

# 期中考试：构建**web**应用程序

Kalu Kalu

期中考试：构建**Web**应用程序

期中说明与问题

客户规范说明

复习内容将与中期测试近乎一致

题目

扣分项

加分项

复习

创建我们的项目和应用程序

关于创建“零售商店”项目的说明

关于创建“淘宝”应用程序的说明

在**setings.py**中添加我们的“淘宝”应用程序

**Project urls.py 5**

**App urls.py 6**

**views.py 6**

**home\_page.html**

添加图片

形式

动作属性

方法属性

**CSRF**标注

名称属性

运行您的服务器

## 期中说明与问题

期中考试测试了学生制作一个网络应用程序的能力，并在 **Github** 上共享了它的开放性资源。

在期中考试中，学生将被要求启动终端，激活虚拟环境并完成以下任务：

1. 使用 **Django** 为我们的客户创建一个 **web-app** 网页应用程序概念的简单证明，这个证明是一个基于客户规范说明的零售商店。
2. 将整个 **Django** 项目上传到 **Github** 上

## 客户规范说明

1. 标题为“**Kool Karen Taobao**”
2. 网页标题应该有一个大标题（一级标题）的零售商店名称，**Kool Karen** 零售商店
3. 零售店名称下面应该是公司的口号，“您的方便是我们的责任”用较小的字体。（你可以使用你自己的标语，但它一定要很吸引人。）
4. 应该有至少 **3** 个有关商店项目的图片以及商店项目的名称
5. 该网站应该有一个工作机制，允许用户提交他们的订单和地址
6. 当用户提交他们的菜订单和地址后，他们应该能看到它打印在订单下面的网页上。
7. 附加项：在一个名为 **Order** 的表中保存数据库中的所有用户输入
8. 附加项：创建一个用户管理端，使我们可以在数据库中看到

数据。

复习内容将与期中测试近乎一致

这次复习与期中考试时会被问到的问题几乎一样。唯一的区别将在应用程序的名称，项目的名称和一些文本上。不仅仅要为“**Kool Karen**零售商店”制作一个便利的商店网站，另外用户还可以张贴他们想要的商品和他们的地址。可能会让你为“**EvelynEats**冰淇淋店”制作一个冰淇淋商店的网站。用户可以在那里发布他们想要的冰淇淋以及他们的地址。

除此之外，期中测试将是完全一样的。

题目

学生们必须在他们的终端中截图，并将其保存在他们的~/**Documents/your\_name** 文件夹中。考试结束后，我将马上看这些截图，并给他们的考试打分。

考试将按以下标准进行评分。

### 1. Django项目(70分)

- 1) 拥有一个可以使用**127.0.0.0 : 8000**网址访问的工作网站。  
( 10分 )
- 2) 包括标题(具体说明1) ( 10分 )
- 3) 包括适当的网页(具体说明2) ( 10分 )
- 4) 包括店名和标语(具体说明3) ( 10分 )
- 5) 包括三张商店物品的图片(具体说明4) ( 10分 )
- 6) 在主页中包括一个工作机制(具体说明5) ( 10分 )
- 7) 当用户提交时，他们可以看到他们的输出内容(具体说明6)  
( 10分 )

### 2. 上传到Github ( 30分 )

- 1) 创建本地存储库
- 2) 创建GitHub存储库

### 3) 将所有更改保存到本地存储库并推送到远程存储库

#### 扣分项

如果你需要我的帮助才能进行下一步，或者如果你和其他同学说话的话，我会从你的分数中扣除5分。（这是不允许的）

#### 加分项

1. 正确创建数据库和数据库表加10分
2. 创建一个工作的管理页面加10分

#### 复习

让我们复习期末考试。

#### 创建我们的项目和应用程序

我们需要做的第一件事是制作**Django**项目和应用程序。我们可以称**Django**项目为“零售商店”，成应用程序为“淘宝”。

首先，让我们打开我们的终端，**CD**进入您的文件夹，并创建我们的项目和我们的应用程序。

```
KaludeMacBook-Pro:~ kalukalu$ cd ~/Documents/kalu/
KaludeMacBook-Pro:kalu kalukalu$ source ~/
django_and_machine_learning_venv/bin/activate
(django_and_machine_learning_venv) KaludeMacBook-Pro:kalu
kalukalu$ django-admin startproject retail_store 1
(django_and_machine_learning_venv) KaludeMacBook-Pro:kalu
kalukalu$ cd retail_store/
(django_and_machine_learning_venv) KaludeMacBook-
Pro:retail_store kalukalu$ python manage.py startapp taobao
2
```

