

温州大学瓯江学院

WENZHOU UNIVERSITY OUJIANG COLLEGE

《Django 综合实验》

题 目： 博客网站的设计与实现

分 院： 数信分院

班 级： 16 计算机科学与技术二班

姓 名： 陈哲

学 号： 16219111221

完成日期： 2019 年 5 月 10 日

温州大学瓯江学院教务部

二〇一二年十一月制

目 录

目录

| | |
|---|----|
| 1. 可行性分析..... | 1 |
| 1.1. 市场可行性..... | 1 |
| 1.2. 操作可行性..... | 1 |
| 2. 项目需求分析..... | 1 |
| 2.1. 项目背景 | 1 |
| 2.2. 需求分析 | 1 |
| 1) 用户管理 | 1 |
| 2) 博客管理 | 2 |
| 3) 评论管理 | 2 |
| 4) RSS 订阅 | 2 |
| 3. 开发工具与编程语言 | 2 |
| 3.1. Django 技术简介 | 2 |
| 3.2. Python 技术简介 | 2 |
| 3.3. SQLite 技术简介 | 3 |
| 3.4. MYSQL 技术简介 | 3 |
| 4. 相关技术..... | 4 |
| 4.1. Django 的 MTV 模式..... | 4 |
| 1) Web 服务器（中间件）收到一个 http 请求..... | 4 |
| 2) Django 在 URLconf 里查找对应的视图(View)函数来处理 http 请求 | 5 |
| 3) 视图函数调用相应的数据模型来存取数据、调用相应的模板向用户展示页面 | 5 |
| 4) 视图函数处理结束后返回一个 http 的响应给 Web 服务器..... | 5 |
| 5) Web 服务器将响应发送给客户端..... | 5 |
| 4.2. Django 的 ORM..... | 5 |
| 1.什么是 ORM? | 5 |
| 2.ORM 的优缺点是什么? | 5 |
| 4.3. django-allauth | 5 |
| 1) 用户注册用户登录、..... | 5 |
| 2) 退出登录 | 5 |
| 3) 第三方 auth 登录(Github、微信、微博等等)..... | 6 |
| 4) 邮箱验证 | 6 |
| 5) 登录后密码重置 | 6 |
| 6) 忘记密码 | 6 |
| 7) 邮箱发送密码重置链接 | 6 |
| 4.4. django-ckeditor | 6 |
| 4.5. django-crispy-form | 7 |
| 4.6. django-mptt | 7 |
| 5. 设计方案概述..... | 8 |
| 6. 界面设计..... | 9 |
| 7. 类图设计..... | 13 |
| 8. 编码实现..... | 14 |
| 8.1. 关键代码说明..... | 14 |

| | |
|--------------------------------|----|
| 通过读取 post 中的文件字典，保存文件 | 16 |
| 通过 upload_to 属性指定头像的上传路径 | 16 |
| 9. 结论 | 22 |
| 9.1. 本文完成的工作 | 22 |
| 9.2. 网站优点与缺点 | 22 |
| 9.2.1. 优点: | 22 |
| 9.2.2. 缺点 | 22 |
| 9.3. 感想 | 22 |

1. 项目需求分析

1.1. 项目背景

国内外也有很多面向开发者的博客如：github.blog, CSDN 等等。但是国内对于一些框架和语言的快速入门的精品教程还比较少，所以希望可以建立一个教程博客来填补这一空缺。

1.2. 需求分析

博客是一个多用户、多界面的操作系统。有以下基本功能：

1) 用户管理

用户注册，登录，更新个人信息，邮箱验证，退出登录等功能。

2) 博客管理

搜索博客，添加博客，删除博客，修改博客，博客类别和标签的添加的功能。

3) 评论管理

博客的评论和回复的功能。

4) RSS 订阅

文章更新的同时并推送。

2. 开发工具与编程语言

2.1. Django 技术简介

Django 是一个开放源代码的 Web 应用框架，由 Python 写成。采用了 MVC 的框架模式，即模型 M，视图 V 和控制器 C。它最初是被开发来用于管理劳伦斯出版集团旗下的一些以新闻内容为主的网站的，即是 CMS（内容管理系统）软件。并于 2005 年 7 月在 BSD 许可证下发布。这套框架是以比利时的吉普

