清华自习助手

设计文档

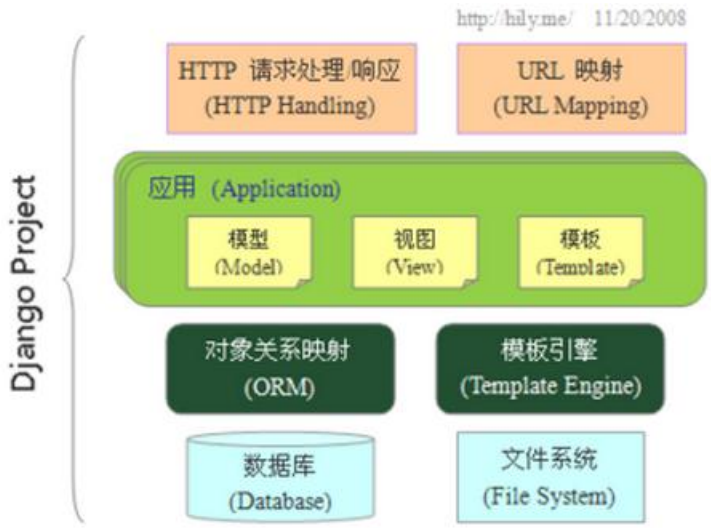
机智的程序猿

目录

一、 总体结构

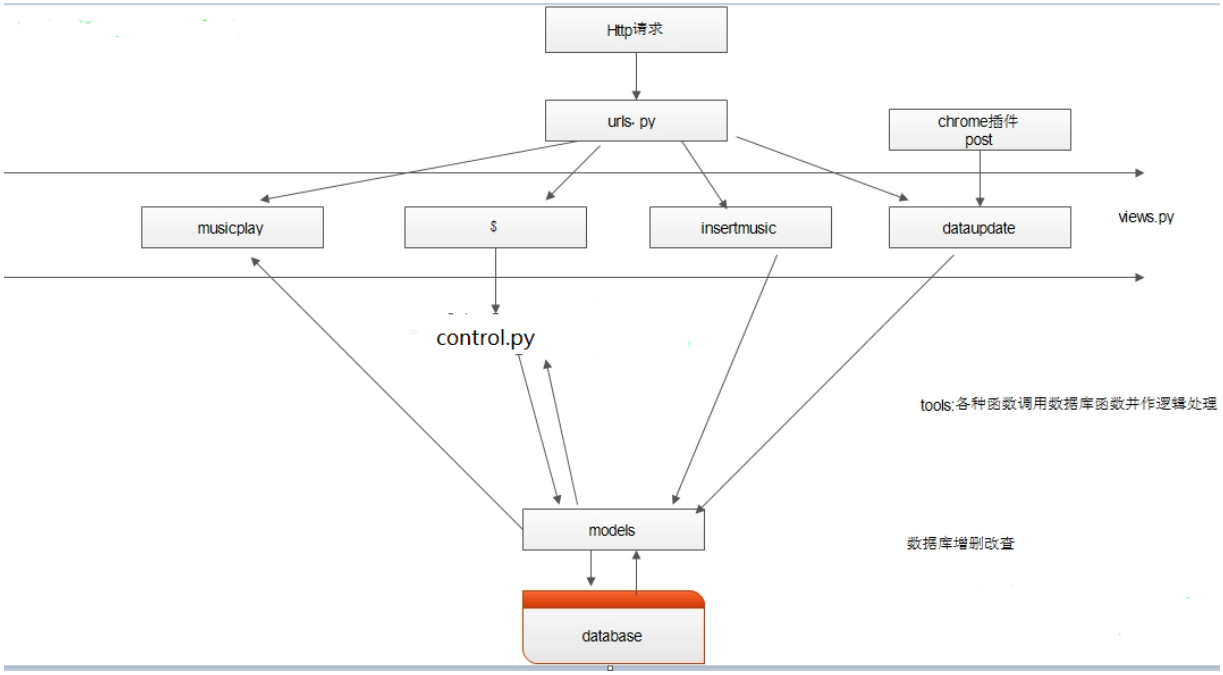
“清华自习助手”是一个搭建 BAE 上的微信公众平台后台。

系统使用 Django 框架。Django 框架的架构特点，使得项目内部的耦合度较小，小组成员可以独立开发，互不影响。



Django 项目结构如上图所示，URL 模块负责路由映射，将请求分配到 view 中对应的函数。View 中的函数对请求进行相应的处理。Model 模块负责与数据库交互。