#### 关于创建“零售商店”项目的说明

这里我们做了两件不同寻常的事情。在此之前，我们的项目名为“**My\_Site**”。现在我们的项目名为“**retail\_store**”，因此我们需要执行**django-admin startproject retail\_store 1**。

关于创建“淘宝”应用程序的说明

接下来，我们的应用程序名称也与以前不同。因此，要创建一个名为“淘宝”的应用程序(而不是以前的博客)，我们需要做 **python manage.py. startapp taobao 2**。

在**setings.py**中添加我们的“淘宝”应用程序

在**sublime**中打开新创建的“**retail\_store**”文件夹，并打开 **settings.py** 文件。在**settings.py**中，应该将新创建的应用程序添加到**INCLUDED\_APPS**列表中。

*your\_name/retail\_store/retail\_store/settings.py*

```
1. INSTALLED_APPS = [  
2.     'django.contrib.admin',  
3.     'django.contrib.auth',  
4.     'django.contrib.contenttypes',  
5.     'django.contrib.sessions',  
6.     'django.contrib.messages',  
7.     'django.contrib.staticfiles',  
8.     'taobao', 1  
9. ]
```

You can see at 1 we added the name of our app as a string.

## Project urls.py

接下来，我们可以设置我们的 **project-wide urls.py**。在“**sublime**”中，打开位于**your\_name/retail\_store/retail\_store/urls.py**中的**urls.py**。然后编辑它，如下所示。  
*your\_name/retail\_store/retail\_store/urls.py*

```
1. from django.contrib import admin  
2. from django.urls import path, include 1  
3.  
4. urlpatterns = [  
5.     path("", include('taobao.urls')), 2  
6.     path('admin/', admin.site.urls),  
7. ]
```

我们只需将“**include**”添加到**Imports 1**中，添加我们的 **url path 2**，并指向我们将要创建的应用程序**URL**。

## App urls.py

*your\_name/retail\_store/taobao/urls.py*

```
1. from django.urls import path
2.
3. from . import views
4. app_name = 'taobao' 1
5. urlpatterns = [
6.     path("", views.home_page, name='home_page'),
7. ]
```

与我们的博客应用不同的是，我们将**app\_name**设置为“淘宝”**1**，而不是“博客”。除此之外，一切都是一样的。

## views.py

*your\_name/retail\_store/taobao/view.py*

```
1. from django.shortcuts import render
2.
3. def home_page(request):
4.
5.     if request.method == 'POST':
6.         item = request.POST.get('item') 1
7.         address = request.POST.get( "address") 2
8.
9.         return render(request, 'taobao/home_page.html', {'item': item,
10. 'address': address}) 3
11.
11.     return render(request, 'taobao/home_page.html' ) 4
```

在检查是否发出**POST**请求的**if**语句块中，我们首先获取用户输入的“**dish**”的值，并将其赋值给一个名为“**dish**”**1**的变量。接下来，我们获取用户名为“**Address**”结构输入的值，并将其分配给一个名为“**Address**”的变量**2**。“**1 item**项目”和“**address**地址”**2**个观点，我们都通过了其各自的要求。**POST.get()**方法调用，将在**home\_page.html**部分中再次讨论。我们在这里的论点，是“项目”和“地址”。

接着，我们将订单和地址都传递给上下文字典中的**home\_page.html**模板。

最后，如果这里不是**POST**请求，那么我们将仅仅返回 **home\_page.html**模板，而没有任何额外的信息用于我们的模板

## home\_page.html

接下来，我们需要在淘宝文件夹中创建一个名为模板的文件夹。应该是**your\_name/retail\_store/taobao/templates**。现在，在模板内部创建另一个名为“淘宝”的文件夹。**your\_name/retail\_store/taobao/templates/taobao**。

最后，在该淘宝文件夹中创建一个文件名为 **home\_page.html**。请在您的终端输入以下内容

`your_name/retail_store/taobao/templates/taobao/home_page..html` .

