

QAQ_OJ 软件用户手册

学号	姓名	分工
15030199012	朱仁博	软件配置、文档整理
15030199014	方小龙	使用指南、文档编写
15030199019	菅宁	软件组织、文档编写
15030199025	张炳婷	使用指南、文档编写

1. 引言

1.1. 标识

完整标识: QAQ_OJ (QAQ_Online Judge System) 1.0

版本号: 1.0

发行号: 1.0

1.2. 系统概述

本 OJ 可以为算法爱好者、普通学生和各高校的老师等群体提供方便的服务。最主要的功能为题目练习与比赛承办。

管理员可以开设一场比赛,普通用户可以选择参加比赛,在规定的时间内,尽可能完成相应的题目。管理员可以在比赛过程中和比赛结束后获得用户的答题统计情况和一系列排名数据。

项目开发过程主要靠四位成员协力合作,根据各自的擅长的方向完成各自的分工,项目前期包括整体逻辑结构的分析、主要功能的而实现(比如账号登录,上传题目与判题等功能)。中期完成后端数据库建设与维护、前端页面的设计与修改、安全性的考虑。后期则完成诸多细节的优化与用户体验的完善。

1.3. 文档概述

本文档为 QAQ_OJ 的用户手册,其中包括软件综述、访问软件、使用软件指南。

由于这只是一个课程项目,不涉及相关公司和相关部门的权利问题,因此没有特定的保密性和私密性要求。

2. 引用文件

《软件工程导论(第五版)》 张海藩编著 清华大学出版社出版

3. 软件综述

3.1. 软件应用

QAQ_OJ 可以为算法爱好者、普通学生和各高校的老师等群体提供方便的服务,旨在提升大家的编程水平。

3.2. 软件清单

```
mysql==5.7
gcc==5.4.0
g++==5.4.0
virtualenv==15.0.1
```

Python3.5 中的库:

```
amqp==2.3.2
asn1crypto==0.24.0
billiard==3.5.0.3
celery==4.2.0
```

cffi==1.11.5
cryptography==2.2.2
Django==2.0.6
idna==2.7
kombu==4.2.1
pyparser==2.18
PyMySQL==0.9.0
pytz==2018.4
redis==2.10.6
six==1.11.0
vine==1.1.4
Lorun==1.0

3.3. 软件环境

计算机设备：服务器，建议磁盘内存大于 50G

通信设备：可以连接到网络的设备

操作系统：Ubuntu 16.04 LTS 64 位

数据库：mysql 5.7

3.4. 软件组织和操作概述

① 超级管理员

- 1) 发布/管理主页的公告；
- 2) 管理所有用户的帐号密码等信息；
- 3) 管理所有比赛的后台数据；
- 4) 管理讨论板块的数据；
- 5) 管理员权限的所有操作；

② 管理员

- 1) 新建问题，管理问题；
- 2) 新建比赛，管理自己创建的比赛；
- 3) 查看练习模块中的所有提交记录；
- 4) 查看联系模块中的所有 Hack 记录；
- 5) 普通用户的所有操作；

③ 普通用户

- 1) 管理个人帐号信息；
- 2) 查看公告信息；
- 3) 在练习模块中提交代码，查看返回结果；
- 4) 查看自己的提交记录；
- 5) 使用 Hack 功能，查看自己的 Hack 记录；
- 6) 在讨论模块中参与讨论；
- 7) 查看 Rank 榜单；
- 8) 参加比赛；

3.5. 意外事故以及运行的备用状态和方式

服务器崩溃：重启服务器

网络错误：检查网络问题

数据库错误：恢复数据库

3.6. 保密性和私密性

开源软件，无特殊保密性和私密性要求。

3.7. 帮助和问题报告

无。

4. 访问软件

4.1. 软件的首次用户

4.1.1. 熟悉设备

请熟悉服务器、路由器、交换机等硬件设备。

4.1.2. 访问控制

需要 mysql 的 root 用户的密码。

4.1.3. 安装和设置

① 安装 Ubuntu 16.04 LTS 64 位操作系统

② 安装 mysql

```
# sudo apt-get install mysql-server mysql-client
```

③ 解决 mysql 中文乱码问题

打开 mysql 配置文件

```
# sudo gedit /etc/mysql/my.cnf
```

在文件末尾加入以下语句

```
[client]
```

```
default-character-set=utf8
```

```
[mysqld]
```

```
character-set-server=utf8
```

```
collation-server=utf8_general_ci
```

```
init-connect='SET NAMES utf8'
```