赛爵士吉他手 Django Reinhardt 来命名的。

Django 是一个基于 MVC 构造的框架。但是在 Django 中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板（Template）和视图（Views），称为 MTV 模式。

Django 适合快速开发 web 开发。

2.2. Python 技术简介

Python 是一种面向对象的解释型计算机程序设计语言，由荷兰人 Guido van Rossum 于 1989 年发明，第一个公开发行版发行于 1991 年。Python 是纯粹的自由软件，源代码和解释器 CPython 遵循 GPL (GNU General Public License) 许可。Python 语法简洁清晰，特色之一是强制用空白符（white space）作为语句缩进。Python 具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是 C/C++）很轻松地联结在一起。常见的一种应用情形是，使用 Python 快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如 3D 游戏中的图形渲染模块，性能要求特别高，就可以用 C/C++ 重写，而后封装为 Python 可以调用的扩展类库。需要注意的是在您使用扩展类库时可能需要考虑平台问题，某些可能不提供跨平台的实现。7 月 20 日，IEEE 发布 2017 年编程语言排行榜：Python 高居首位。

2.3. SQLite 技术简介

SQLite 是遵守 ACID 的关系数据库管理系统，它包含在一个相对小的 C 程式库中。它是 D.Richard Hipp 建立的公有领域项目。不像常见的客户端/服务器结构范例，SQLite 引擎不是个程序与之通信的独立进程，而是连接到程序中

成为它的一个主要部分。所以主要的通信协议是在编程语言内的直接 API 调用。这在消耗总量、延迟时间和整体简单性上有积极的作用。整个数据库（定义、表、索引和数据本身）都在宿主主机上存储在一个单一的文件中。它的简单的设计是通过在开始一个事务的时候锁定整个数据文件而完成的。

2.4. MYSQL 技术简介

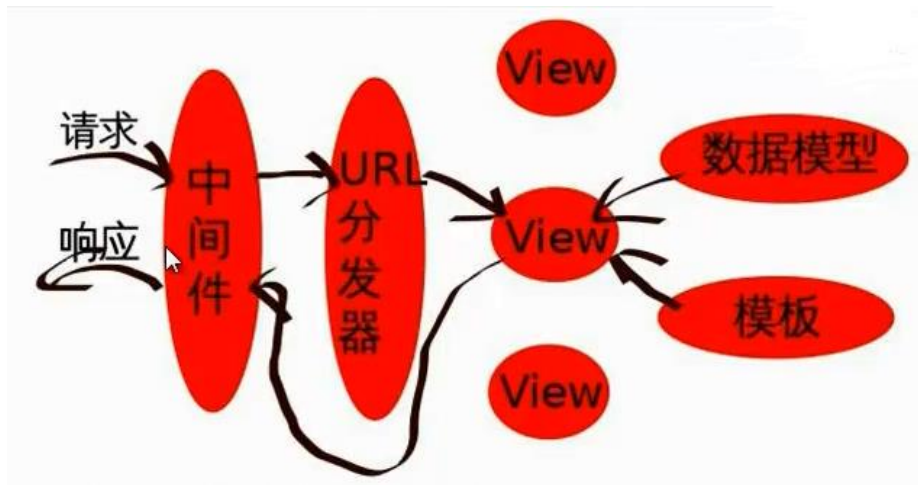
MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS (Relational Database Management System, 关系数据库管理系统) 应用软件之一。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，它分为社区版和商业版，由于其体积小、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。由于其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

3. 相关技术

3.1. Django 的 MTV 模式

- M 代表模型 (Model)：负责业务对象和数据库的关系映射(ORM)。
 - T 代表模板 (Template)：负责如何把页面展示给用户(html)。
 - V 代表视图(View)：负责业务逻辑，并在适当时候调用 Model 和 Template。
- 除了以上三层之外，还需要一个 URL 分发器，它的作用是将一个个 URL 的页面请求分发给不同的 View 处理，View 再调用相应的 Model 和 Template，

MTV 的响应模式如下所示：



- 1) Web 服务器（中间件）收到一个 http 请求
- 2) Django 在 URLconf 里查找对应的视图(View)函数来处理 http 请求
- 3) 视图函数调用相应的数据模型来存取数据、调用相应的模板向用户展示页面
- 4) 视图函数处理结束后返回一个 http 的响应给 Web 服务器
- 5) Web 服务器将响应发送给客户端

3.2. Django 的 ORM

1.什么是 ORM?