“清华自习助手”的整体结构如下图所示。其中 control.py 负责解析消息类型和内容，并调用相应的函数（位于其他文件中，图中未表示）。



1. 运行机制

用户使用微信客户端向微信服务器发消息，微信服务器收到消息后发消息给 BAE 服务器。搭建在 BAE 上的后台处理消息后，返回给微信服务器。微信服务器再向用户回复消息。

2. 文件结构

架构相关文件：

View.py，utils.py，wsgi.py，modols.py，settings.py

功能相关文件：

Classroom.py，Control.py 等

3. 扩展方式

要添加新的消息处理，需要在 control.py 中添加判断，并调用相应的函数。处理一类消息的函数位于同一个文件中。

二、 功能与交互

1. 功能设计

清华自习助手的目标受众是上自习的清华学生。特点是全面周到。包括功能性的全面方便的各类教室、图书馆座位查询和使用帮助。娱乐性的签到、美食推荐、音乐推荐及播放。定位明确，针对性强，方便使用。

2. 交互设计

“清华自习助手”注重于用户的交互，美观精致。

“清华自习助手”使用了大量的图文消息和 html 页面，每一条消息都充分考虑了用户体验，精心制作。其中包括关注消息、帮助消息、文图座位查询结果、签到、音乐推荐列表和美食推荐。另外，有四个 HTML 页面，包括文图座位详细信息，帮助页面，美观简洁的播放器页面和关于我们。

3. 具体技术设计

(1) 菜单实现

Python 语言向微信服务器 post 数据，数据中是一个数组。

(2) 消息处理

文字消息和菜单事件由 view 调用 control.py 中的 processMessage 函数进行处理。处理过程包括调用其他逻辑函数，处理逻辑，并对数据库进行访问。之后通过调用 message.py 中的 5 个函数分别构造文字消息，图片消息，音频消息，视频消息和图文消息并返回。

三、 数据库设计

1. 表项设计

根据本次项目的需要，共设计了 3 张数据表。这 3 张表之间没有关联。

ThuHelper_user 表记录的是用户的基本信息，ThuHelper_onlinemusic 表记录的是音乐的基本信息，ThuHelper_classroom 表录的是教室的基本信息。具体设计如下：

ThuHelper_user

字段	数据类型	含义
id	int	用户编号，主键
openid	varchar	微信生成的用户 id
latestsignuptime	double	上一次签到时间的时间戳
signupstatus	varchar	最近一个月的签到情况
sumtime	int	总签到次数

ThuHelper_onlinemusic

字段	数据类型	含义
id	int	音乐编号，主键
title	varchar	歌名
singer	varchar	歌手
description	varchar	描述
type1	varchar	类型一
type2	varchar	类型二

type3	varchar	类型三
imageURL	varchar	图片 url

ThuHelper_classroom

字段	数据类型	含义
id	int	教室编号，主　键
building	varchar	教学楼
floor	int	楼层
roomnumber	varchar	教室名称
Monday	varchar	星期一教室使用情况
Tuesday	varchar	星期二教室使用情况
Wednesday	varchar	星期三教室使用情况
Thursday	varchar	星期四教室使用情况
Friday	varchar	星期五教室使用情况
Saturday	varchar	星期六教室使用情况
Sunday	varchar	星期日教室使用情况

2. 数据库的生成

本次项目采用 Django 框架，与 mysql 能够很好地配合。数据库的设计就是通过在 Django 框架下执行 python 代码实现的。如下面代码所示：

```
class Classroom(models.Model):
    building = models.CharField(max_length=10)
    floor = models.IntegerField()
    roomnumber = models.CharField(max_length=50)
    Monday = models.CharField(max_length=10)
    Tuesday = models.CharField(max_length=10)
    Wednesday = models.CharField(max_length=10)
    Thursday = models.CharField(max_length=10)
    Friday = models.CharField(max_length=10)
    Saturday = models.CharField(max_length=10)
    Sunday = models.CharField(max_length=10)

    def __unicode__(self):
        return self.roomnumber

class Onlinemusic(models.Model):
    imageURL = models.CharField(max_length=250)
    title_= models.CharField(max_length=50)
    singer = models.CharField(max_length=50)
    description = models.CharField(max_length=250)
    type1 = models.CharField(max_length=10)
    type2 = models.CharField(max_length=10)
    type3 = models.CharField(max_length=10)

    def __unicode__(self):
        return self.title
class User(models.Model):
    openid = models.CharField(max_length=250)
    latestsignuptime = models.FloatField()
    signupstatus = models.CharField(max_length=50)
    sumtime = models.IntegerField()

    def __unicode__(self):
        return self.openid
```

