JDS目录中，包含的目录有：

* bin： 应用系统命令，
* config：应用系统配置文件，都是链接，不是最终文件。
* log：日志文件存放目录，暂未使用。
* src：JDS 版本目录。
* sql：数据库相关SQL文件。
* temp：临时目录

在目录 /JamesSoft 下创建目录 software，并将相关系统软件下载到此目录。

## 系统软件安装

系统软件包括web服务器nginx，数据库PostgreSQL，缓存服务器Redis，Python等。

安装系统软件前请确保服务器已经可以访问Internet。如果需要进行离线安装，请参考《附录：离线安装指南》。

以下软件的安装操作，如果没有特别说明，均是以root用户进行。

**注意：**

因系统软件版本升级的问题，使得在 /JamesSoft/software 目录中随版本发布的系统相关软件包不是最新版本，需要根据需要，自行到相关网站上下载最新版本。

而且随版本提供的系统软件包文件名也因版本更新而可能和安装手册中提到的文件名有差异。

### 自动安装

进入 /JamesSoft/JDS/install目录，使用 install.sh 脚本自动进行安装：

cd /JamesSoft/JDS/install

./install.sh

自动安装会完成相关系统软件、数据库软件、数据库创建等所有工作。基本上，安装时间需要半小时左右。在安装过程中，需要关注可能出现的错误信息。

以下安装说明为手动安装过程。

### 安装web服务

以root用户登录，使用命令：apt-get -y install nginx进行安装。

使用vi编辑/etc/nginx/sites- available/default，增加服务项：

server

{

listen 448;

server\_name default;

root /JamesSoft/JDS/src/app;

index index.html index.htm;

autoindex on;

autoindex\_exact\_size off;

autoindex\_localtime on;

#ssl on;

#ssl\_certificate /JamesSoft/cert/cdJamesSoft.tech.crt;

#ssl\_certificate\_key /JamesSoft/cert/cdJamesSoft.tech.key.unsecure;

# Only used for PIS service

location ^~ /screen {

alias /home/controller/screen/;

}

location ^~ /tmp1 {

#proxy\_pass http://localhost:5555/tmp1;

alias /usr/local/nginx/html/tmp1/;

}

location /upload

{

proxy\_pass http://localhost:5555/upload;

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header Host $http\_host;

}

}

保存后退出，并重启web服务：service nginx restart

注意：

* 监听端口448可以更改为其他端口，用户在浏览器上使用系统时，需要输入：http://server\_addr:448。
* Root路径：在前面已经新增了目录JDS，如果有变化，可以根据实际目录进行调整。
* 此处的配置，和后面文件上传服务时对nginx的配置是一致的。

### 安装缓存服务

以root用户登录，使用命令apt-get -y install redis-server安装。

安装完成后，更改密码，编辑 /etc/redis/redis.conf，并分别找到以下两行，设置密码123456

masterauth 123456

requirepass 123456

然后重新启动服务：

service redis-server restart

### 安装FTP服务

FTP服务由控制器使用，用于上传日志、截屏图片及下载节目单、版式文件、版本文件等。

以root用户登录，使用命令apt-get -y install vsftpd 安装。

安装完成后，编辑 /etc/vsftpd.conf文件，设置如下属性

anonymous\_enable=NO #禁止匿名访问

local\_enable=YES

write\_enable =YES

启动vsftpd服务器

service vsftpd start

如果需要root用户也能够登录FTP服务，需要修改 /etc/ftpuser文件，将 root 注释掉。

### 安装NTP服务

NTP服务用于控制器同步中心服务器时钟。

以root用户登录，使用命令apt-get -y install ntp安装。

安装完成后，编辑 /etc/ntp.conf文件，在“restrict -6 default kod notrap nomodify nopeer noquery limited”后面添加一行：

restrict 192.168.0.0 mask 255.255.255.0 nomodify

其中，192.168.0.0 为网段，255.255.255.0为子网掩码，nomodify表示Client 端不能更改 Server 端的时间参数，不过Client端仍然可以透过Server 端來进行网络校时。