重启 mysql 服务器

```
# service mysql restart
```

重新打开命令行终端

④ 安装 virtualenv

```
# sudo apt-get install virtualenv
```

⑤ 创建 Python 虚拟环境，并进入

```
cd qaqoj/
```

```
virtualenv . -p python3
```

```
source bin/activate
```

⑥ 安装配置的库

```
# pip install -r requirements.txt
# sudo apt-get install redis-server
# cd Lo-runner/
# python setup.py install
# cd ..
```

⑦ 配置 Django 数据库选项

打开配置文件

```
# gedit qaqoj/settings.py
```

找到以下语句

```
DATABASES = {
    'default': {
        #'ENGINE': 'django.db.backends.sqlite3',
        #'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
        'ENGINE': 'django.db.backends.mysql',
        'HOST': '127.0.0.1',
        'PORT': '3306',
        'NAME': 'mysql',
        'USER': 'root',
        'PASSWORD': '*****',
    }
}
```

用 mysql 的 root 账户的密码替换*****

⑧ 迁移数据库表单

```
# python manage.py migrate
```

⑨ 导入已有的公告和题库

```
# python manage.py loaddata announcement.json
# python manage.py loaddata problem.json
```

⑩ 创建 QAQ_OJ 超级管理员

```
# python manage.py createsuperuser
```

4.2. 启动过程

① 开启 Django 服务器

```
# python manage.py runserver 0.0.0.0:端口号
```

② 开启评测服务

新开一个命令行终端，进入虚拟环境，开启评测服务

```
# cd qaqoj/
# source bin/activate
# celery -A qaqoj worker --concurrency=1 -l info
```

4.3. 停止和挂起工作

关闭服务器:在两个终端 CTRL+C

5. 使用软件指南

5.1. 能力

QAQ_OJ 包含了用户模块、公告模块、讨论模块、问题模块、提交模块、Hack 模块及比赛模块。

5.2. 约定

无。

5.3. 处理过程

5.3.1. 用户

用户必须先注册登录，才可以提交问题代码、提交 Hack、参加比赛。

5.3.2. 比赛

必须先报名才可以参加比赛，必须在比赛进行的时间才可以提交比赛代码。

5.3.3. Hack

被 Hack 的提交必须为结果正确，且 Hacker 必须已经通过了这道题目。

5.4. 相关处理

用户只与网站前端交互，所有相关的处理在 Django 后台解决，在 mysql 上的操作也是由 Django 负责。

5.5. 数据备份

① 数据库的导出

```
python manage.py dumpdata > database.json
```

② 数据库的恢复

```
python manage.py loaddata database.json
```

5.6. 错误，故障和紧急情况时的恢复

服务器崩溃：重启服务器

网络错误：检查网络问题

数据库错误：恢复数据库

5.7. 消息

无。

5.8. 快速引用指南

请百度 `ACM OJ 心得`

6. 注解

① Django

Django 在新一代的 Web 框架中非常出色，为什么这么说呢？

为回答该问题，让我们考虑一下不使用框架设计 Python 网页应用程序的情形。贯穿整本书，我们多次展示不使用框架实现网站基本功能的方法，让读者认识到框架开发的方便。（不使用框架，更多情况是没有合适的框架可用。最重要的是，理解实现的来龙去脉会使你成为一个优秀的 web 开发者。）

使用 Python 开发 Web，最简单，原始和直接的办法是使用 CGI 标准，在 1998 年这种方式很流行。现在从应用角度解释它是如何工作：首先做一个 Python 脚本，输出 HTML 代码，然后保存成 .cgi 扩展名的文件，通过浏览器访问此文件。

② bootstrap

Bootstrap 是最受欢迎的 HTML、CSS 和 JS 框架，用于开发响应式布局、移动设备优先的 WEB 项目。简洁、直观、强悍的前端开发框架，让 web 开发更迅速、简单。