同时，我们借助 SQLyog 对数据库进行查看和一定的管理。

3. 数据库的操作

ThuHelper_user 相关函数

函数声明：def adduser(openid)

参数：openid，用户的微信 openid

返回值：无

功能：当用户关注微信服务号时，将该用户的信息存入数据库中

函数声明：def deluser(openid)

参数：openid，用户的微信 openid

返回值：无

功能：当用户取消关注微信服务号时，将该用户的信息从数据库中删除

函数声明：def getRecentInfobyID(ID)

参数：ID，用户的微信 openid

返回值：用户最近三十天签到情况的字符串，字符串长度为 30，每一位代表一天的签到情况，0 表示未签到，1 表示已签到

功能：根据用户的 openid 获取该用户最近三十天签到的情况

函数声明：def changeRecentInfo(ID, info)

参数：ID，用户的微信 openid；info，新的用户最近三十天签到情况字符串

返回值：若修改成功，则没有返回值；若修改失败，则返回字符串“Error”

功能：根据用户的 openid 修改该用户最近三十天的签到情况为 info

函数声明：def getLastTimebyID(ID)

参数：ID，用户的微信 openid

返回值：若获取成功，则返回用户最近一次签到时间的时间戳；若获取失败，则返回 0

功能：根据用户的 openid 获取该用户上一次签到时间的时间戳

函数声明：def changeLastTime(ID, now)

参数：ID，用户的微信 openid；now，当前时间

返回值：若修改成功，则没有返回值；若修改失败，则返回字符串“Error”

功能：根据用户的 openid 将该用户上一次签到时间的时间戳改为 now

函数声明：def addsignintime(ID)

参数：ID，用户的微信 openid

返回值：若修改成功，则没有返回值；若修改失败，则返回字符串“Error”

功能：根据用户的 openid 将该用户的总签到时间加一

函数声明：def getsignintimebyID(ID)

参数：ID，用户的微信 openid

返回值：若获取成功，则返回用户总的签到次数；若获取失败，则返回 0

功能：根据用户的 openid 获取该用户的总签到次数

函数声明：def getrankbyID(ID)

参数：ID，用户的微信 openid

返回值：类型为 dict，其中“rank”字段表示的是用户的签到次数排名，“total”字段表示的是总的用户数量

功能：根据用户的 openid 获取该用户的签到次数排名和总用户数量

ThuHelper_onlinemusic 相关函数

函数声明：def insertonlinemusic(music)

参数：music，dict 类型，包含一首音乐的所有信息

返回值：无

功能：将 music 插入到数据库中

函数声明：def getOneMusicByType(dict)

参数：dict，dict 类型，“type1”字段表示的是音乐类型一限制，“type2”字段表示的是音乐类型二限制，“type3”字段表示的是音乐类型三限制

返回值：若获取到符合条件的音乐，则返回一个 dict 类型的值，包括这首歌的所有信息；若没有获取到符合条件的音乐，则返回 None

功能：根据三个维度的类型限制返回符合条件的一首歌

ThuHelper_classroom 相关函数

函数声明：def getclassroomsbyfloor(building, floor, time, weekday)

参数：building，教学楼名称；floor，楼层；time，取值为 1 到 6，表示第几节课；weekday，取值为 1 到 7，表示星期几

返回值：一个数组，数组的每个变量为一个 dict，其中的“roomnumber”存储的是教室名称

功能：根据教学楼名称、楼层、第几节课、星期几四个条件查找数据库中在该条件下空闲的所有教室

函数声明：def getcoursebyroom(room)

参数：room，教室名称

返回值：一个 room 一天教室安排的字符串

功能：根据教室名称和当前时间返回该教室今天的课程安排

函数声明：def insertclassroom(building, roomnumber, status)

参数：building，教学楼名称；roomnumber，教室名称；status，教室一个星期课程安排的字符串

返回值：无

功能：将某个教室插入数据库中，用于第一版教室信息初始化程序中

函数声明：updateclassroombyweek(building, roomnumber, week, status)

参数：building，教学楼名称；roomnumber，教室名称；week，星期几；status，教室一天课程安排的字符串

返回值：无

功能：更新某个教室某一天的课程安排信息。如果该教室还不在数据库中，则在数据库中新建这个教室。用于第二版教室信息初始化程序中