启动ntp服务器：

service ntp restart

### PostgreSQL 数据库

以root用户登录，使用命令apt-get -y install postgresql安装数据库。

修改默认的账户和密码为：postgres/postgres：

su postgres

psql -U postgres

进入PostgreSQL的命令行模式，输入命令：

alter user postgres with password 'postgres';

执行完成后，输入“\q”退出命令行模式，然后执行“exit”退出。

打开 /etc/postgresql/10/main/postgresql.conf，修改：

#listen\_addresses = 'localhost'

listen\_addresses = '\*'

打开 /etc/postgresql/10/main/pg\_hba.conf，在文件最后新增一行：

host all all all md5

然后重新启动服务：

service postgresql restart

说明：

具体路径，要看PostgresSQL的版本。最新版本是10。

### python3相关模块

以root用户登录，分别进行以下安装：

apt-get -y install python3

apt-get -y install python3-dev

apt-get -y install python3-pip

apt-get -y install python3-pil

apt-get -y install python3-msgpack

apt-get -y install libjpeg8-dev

apt-get -y install zlib1g-dev

apt-get -y install libfreetype6-dev

apt-get -y install libpq-dev

apt-get -y install libmysqlclient-dev

pip3 install psycopg2

pip3 install psycopg2-binary

pip3 install pycrypto

pip3 install tornado

pip3 install ujson

pip3 install redis

pip3 install apscheduler # 任务定时调度

pip3 install reportlab #

pip3 install wechatpy # 微信

pip3 install pydicom # 短信猫

pip3 install xlrd # excel支持

pip3 install xlwt # excel xls支持

pip3 install pypcap

pip3 install dpkt

pip3 install openpyxl # excel xlsx支持

add-apt-repository ppa:kirillshkrogalev/ffmpeg-next

apt-get update

apt-get -y install ffmpeg

安装mysql & connector：

cd /JamesSoft/software

pip3 install ConfigParser

pip3 install mysql

wget <http://dev.mysql.com/get/Downloads/Connector-Python/mysql-connector-python-2.1.1.tar.gz>

tar xvf mysql-connector-python-2.1.1.tar.gz

cd mysql-connector-python-2.1.1/

python3 setup.py install

如果通过wget不能进行安装，可以使用随版本附带的软件包进行安装。软件包位置在：/JamesSoft/software/目录下。

### 文件上传/下载

进入到/JamesSoft/software/fileupload目录中，将fileupload.tar.gz打开压缩包：

cd /JamesSoft/software/fileupload/

tar zxvf fileupload.tar.gz

得到以下文档及目录

nginx.conf

nginx-1.4.7/

nginx\_upload\_modul-2.2/

nginx-upload-progress-module-master/

然后将压缩包删除：rm fileupload.tar.gz。

#### 安装nginx依赖库

依赖库包括openssl、pcre、zlib、libssl-dev等：

apt-get -y install libpcre3 libpcre3-dev zlib openssl libssl-dev

检查是否需要打补丁：

patch ngx\_http\_upload\_module.c davromaniak.txt

/JamesSoft/software/fileupload/nginx-1.4.7目录中执行下面语句：

cd /JamesSoft/software/fileupload/nginx-1.4.7

./configure --prefix=/usr/local/nginx --with-openssl=/usr/include/openssl --add-module=/JamesSoft/software/fileupload/nginx-upload-module-2.2 --add-module=/JamesSoft/software/fileupload/nginx-upload-progress-module-master

注意：上面的命令中涉及到的路径，如/usr/local/nginx、/usr/include/openssl 、/JamesSoft/software/fileupload/nginx-upload-module-2.2、/JamesSoft/software/fileupload/nginx-upload-progress-module-master等，需要根据实际安装路径来修改。

最后进行安装：

make

make install

#### 配置nginx

下载服务使用到的配置文件目录为：/usr/local/nginx/conf/nginx.conf。

把 /JamesSoft/software/fileupload/nginx.conf 复制到nginx目录中：

cd

cp /JamesSoft/software/fileupload/nginx.conf /usr/local/nginx/conf

在这个文件中，需要注意：

文件上传监听端口 5555

文件上传大小最大值client\_max\_body\_size 2048m;

proxy\_pass http://localhost:92;