为所有开发者、所有应用场景而设计。

Bootstrap 让前端开发更快速、简单。所有开发者都能快速上手、所有设备都可以适配、所有项目都适用。



预处理脚本

虽然可以直接使用 Bootstrap 提供的 CSS 样式表，不要忘记 Bootstrap 的源码是基于最流行的 CSS 预处理脚本 - Less 和 Sass 开发的。你可以采用预编译的 CSS 文件快速开发，也可以从源码定制自己需要的样式。



一个框架、多种设备

你的网站和应用能在 Bootstrap 的帮助下通过同一份代码快速、有效适配手机、平板、PC 设备，这一切都是 CSS 媒体查询 (Media Query) 的功劳。



特性齐全

Bootstrap 提供了全面、美观的文档。你能在这里找到关于 HTML 元素、HTML 和 CSS 组件、jQuery 插件方面的所有详细文档。

③ Mysql

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 [Oracle](#) 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件。

MySQL 是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

④ Redis

Redis 是完全开源免费的，遵守 BSD 协议，是一个高性能的 key-value 数据库。Redis 与其他 key - value 缓存产品有以下三个特点：

1) Redis 支持数据的持久化，可以将内存中的数据存储存在磁盘中，重启的时候可以再次加载进行使用。

2) Redis 不仅仅支持简单的 key-value 类型的数据，同时还提供 list，set，zset，hash 等数据结构的存储。

3) Redis 支持数据的备份，即 master-slave 模式的数据备份。

Redis 有着更为复杂的数据结构并且提供对他们的原子性操作，这是一个不同于其他数据库的进化路径。Redis 的数据类型都是基于基本数据结构的同时对程序员透明，无需进行额外的抽象。

Redis 运行在内存中但是可以持久化到磁盘，所以在对不同数据集进行高速读写时需要权衡内存，因为数据量不能大于硬件内存。在内存数据库方面的另一个优点是，相比在磁盘上相同的复杂的数据结构，在内存中操作起来非常简单，这样 Redis 可以做很多内部复杂性很强的事情。同时，在磁盘格式方面他们是紧凑的以追加的方式产生的，因为他们并不需要进行随机访问。

⑤ Celery

Celery 是一个基于 python 开发的分布式异步消息任务队列，通过它可以轻松的实现任务的异步处理，如果你的业务场景中需要用到异步任务，就可以考虑使用 celery，举几个实例场景中可用的例子：

你想对 100 台机器执行一条批量命令，可能会花很长时间，但你不想让你的程序等着结果返回，而是给你返回一个任务 ID，你过一段时间只需要拿着这个任务 id 就可以拿到任务执行结果，在任务执行 ing 进行时，你可以继续做其它的事情。

你想做一个定时任务，比如每天检测一下你们所有客户的资料，如果发现今天是客户的生日，就给他发个短信祝福

Celery 在执行任务时需要通过一个消息中间件来接收和发送任务消息，以及存储任务结果。Celery 有以下优点：

- 1) 简单：一旦熟悉了 celery 的工作流程后，配置和使用还是比较简单的
- 2) 高可用：当任务执行失败或执行过程中发生连接中断，celery 会自动尝试重新执行任务
- 3) 快速：一个单进程的 celery 每分钟可处理上百万个任务
- 4) 灵活：几乎 celery 的各个组件都可以被扩展及自定义

⑥ Docker

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 [Linux](#) 机器上，也可以实现虚拟化，容器是完全使用沙箱机制，相互之间不会有任何接口。

一个完整的 Docker 有以下几个部分组成：

1. dockerClient 客户端
2. Docker Daemon 守护进程
3. Docker Image 镜像
4. DockerContainer 容器

⑦ Lorun

lorun 是 github 上的一个开源项目，项目地址：<https://github.com/lodevil/Lo-runner>，原文是这么说的：We use this python-c library to run program in a sandbox-like environment. With it, we can accurately known the resource using of the program and limit its resource using including system-call interrupt.

这是用 C 语言写的一个 python 扩展模块，让程序在一个类似沙盒的环境下执行，然后精准的获取程序的执行时间和内存，还能对程序进行限制，限制程序的系统调用。