ORM，即 Object-Relational Mapping（对象关系映射），它的作用是在关系型数据库和业务实体对象之间作一个映射，这样，我们在具体的操作业务对象的时候，就不需要再去和复杂的 SQL 语句打交道，只需简单的操作对象的属性和方法。

2.ORM 的优缺点是什么？

优点：摆脱复杂的 SQL 操作，适应快速开发；

让数据结构变得简洁；

数据库迁移成本更低（如从 mysql->oracle）

缺点：性能较差、不适用于大型应用；复杂的 SQL 操作还需通过 SQL 语句实现

3.3. django-allauth

管理用户登录与注册的第三方 Django 安装包。它非常强大，能实现以下核心功能。

- 1) 用户注册用户登录、
- 2) 退出登录
- 3) 第三方 auth 登录(Github、微信、微博等等)
- 4) 邮箱验证
- 5) 登录后密码重置
- 6) 忘记密码
- 7) 邮箱发送密码重置链接

3.4. django-ckeditor

富文本编辑器，即使不会编写 html 代码的用户也可以设置各种文本格式

同时项目配置文件中可以定制各种需要的功能，例如文件上传功能等。同时支持安装第三方插件，如代码高亮插件(Prism)


```
# tab键转换空格数
'tabSpaces': 4,
# 工具栏风格
'toolbar': 'Custom',
# 工具栏按钮
'toolbar_Custom': [
    # 表情 代码块
    ['Smiley', 'CodeSnippet'],
    # 字体风格
    ['Bold', 'Italic', 'Underline', 'RemoveFormat', 'Blockquote'],
    # 字体颜色
    ['TextColor', 'BGColor'],
    # 链接
    ['Link', 'Unlink'],
    # 列表
    ['NumberedList', 'BulletedList'],
    # 最大化
    ['Maximize'],
    ['Image'],
],
# 插件code代码块
'extraPlugins': ','.join(['codesnippet','prism', 'widget',
'lineutils']),
```

CodeTest

CodeTest · 2019年5月15日 23:12 · admin · 评论 · 14 阅读

```
1 def profile_update(request):
2     user = request.user
3     user_profile = get_object_or_404(Profile, user=user)
4
5     if request.method == "POST":
6         form = ProfileForm(request.POST, request.FILES)
```

标签:

3.5. django-crispy-form

django-crispy-forms 为您提供了一个|crispy 过滤器和标签，可让您以非常优雅的方式控制 Django 表单的呈现行为。无需编写自定义表单模板即可完全控制。所有这些都未打破 Django 中标准的做事方式，所以它与任何其他表单应用程序都很好。{% crispy %}

3.6. django-mptt

Django mptt 是个 Django 第三方组件，目标是使 Django 项目能在数据库中存储层级数据（树形数据）。它负责管理数据库表作为树结构的细节，并提供了处理模型实例树的工具。

4. 界面设计



CodeTest

CodeTest · 2019年5月15日 23:12 · admin · 15 阅读

```
1 def profile_update(request):
2     user = request.user
3     user_profile = get_object_or_404(Profile, user=user)
4
5     if request.method == "POST":
6         form = ProfileForm(request.POST, request.FILES)
```

标签:

尚未登录，请[登录后](#)参与评论

😊

🔗

B

I

U

I_x

”

A

A

🔗

🔗

1=

2=

🔗

🔗

🔗

提交评论

最新文章

- > Selenium爬取京东手机商品列表
- > CodeTest
- > 豆瓣电影 Top 250
- >

归档


2019 年 5 月

分类

- ☑ JdMobile (1)
- ☑ doubanTop250 (1)
- ☑ CodeTest (1)

标签云

暂无标签!