以上参数需要和保持一致。其中，端口92为前台和后台使用的非SSL端口，在前台及后台的配置文件中需要保持一致。

进入到/usr/local/nginx/html，创建上传文件存储目录：

mkdir tmp1

cd tmp1

mkdir 0 1 2 3 4 5 6 7 8 9 temp

chmod –R 777 /usr/local/nginx/html/tmp1

#### 启动nginx

转到/usr/local/nginx/sbin目录下，启动nginx：

./nginx

如果需要重新启动，则执行命令：

./nginx -s reload

## PIS业务

### 用户创建

控制器需要远程连接到服务器上来上传或下载相关文件，需要给控制器创建用户 controller及相应的目录，在以root登录后：

adduser controller

输入用户密码，默认为 controller。

在该目录下，会有以下目录：

list

screen

log

formatfile

update

videofile

控制器到这几个目录中来下载或上传相关文件。

### 创建文件上传及下载目录

创建目录：

su – controller

mkdir list formatfile log screen update videofile

这6个目录分别用于存放节目单文件、版式文件及背景图片、上传的日志文件、截屏文件、控制器版本更新版式文件、节目文件。

其中，list目录结构为：

* 节目单文件名为：program\_list\_id.txt，id为节目单ID，节目文件单独存放在videofile目录。

版式文件为单一文件，存放在formatfile目录。版式文件背景图片也存放在这个目录。在系统安装时背景文件已经存放到该目录。如果需要更新版式文件的背景图片，需要将相关文件传入到该目录。

log目录为存放控制器上传的日志，其目录结构为：

* 按天创建目录，如20180101。
* 每个控制器单独创建一个目录，目录名称为控制器编号，规则见接口规范。
* 控制器每天上传日志文件，文件名为：控制器编号-YYYYMMDD-HHMMSS.log。控制器编号规则见接口规范，YYYYMMDD-HHMMSS为日志产生的日期及时间，不是上传日期。
* 日志文件格式见接口规范。

screen 目录存放控制器截屏文件，其目录结构为：

* 按天创建目录，如20180101。
* 每个控制器单独创建一个目录，目录名称为控制器编号，规则见接口规范。
* 在每个控制器的目录下，存放当天产生的截屏文件。文件类型为 JPG 文件，文件名称为截屏时的时间，控制器编号-YYYYMMDD-HHMMSS.jpg。
* 截屏文件只保留三个月时间，超过三个月的截屏文件由维护人员导出其其他系统并删除，以避免占用存储空间。

videofile目录存放上传的节目文件。系统会定期清理未使用的节目文件，清理规则：在三个月内的节目单都没有使用到的节目文件将被删除。

## IOT业务

IOT业务为定位模块业务。

IOT业务的配置涉及到2部分：socket配置、iot配置

### Socket 配置

Socket配置为使用socket消息进行通信的业务共同使用的配置。配置文件在 libs/socketMsg/messageConfig.py文件中。

基本上，此文件中的参数可以不用修改。

### IOT配置

Iot配置文件在iot/iotConfig.py文件中。参数包括：

# 服务端IP地址

serverAddr = "192.168.0.201"

# 服务端监听端口，接收定位模块上报的位置信息

serverPort = 11615

# 百度地图App Key

appKey = 'sEKdIjahGM3zyXoXx2FSvVbT7NhcSrrT'

# socket连接模式

SOCKET\_TYPE = SOCK\_DGRAM

服务端的地址、端口需要根据实际配置进行修改。百度地图的AppKey需要根据最终用户申请的APPKey进行修改。

Socket连接模式为TCP连接或UDP连接。TCP连接则值为SOCK\_STREAM，UDP连接则为SOCK\_DGRAM。此值默认为UDP连接。

另外，在iot/iotMessage.py中，还有关于IOT消息的配置，此部分配置可以不用修改，使用默认配置即可。

# JDS配置及启动

## 创建数据库

执行以下命令开始创建数据库：

cd /JamesSoft/JDS/install

chmod 777 installdb.sh

export WORKDIR=/JamesSoft

export SRVDIR=$WORKDIR/JDS

su postgres –s ./installdb.sh

如果出现错误，请确保目录/JamesSoft/JDS/install 是否权限正确，以及postgres用户是否具有shell脚本installdb.sh 有可执行权限。

