

用戶手冊

目录

1、需求分析.....	错误! 未定义书签。
2、概要设计.....	错误! 未定义书签。
2.1 系统架构设计.....	错误! 未定义书签。
2.2 软件结构设计.....	错误! 未定义书签。
2.2.1 概述.....	错误! 未定义书签。
2.2.2 前端结构设计.....	错误! 未定义书签。
2.2.3 后端结构设计.....	错误! 未定义书签。
3、详细设计.....	错误! 未定义书签。
3.1 运行环境.....	错误! 未定义书签。
3.1.1 软件环境.....	错误! 未定义书签。
3.2 开发环境.....	错误! 未定义书签。
3.2.1 软件环境.....	错误! 未定义书签。
3.3 界面设计.....	错误! 未定义书签。
3.3.1 界面展示.....	错误! 未定义书签。
4、测试报告.....	错误! 未定义书签。
4.1 后端单元测试.....	错误! 未定义书签。
4.1.1 测试方式.....	错误! 未定义书签。
4.2.2 测试结果:	错误! 未定义书签。
4.2 后端集成测试.....	错误! 未定义书签。
4.2.1 测试方式.....	错误! 未定义书签。
4.1.2 测试结果:	错误! 未定义书签。
4.1 前端兼容性测试.....	错误! 未定义书签。
4.1.1 测试方式.....	错误! 未定义书签。
4.1.2 测试结果:	错误! 未定义书签。
4.2 前端可视化质量测试.....	错误! 未定义书签。
4.2.1 测试方式.....	错误! 未定义书签。
4.2.2 测试结果:	错误! 未定义书签。
4.3 前后端结合测试.....	错误! 未定义书签。
4.3.1 测试方式.....	错误! 未定义书签。
4.3.2 测试结果:	错误! 未定义书签。
4.4 环境配置.....	1
4.4.1 JDK 安装.....	1
4.4.2 MySQL 安装.....	1
4.4.3 Maven-3.5.4 安装.....	2
4.5 系统部署.....	3
4.5.1 前端部署.....	3
4.5.2 后端部署.....	3

1 环境配置

1.1 JDK 安装

1、把下载好的 Linux 版本的 jdk1.8 的压缩文件，用 xftp 工具上传到 Linux 服务器中的 usr/local 文件中

```
cd/usr/local
jdk-8u261-linux-x64.tar.gz
```

2、解压安装包

```
tar -zxvf jdk-8u261-linux-x64.tar.gz
```

3、建立软连接(可以省略)

```
ln -s /usr/local/jdk1.8.0_121/ /usr/local/jdk
```

4、修改环境变量

```
vim /etc/profile
export JAVA_HOME=/usr/local/jdk
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JRE_HOME/lib
export PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
（修改完成后按 ESO 键，然后输入:wq 保存退出）
通过命令 source /etc/profile 让 profile 文件立即生效
source /etc/profile
```

5、测试是否安装成功

```
java -version
```

1.2 MySQL 安装

1、

```
#cd /data/software
```

2、解压压缩包

```
#tar -xzf /data/software/mysql-5.7.17-linux-glibc2.5-x86_64.tar.gz
```

3、移动并修改文件名

```
#mv /data/software/mysql-5.7.17-linux-glibc2.5-x86_64 /usr/local/mysql
```

4 创建数据仓库目录

```
--/data/mysql 数据仓库目录
```

```
# mkdir /data/mysql
```

```
#ls /data/
```

5、新建 mysql 用户、组及目录

```
# useradd -r -s /sbin/nologin -g mysql mysql -d /usr/local/mysql
```

6、改变目录属有者

```
#cd /usr/local/mysql
```

```
#pwd
```

```
#chown -R mysql .
```

```
#chgrp -R mysql .
```

```
#chown -R mysql /data/mysql
```

7、配置参数

```
# bin/mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/data/mysql
```

此处需要注意记录生成的临时密码，如上文结尾处的：YLi>7ecpe;YP

```
#bin/mysql_ssl_rsa_setup --datadir=/data/mysql
```

8、修改系统配置文件

```
#cd /usr/local/mysql/support-files
```

```
# cp my-default.cnf /etc/my.cnf
```

```
# cp mysql.server /etc/init.d/mysql
```

```
# vim /etc/init.d/mysql
```

1.3 Maven-3.5.4 安装

1、上传或下载安装包

```
cd/usr/local
```

```
apache-maven-3.5.4-bin.tar.gz
```

2、解压安装包

```
tar -zxvf apache-maven-3.6.1-bin.tar.gz
```

3、建立软连接(可以省略)

```
ln -s /usr/local/apache-maven-3.6.1/ /usr/local/maven
```

4、修改环境变量

```
vim /etc/profile
```

```
export MAVEN_HOME=/usr/local/maven
```

```
export PATH=$PATH:$MAVEN_HOME/bin
```

(修改完成后按 ESO 键，然后输入:wq 保存退出)

```
source /etc/profile (通过命令 source /etc/profile 让 profile 文件立即生效)
```

5、测试是否安装成功

```
mvn -v
```

2 系统部署

2.1 前端部署

1. 执行 **npm run build** 命令对项目打包(在 windows 中可以直接在项目文件夹直接 cmd 进行打包；在 Linux 中进行打包，不过要先在 Linux 系统中安装 node.js
注意：打包前修改 vue.config.js 中的 publicPath，将其修改为 ./，同时修改路由为后端地址
2. 利用 xftp 工具把打包后的 dist 文件夹上传到 Linux 中的 www 文件下
3. 在服务器 Linux 上安装 web 服务器 nginx1.18
4. 进入到 Nginx 目录下的 sbin 目录，输入 “./nginx” 命令启动 Nginx
5. 打包到 admin 文件下，所以配置 nginx 的 http 部分如下即可访问

```
server {
    listen      80;
    server_name 175.24.147.35;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    /root/MyProjects/dmas-web/admin;
        try_files $uri $uri/ /index.html;
        index   index.html index.htm;
    }
}
```

2.2 后端部署

1. 安装 MySQL 5.7 数据库
2. 安装 JDK 1.8.0_261
3. 安装 Maven-3.5.4
4. 使用 “mvn package” 命令进行项目打包
5. 编写脚本

1. 启动脚本 (start.sh)

```
nohup java -jar *****.jar --spring.profiles.active=prod > nohup.out 2>&1 &
```

2. 停止脚本 (stop.h)

```
PID=$(ps -ef | grep eladmin-system-2.6.jar | grep -v grep | awk '{ print $2 }')
if [ -z "$PID" ]
then
echo Application is already stopped
else
```

```
echo kill -9 $PID
```

```
kill -9 $PID
```

```
fi
```

3. 查看日志脚本 (log.sh)

```
tail -f nohup.out
```

4. 配置 nginx

```
server {  
    listen 80;  
    server_name 域名/当前服务器外网 IP;  
    location / {  
        proxy_pass http://127.0.0.1:8000; #这里的端口记得改成项目对应的哦  
        proxy_set_header X-Forwarded-Proto $scheme;  
        proxy_set_header X-Forwarded-Port $server_port;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection "upgrade";  
    }  
}
```

2.3 初始账号

默认账户: admin

密码: 111111