 [RSS 订阅](#)

第三方登陆

- GitHub

还没有账号? [注册新账号](#)

账号*

密码*

☐ 记住我

[忘记密码了?](#)

[登陆](#)

最新文章

- > Selenium爬取京东手机商品列表
- > CodeTest
- > 豆瓣电影 Top 250
- >

归档


2019 年 5 月

分类

- 🕒 JdMobile (1)
- 🕒 doubanTop250 (1)
- 🕒 CodeTest (1)

标签云

暂无标签!

 [RSS 订阅](#)

[更新用户信息](#)[邮箱管理](#)[修改密码](#)[注销](#)

用户名: Albert

头像

[选择文件](#) 未选择任何文件

First Name

Albert

Last Name

Chen

个人简介

瓯江学院学生

Telephone

18808666688

[提交](#)

最新文章

[> Selenium爬取京东手机商品列表](#)[> CodeTest](#)[> 豆瓣电影 Top 250](#)[>](#)

归档

📅 2019 年 5 月

分类

[🕒 JdMobile \(1\)](#)[🕒 doubanTop250 \(1\)](#)[🕒 CodeTest \(1\)](#)

标签云

[Spider](#)[Code](#)[RSS 订阅](#)Copyright© 2019 Albert's Blog
Powered by Django & Bootstrap