**注意：**

因数据库经常变动，有可能对数据库的改动没有在sql文件中修改，导致数据库虽然安装没有出错，但是在业务执行中出现错误。因此，在安装数据库去前，请确保数据库的表是否完全，表结构是否有变动，数据字典的数据是否为最新数据。

数据库创建也可以使用相应的软件工具将现有数据库备份后然后在新的服务器上恢复的方式进行创建。

## 后端参数配置

以root登录，使用vi编辑/JamesSoft/JDS/src /service/config.py，配置后端参数，参数可能如下：

#coding:utf-8

import os

App\_Port = 92

App\_PortSSL = 446

App\_Key = 'fb98ab9159f51fd2'

App\_Secret = '09f7c8cba635f7616bc131b0d8e25947s'

UserAuthTimeout = 3600\*5

UserAuthOneSession = False

#标记系统业务部件，用于某些业务进行判断

SYSTEM = ['PIS', 'IOT']

#文件服务器url基路径

FileUrlRoot=""

#文件服务器文件保存根路径

FileSavePathRoot="/usr/local/nginx/html"

#默认头像地址,相对于网页路径

DefaultAvatar="/resourses/images/default.png"

# PIS业务上传下载文件的根目录

PISFileRoot="/home/controller"

PostgresqlDbConfig = {

'minconn' : 1,

'maxconn' : 50,

'host' : 'localhost',

'database' : 'JDS',

'user' : 'postgres',

'password' : 'postgres',

'port' : 5432,

'datatype' : 'POSTGRESQL'

}

AppRedisConfig = {

'host' : 'localhost',

'port' : 6379,

'db' : 12,

'password' : '123456'

}

PisConfig = {

"formatSwitchTime" : 10,

"snatchScreenTime" : 10,

"packetDebug" : False,

"backupAfterDays" : 7,

"stationServer" : False,

"stationCode" : "N-000012"

}

DbConfig = PostgresqlDbConfig

RedisConfig = AppRedisConfig

DefaultCacheExpireTime = 60\*60

DcmConfig = {

'qureyConfig' : 'localhost 11112 -aec DCM4CHEE',

'revConfig' : '-aem GETDCM --port 12345'

}

TableToRedisNo = {

'tbProductList' : '08',

'tbAdSense' : '09',

'tbMsg' : '10',

'tbUser' : '12'

}

DEBUG = True

# 调试级别：1：必须输出的信息， 2：调试信息，3：调试信息+数据

Debug\_Level = 3

配置文件随着开发的进展，会有相应的变化。但最主要的参数主要是：

* 端口：这个端口是指前台和后台内部通讯使用的端口，和nginx中配置的端口不同。
* 数据库名称：根据实际情况配置。
* 证书文件路径：根据实际情况配置。

相关端口都可以保留现有配置，需要注意的是涉及到的相关参数的路径，需要和实际安装的路径保持一致。

## 前端参数配置

登录后使用vi 编辑/JamesSoft/JDS/src/app/config.js：

var projName = "pis";

var SYSTEM = ['SystemManage', 'PIS', 'IOT'];

var webSvcPort = ("https:" == document.location.protocol) ? 446 : 92;

var \_baseUrl = location.protocol + "//" + location.hostname;

var projConfigs = {

"pis" : {

pis : true,

projName : "PIS",

projNameZh : "元钧PIS管理系统",

ownerName : "成都轨道交通有限公司",

ownerLogoURL : "resourses/images/logo.jpg",

ownerLoginURL : "resourses/images/logo.jpg",

ownerLogoPrint : "resourses/images/logo.jpg",

navLeftWidth : "35%",

navRightWidth : "65%",

navPicWidth : "120px",

pageSize : 20,

},

};

var projConfigs = {

"IOT" : {

IOT : true,

projName : "IOT",

projNameZh : "元钧物联网管理系统",

ownerName : "新誉集团",

ownerLogoURL : "resourses/images/logo.jpg",

ownerLoginURL : "resourses/images/logo.jpg",

ownerLogoPrint : "resourses/images/logo.jpg",

navLeftWidth : "35%",

navRightWidth : "65%",

navPicWidth : "120px",

pageSize : 20,

},

};