`your_name/retail_store/taobao/templates/taobao/home_page.html`

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Kool Karen Taobao</title>`
5. `</head>`
6. `<body>`
7. `<h1>Kool Karen Retail Store</h1>`
8. `<p>Your convenience is our duty</p>`
- 9.
10. `<h2>Here are all our items!</h2>`
- 11.
12. `<!-- Item 1 -->`
13. `<p>Water: 12RMB</p>`
14. `` 1
15. `<br>`
- 16.
17. `<!-- Item 2 -->`
18. `<p>Toothbrush: 7RMB</p>`
19. ``
- 20.
21. `<!-- Item 3 -->`
22. `<p>Chips: 10RMB</p>`
23. ``
- 24.
25. `<!-- Order Form -->`



```

26. <h2>Place your Order</h2>
27. <form action="{% url 'taobao:home_page' %}" method= "POST"> 2
28.     {% csrf_token %} 3
29.     <input type="text" name="item" placeholder="Please Type Which Item
    You Want"><br> 4
30.     <textarea name="address" placeholder="Please Type Your
    Address" ></textarea><br> 5
31.     <input type="submit" value="Place Your Order" >
32. </form>
33.
34. <!-- Show Order -->
35. <h1>Your Order:</h1>
36. <p>Item Ordered: {{item}}</p>
37. <p>Delivery Address: {{address}}</p>
38.
39. </body>
40. </html>

```

## 添加图片

添加图片很简单。使用**bing.com**查找图片。然后右击图片，然后单击“复制图像位置”。网址必须来自实际网站，而不是**Bing**。然后，在模板内部，只要输入**IMG 1**并按**Tab**键，就可以在任何地方插入图片。**img html**标记应该自动完成，将光标放置在**src=""**内粘贴图像的**URL**地址所需的位置。

## 形式

### 动作属性

这其中的特别之处在于窗口。要使窗口正常运行，最重要的是在窗口的开始标记中包含一个具有正确设置值的**action**属性。在我们的操作中，我们必须将值设置为“**{% url 'taobao:home\_page' %}**”。这个值意味着这个请求应该在“淘宝”应用程序中发送名为“**home\_page**”的url。这些都在我们的设置 **~/your\_name/retail\_store/taobao/urls.py** 中

### 方法属性

在我们的窗口中，开始标记也应该是一个方法属性，并设置为“**post**”值。

这是为了提出请求，一个发布请求。

## CSRF标注

窗口元素中的第二个重要内容是 `{% csrf_token %}` 3 .

根据Django的说法，**CSRF** 标注：

提供易于使用的保护，以防止[Cross Site Request Forgeries](#)。当恶意网站包含一个链接、一个窗口按钮或一些**JavaScript**，并使用在浏览器中访问恶意站点登录的用户的凭据在您的网站上执行某些操作时，就会发生这种攻击。一个相关类型的攻击，“login CSRF ”，攻击站点欺骗用户的浏览器，用其登陆并使用其他人的凭据登录用户信息。

欲知更多<https://docs.djangoproject.com/en/2.0/ref/csrf/>.

名称属性

在name属性。我们将每个**name**属性分别设置为“**Item**”和“**Address**”。名称属性是指用户如何在**views.py**中提供输入的值。如果您记得在**views.py**中，我们有行代码是**request.POST.get('item')** and **request.POST.get('address')**。这允许我们获取用户在**views.py**中的每个输入或文本框中输入的值。在其他情况下，**home\_page.html**输入的值或文本区域名称属性中的值必须与传递给**views.py**中**request.POST.get()**方法的参数相对应。

运行您的服务器

一旦您输入了**HOME\_Page.html**，您就完成了！恭喜！我们现在只需要运行我们的服务器和检查我们的网站。回到您的终端去，确保您的虚拟环境已被激活，并且您与**manage.py** (**your\_name/retail\_store/**)位于同一个文件夹中。然后输入以下命令来运行您的服务器；**python manage.py runserver**。  
(<---不包括句点)

```
(django_and_machine_learning_venv) KaludeMacBook-  
Pro:retail_store kalukalu$ python manage.py runserver
```

然后右键单击**127.0.0.0 : 8000** url地址，然后单击复制url地址。粘贴地址到您的网页浏览器地址栏，并欣赏您的网站！