About Me

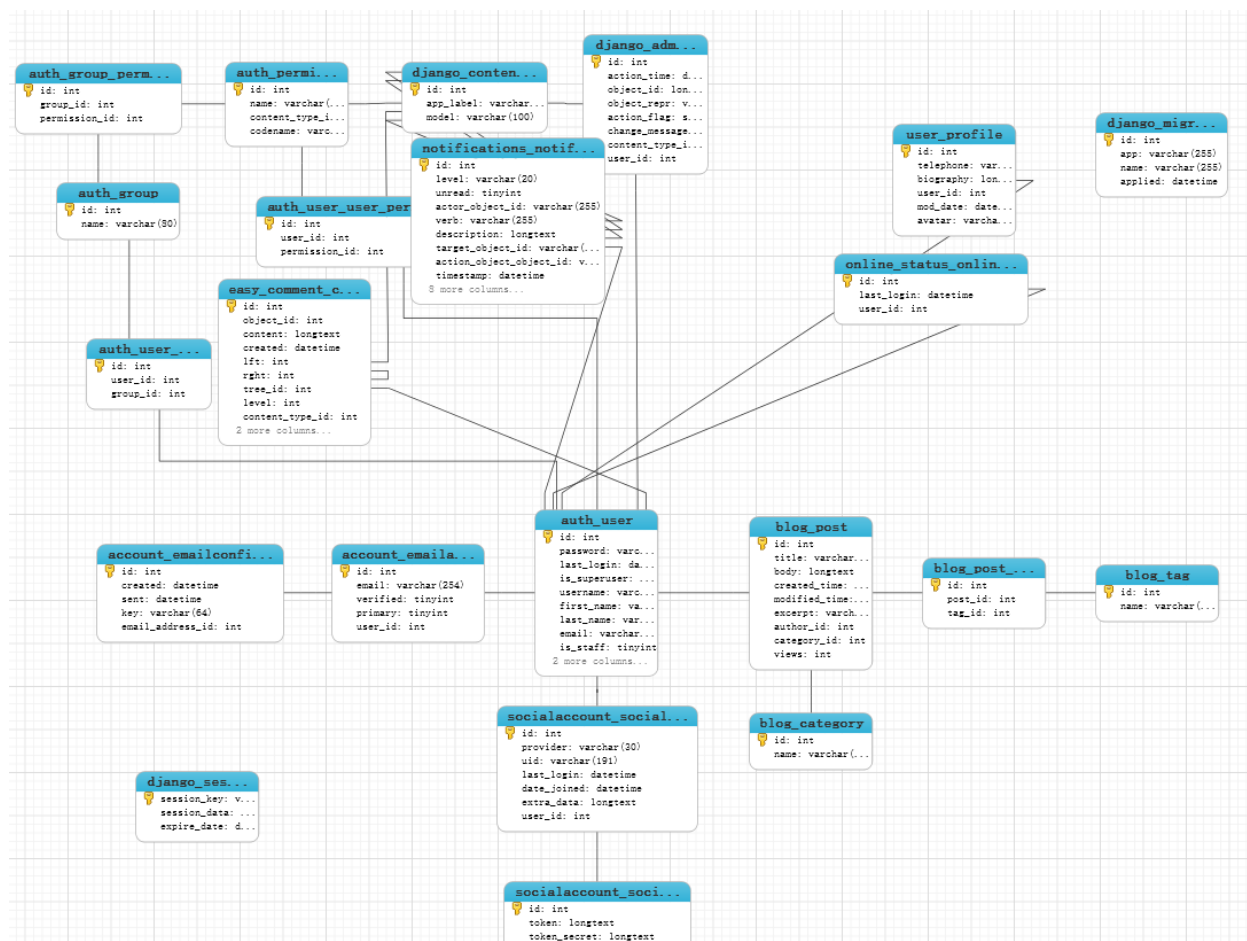
温州大学瓯江学院在校本科生

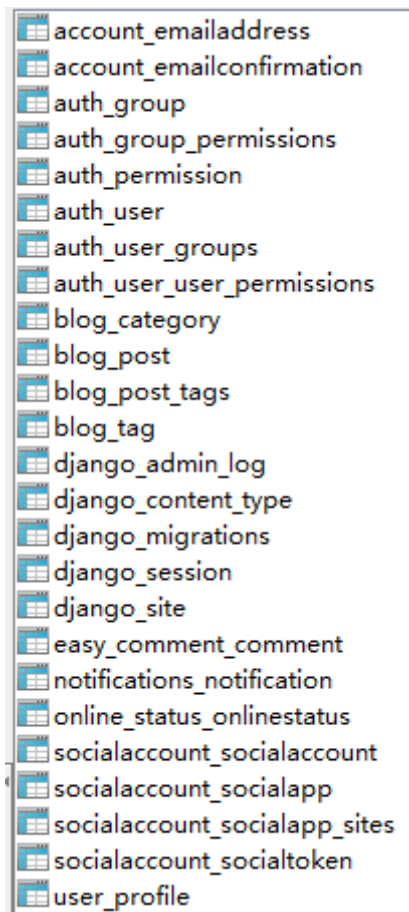
主要技能: Java,Scrapy,NET,Django,Python,Bootstrap,MongoDB...

Social

[f](#)[t](#)[g+](#)感谢你的小小停留
愿你会有小小收获Copyright© 2019 Albert's Blog
Powered by Django & Bootstrap

5. 类图设计



A screenshot of a Django project's database models list, showing a vertical scrollable list of model names. Each model name is preceded by a small icon representing a database table. The models listed are: account_emailaddress, account_emailconfirmation, auth_group, auth_group_permissions, auth_permission, auth_user, auth_user_groups, auth_user_user_permissions, blog_category, blog_post, blog_post_tags, blog_tag, django_admin_log, django_content_type, django_migrations, django_session, django_site, easy_comment_comment, notifications_notification, online_status_onlinestatus, socialaccount_socialaccount, socialaccount_socialapp, socialaccount_socialapp_sites, socialaccount_socialtoken, and user_profile.

| |
|-------------------------------|
| account_emailaddress |
| account_emailconfirmation |
| auth_group |
| auth_group_permissions |
| auth_permission |
| auth_user |
| auth_user_groups |
| auth_user_user_permissions |
| blog_category |
| blog_post |
| blog_post_tags |
| blog_tag |
| django_admin_log |
| django_content_type |
| django_migrations |
| django_session |
| django_site |
| easy_comment_comment |
| notifications_notification |
| online_status_onlinestatus |
| socialaccount_socialaccount |
| socialaccount_socialapp |
| socialaccount_socialapp_sites |
| socialaccount_socialtoken |
| user_profile |

