

教室预订系统 安装部署说明

目录

| | |
|---------------------------|---|
| 修改记录..... | 3 |
| 1. 引言..... | 4 |
| 1.1 编写目的..... | 4 |
| 1.2 系统背景及介绍..... | 4 |
| 1.3 适应人群..... | 4 |
| 2. 环境部署..... | 4 |
| 2.1 操作系统..... | 4 |
| 2.2 数据库配置..... | 4 |
| 3. 软件环境部署..... | 5 |
| 3.1 软件清单..... | 5 |
| 3.2 软件配置步骤..... | 5 |
| 3.2.1 配置 mysql 数据库..... | 5 |
| 3.2.2 修改部分文件..... | 6 |
| 3.2.3 构建并运行项目 Docker..... | 6 |
| (2) 构建镜像..... | 6 |
| (3) 启动容器..... | 6 |

修改记录

| 修订人 | 修订日期 | 简要说明 |
|-----|-----------|----------------|
| 敖津聪 | 2018.6.22 | 第一次完成文档 |
| 敖津聪 | 2018.6.26 | 修改文档格式，课室统一为教室 |
| | | |
| | | |

1. 引言

1.1 编写目的

教室预定系统安装部署说明书用于详细描述本系统的软硬件组成、架构，以及各组成部分的安装部署方法、配置方法等信息，通过本文档可以对整体系统进行部署。

1.2 系统背景及介绍

在大学，班级、社团以及一些校外活动往往需要学校的教室作为场地，但是目前的情况是，在申请教室时有许多困难，导致这些困难的原因有教室申请流程复杂、教室资源分配不合理、教室使用不透明等。

本系统旨在能够让师生通过网站来对教室进行申请和审批，以此提高申请教室的效率，给广大师生带来便利。

本系统将系统角色划分为学生、管理员和审批部门。这三类用户均可以登录该教室预订系统，并且登录信息将保存在中山大学信息数据库中。

学生可以查询教室信息，可查询内容包括教室空闲时间、教室容量等。学生用户选择相应的教室之后可以提交预订，也可以保存信息下次继续编辑，若提交预订，系统会显示预订该预订信息当前状态（待审核、通过、未通过）。

管理员可以进行维护教室信息、学生信息和审批部门信息等操作；审批部门可以审批教室预定申请，对于符合条件的教室预订申请确认通过，若不符合条件则驳回该申请（未通过）并附上原因。

同时本系统提供的安全机制防止他人冒充进行相应的操作，另外只有审批部门才能进行教室预订审批任务。

1.3 适应人群

本说明书使用于以下人群使用：

系统建设负责人：组织新建系统/功能的安装部署，以及新建系统/功能的安装部署说明书完善。

系统维护负责人：了解系统架构和安装部署方法，负责或组织进行系统重新安装部署，在系统部署变更时及时更新说明书内容。

2. 环境部署

2.1 操作系统

系统部署仅限于 Linux。

2.2 数据库配置

后端采用了 `goxorm` 数据库库框架，自动完成了代码数据表到数据库数据表的映射。需要自行配置 `mysql`。

3. 软件环境部署

3.1 软件清单

需安装工具如下：

docker

docker-compose

3.2 软件配置步骤

在进行测试前，需要进行简单的配置，步骤及具体操作步骤如下：

3.2.1 配置 mysql 数据库

(1) 构建镜像

由于使用 docker 部署测试，您需要先构建合适的 mysql 镜像。具体的配置和 Dockerfile 文档已经编写完成，只需要在项目根目录执行以下命令

```
$ cd testconfig/mysql/dbmaster
$ docker run build -t registry.cn-shenzhen.aliyuncs.com/selfmysql/master:latest .
$ cd ../dbslave
$ docker run build -t registry.cn-shenzhen.aliyuncs.com/selfmysql/slave .
```

图 3-1

运行命令后，可以得到以下两个镜像

```
$ docker images
registry.cn-shenzhen.aliyuncs.com/selfmysql/slave    latest    ad0a4971c1a5
registry.cn-shenzhen.aliyuncs.com/selfmysql/master   latest    cfa626cc975d
```

图 3-2

(2) 修改 docker-compose 文件配置

对 testconfig/mysql 目录下 docker-compose.yml 文件进行修改。主要修改 volumes 配置项，根据自己的情况挂载数据卷。例如，文件默认的配置为：

将本地目录 `$HOME/Work/data/dbslave` 挂载到镜像的 `/var/lib/mysql` 目录下。
将本地目录 `$HOME/Work/data/dbmaster` 挂载到镜像的 `/var/lib/mysql` 目录下。

图 3-3

(3) 启动

在 testconfig/mysql 目录下运行如下命令

```
$ docker-compose up -d
```

在 testconfig/mysql 目录下使用如下命令查看是否启动

```
$ docker-compose logs
```

(4) 主从配置

参考网址：<https://blog.csdn.net/kiloveyousmile/article/details/79833043>

(5) 备注

手动构建镜像步骤如下:

直接在 `testconfig/mysql` 目录下执行以下命令

```
$ sudo chmod +x update.sh
```

```
$ ./update.sh
```

3.2.2 修改部分文件

(1) 分别按照以下方式修改指定文件:

`Models/entities/init.go`

```
// Just for test
MasterEngine, err = xorm.NewEngine("mysql", "root:master@tcp(localhost:3307)/ccrsystem?charset=utf8&parseTime=true")
// True connection for mysql
// MasterEngine, err = xorm.NewEngine("mysql", "root:master@tcp(dbmaster:3306)/ccrsystem?charset=utf8&parseTime=true")

// Just for test
SlaveEngine, err = xorm.NewEngine("mysql", "root:slave@tcp(localhost:3308)/ccrsystem?charset=utf8&parseTime=true")
// True connection for mysql
// SlaveEngine, err = xorm.NewEngine("mysql", "root:slave@tcp(dbslave:3306)/ccrsystem?charset=utf8&parseTime=true")
```

图 3-4

(2) 在指定目录启动测试

进入根目录, 执行以下命令:

```
$ go test .
```

3.2.3 构建并运行项目 Docker

(1) 创建数据文件夹

以项目人员测试为例, 如下:

```
# 系统日志文件
$HOME/data/ccrsystem-data/ccrsystemlog
# sql数据库日志文件
$HOME/data/ccrsystem-data/sql
# session文件
$HOME/data/ccrsystem-data/sessions
```

图 3-5

(2) 构建镜像

在项目根目录运行

```
$ docker build -t image/ccrsystem:latest .
```

(3) 启动容器

在 `/testconfig/ccrsystem/` 目录下运行

```
$ docker-compose up -d
```

4. 常见问题

尚无。