//js代码通过projConfig获取项目相关的配置项

//也可以在代码判断当前为何项目

var projConfig = projConfigs[projName];

//js代码通过appConfig获取应用相关的通用配置

var appConfig = {

appKey : "8uhnIKOLpl",

appSec : "09f7c8cba635f7616bc131b0d8e25947s",

webSvcUrl : \_baseUrl + ":" + webSvcPort + "/pis/",

projConfig : projConfig,

system : SYSTEM,

sysMap : (function () {

var maps = {};

for (var i = 0; i < SYSTEM.length; i++) {

maps[SYSTEM[i]] = true;

}

return maps;

})()

};

var getBaseUrl = function (){

return appConfig.webSvcUrl;

};

//文件服务器地址

window.uploadURL = location.origin + "/upload";

//头像上传地址

window.avatarURL = appConfig.webSvcUrl + "avatarUpload";

前端配置文件也会随着需求的变化而变化。需要注意的是端口配置，和后端保持一致即可。

业主名称及Log文件根据实际用户的情况来更新相关图标文件。

## 系统启动

进入/JamesSoft/JDS/bin 目录，使用命令：

cd /JamesSoft/JDS/bin

./JDS.sh

也可以进入 / JamesSoft/JDS/src/service目录，执行：

python main.py

这样可以直接看到应用的调试输出信息。

可以使用命令 JDS stop来停止系统运行。

# Docker安装

安装 apt 依赖包：

apt-get install -y apt-transport-https ca-certificates curl gnupg-agent software-properties-common

添加 Docker 的官方 GPG 密钥：

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

设置稳定版仓库：

add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

安装最新版本的 Docker Engine-Community 和 containerd：

apt-get install docker-ce docker-ce-cli containerd.io

# 附录：离线安装

在离线安装时，需要分别安装两部分软件包，一部分是通过apt-get安装的软件包，一部分是python安装的软件包。

## apt-get资源包

apt-get install安装软件时，下载的deb文件都会存放在/var/cache/apt/archives/下，为了今后重装或给别人用，特别是没有网络时，这些deb文件实际上是可以派上用途的，也就是可以用这些deb制作成本地源。

现在已经将需要的相关软件包整合成了一个压缩的tar文件，并由相应的脚本，可以在服务器上一键安装所有需要的软件包。

### 软件包生成

确认服务器已经连接到网络，然后在/JamesSoft/software/offline/system目录中，执行脚本：

./makeOfflinePackage.sh

通过这个命令可以生成离线资源包，包括：

offlinePackage.tar.gz

sources.list

### 软件包安装

在新的虚拟机上，将生成的离线资源包及其他脚本文件setOfflinePackage.sh、setup.sh、sources.list拷贝到/JamesSoft/software/offline/system中执行：

./setOfflinePackage.sh

这一步是更新apt-get的源为本地源。然后执行：

./setup.sh

可以将需要的资源包都安装完成。

### 软件包的新增或删除

如果需要新增资源包，可以编辑makeOfflinePackage.sh、setup.sh两个文件，将相应的软件包名称添加上去，重新执行一次生成和安装过程即可。

## Python软件包安装

在/JamesSoft/software/offline/python目录中，包含了需要的Python软件包资源。进入到这个目录后，执行：

./setup.sh

可以安装Python相关的软件包。

需要注意的是，有些软件包的安装是需要依赖其他软件包的，这些被依赖的软件包需要被先安装才可以。通常，需要先进行安装的软件包有：

backports\_abc-0.5

python-dateutil-2.6.0

tzlocal-1.3

pytz-2016.10

optionaldict-0.1.1

requests-2.7.0

setuptools\_scm-1.15.0

xmltodict-0.9.2

six-1.9.0

如果需要新增或删除相关的软件包，可以在setup.sh中进行调整。