6. 编码实现

6.1. 关键代码说明

6.1.1. 新建 Blog 应用

- 添加 Blog 基本模型，建立视图函数
- 将用户请求的 URL 链接与视图函数关联起来


```
urlpatterns = [
    #url(r'^$', views.index, name='index'),
    url(r'^post/(?P<pk>[0-9]+)/$', views.detail, name='detail'),
    #url(r'^archives/(?P<year>[0-9]{4})/(?P<month>[0-9]{1,2})/$',
    views.archives, name='archives'),
    #url(r'^category/(?P<pk>[0-9]+)/$', views.category, name='category'),
    url(r'^$', views.IndexView.as_view(), name='index'),
    url(r'^category/(?P<pk>[0-9]+)/$', views.CategoryView.as_view(),
    name='category'),
    url(r'^archives/(?P<year>[0-9]{4})/(?P<month>[0-9]{1,2})/$',
    views.ArchivesView.as_view(), name='archives'),
    url(r'^tag/(?P<pk>[0-9]+)/$', views.TagView.as_view(), name='tag'),
    url(r'^about.html/$', views.AboutView, name='about'),
    url(r'^full-width.html/$', views.BlogWideView.as_view(), name='wideblog'),
    url(r'^search/$', views.search, name='search'),
]
```

- 添加 Blog 标签供网页模板调用

```
#最新文章标签
@register.simple_tag
def get_recent_posts(num=5):
    return Post.objects.all().order_by('-created_time')[:num]

#归档模板标签
@register.simple_tag
def archives():
    return Post.objects.dates('created_time', 'month', order='DESC')

#分类模板标签
@register.simple_tag
def get_categories():
    # Count 计算分类下的文章数，其接受的参数为需要计数的模型的名称
    return Category.objects.annotate(num_posts=Count('post')).filter(
        (num_posts__gt=0))

#标签云
@register.simple_tag
def get_tags():
    return Tag.objects.annotate(num_posts=Count('post')).filter(
        (num_posts__gt=0))
```

6.1.2. 扩展用户信息，支持用户头像上传

因为 django-allauth 在用户注册只会创建 Django_User 对象只包含少量的属性，博客网站的用户信息并不复杂，因此扩展 User 就足够了。增加 user 应用和与之关联的 Profile 对象，我们希望用户在注册时两个对象一起被创建，并存储到数据库中。

```
# 用户扩展信息
class Profile(models.Model):
    # 与 User 模型构成一对一的关系
    user = models.OneToOneField(User, on_delete=models.CASCADE,
                                related_name='profile')
    # 电话号码字段
    telephone = models.CharField(max_length=20, blank=True)
    # 头像
    avatar = models.ImageField(upload_to='avatar/%Y%m%d/', blank=True)
    # 个人简介
    biography = models.TextField(max_length=500, blank=True)

    mod_date = models.DateTimeField('Last modified', auto_now=True)

    def __str__(self):
        return 'user {}'.format(self.user.username)
```

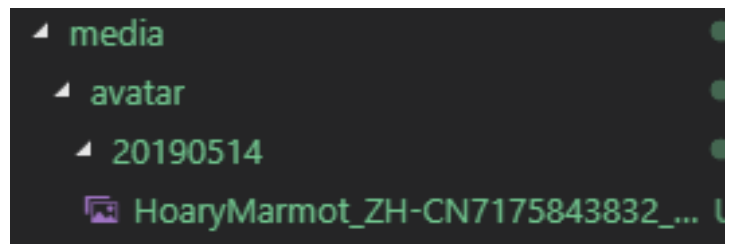
通过读取 post 中的文件字典，保存文件

```
# 上传的文件保存在 request.FILES 中，通过参数传递给表单类
profile_form = ProfileForm(request.POST, request.FILES)

if profile_form.is_valid():
    ...

    # 添加在 profile.bio = profile_cd['bio'] 后面
    # 如果 request.FILES 存在文件，则保存
    if 'avatar' in request.FILES:
        profile.avatar = profile_cd["avatar"]
```

通过 upload_to 属性指定头像的上传路径



6.1.3. 重写表单

django-allauth 提供的`{{ form.as_p }}`的表单非常简陋而使用 `crispy_forms` 后

Albert c Blog

E-mail地址

以下e-mail地址已关联到您的帐号:

● **czforwork@yeah.net** 已验证 首选e-mail

设置首选e-mail

重发验证e-mail

删除

添加E-mail地址

E-mail:

添加E-mail

只需在模板页中添加`{{ form|crispy }}`过滤器即可自动完成 bootstrap3 样式的

添加

以下e-mail地址已关联到您的帐号：

● **czforwork@yeah.net** 已验证 首选e-mail

设置首选e-mail

重发验证e-mail

删除

添加E-mail地址

E-mail*

E-mail地址

添加邮箱

6.1.4. 重写 comment 应用，实现多级评论

因为第一次添加的 comment 应用只支持顺序评论没有回复的功能，所以进行重写，并添加 mptt 来建立树状模型，继而支持可以回复别人的评论。

```
parent = TreeForeignKey('self', blank=True, null=True, verbose_name='父级  
评论', on_delete=models.CASCADE)
```

前端通过 Ajax post 数据，然后以 jsonresponse 的格式返回对应的 html 和参与评论数量。



6.1.5. 增加回到顶部浮动按钮

添加按钮样式和按钮标签，通过脚本进行控制按钮的显示以及功能的实现。



```
/* 按钮边框的大小、位置、样式 */
.toTop-arrow {
    width: 5rem;
    height: 5rem;
    padding: 0;
    margin: 0;
    border: 0;
    border-radius: 33%;
    opacity: 0.7;
    background-image: url('../img/arrow.png');
    background-repeat: no-repeat;
    background-position: center;
    cursor: pointer;
    position: fixed;
    right: 1.5rem;
    bottom: 1.5rem;
    display: none;
    z-index: 100;
}

.toTop-arrow2::before, .toTop-arrow2::after {
    width: 31px;
    height: 7px;
```

```

<script>
    // 向上滚动的函数
    $(function () {
        $('#BackTop').click(function () {
            $('html,body').animate({ scrollTop: 0 }, 500);
        });
        $(window).scroll(function () {
            if ($(this).scrollTop() > 300) {
                $('#BackTop').fadeIn(300);
            } else {
                $('#BackTop').stop().fadeOut(300);
            }
        }).scroll();
        $(".django-ckeditor-widget").removeAttr('style');
    });
</script>

```

6.1.6. 动态绑定 click 事件

我们可以使用 on()方法，通过将事件绑定到 document 来进行事件委派，从而为后面通过 ajax 加载的评论添加 click 事件。

```

$(document).on("click", '.reply', function(){
    if ($("#submit-button").hasClass('no-login')) {
        alert('请登录后再参与评论! ');
        return false;
    }
    // 评论框显示被回复的用户名，显示取消回复按钮
    $("#recipient").text($(this).data('name')).css('display', 'inline-block');
    $("#reply-action").css('display', 'inline-block');
    $("#cancel-button").css('display', 'inline-block');
    sessionStorage.setItem('reply', true);
    // 获取父级评论的id存在session-storage里，提交评论时提取出来
    sessionStorage.setItem('pid', $(this).attr('id'));
});

$('.pinglun-text img').removeAttr('style');
$(".django-ckeditor-widget").removeAttr('style');

```

7. 结论

7.1. 本文完成的工作

通过 Django Web 开发框架用最快的速度、更少的代码搭建了一个满足核心需求的博客网站。

7.2. 网站优点与缺点

7.2.1. 优点：

- 能够利用第三方开源库快速添加常用的功能
- 具备博客核心功能，交互友好
- 能够个性化展示爬虫数据

7.2.2. 缺点

- 网站尚未部署到服务器测试
- 缺少教程博客内容

7.3. 感想

通过此次实验，我对 MVC 的软件设计模式有了更加深入的认识。Django 是 MTV 模式框架（MODLE 数据， Template 样式， VIEW 处理），Django 将 Controller 弱化，让开发者专注业务逻辑的开发。用户提交 http 请求到 web 服务器接口经之变为 url 并且传给 web 框架最先传给路由模块使其与特定功能相组合。我们的功能放在了 views 模块中，由 view 选取特定的 templates 和特定的 models。每个框架都有各自的优缺点，但是总的设计模式都是大致相同的。

在遇到搜索引擎解决不了的问题时，不妨仔细阅读官方文档。