

Django项目实例详解(一)

在本节我们继续以项目实例的形式带领小伙伴们使 Django 进行相关的 Web 开发任务，在《[Django HTML表单实例应用](#)》一节中，我们使用 HTML 表单完成了书籍的查询功能。在本节我们将以完整的逻辑呈现图书的管理操作，本节的代码会较多，希望小伙伴们多加领悟，善于总结，并有所收获。下面我们正式开始本节的内容。

1. 获取全量书籍信息

将书籍的信息以表格的信息呈现的前端页面，通过页面中的增加、删除按钮对表中的每条数据进行操作。这是本节要实现的主体功能。我们首先将书籍的信息按照表格的形式展现出来。在 `index/templates` 中编写 `book_table.html` 页面，代码如下所示：


```
1. <html lang="en">
2. <head>
3.     <meta charset="UTF-8">
4.     <title>书籍信息表</title>
5. </head>
6. <body>
7.     <table border="1">
8.         <tr>
9.             <th>id</th>
10.            <th>title</th>
11.            <th>price</th>
12.            <th>retail_price</th>
13.            <th>public</th>
14.            <th>option</th>
15.        </tr>
16.        {% for book in all_book %}
17.        <tr>
18.            <td>{{ book.id }}</td>
19.            <td>{{ book.title }}</td>
20.            <td>{{ book.price }}</td>
21.            <td>{{ book.retail_price }}</td>
22.            <td>{{ book.pub.pubname }}</td>
23.            <td>
24.                <a href="/index/update_book/{{ book.id }}">更新</a>
25.                <a href="/index/delete_book/{{ book.id }}">删除</a>
26.            </td>
27.        </tr>
28.        {% endfor %}
29.    </table>
30. </body>
```

31. </html>

上述代码中我们我们构建了一个有关书籍信息表的表格，下面就需要在 index.views.py 文件中编写视图层代码逻辑了，代码如下所示：

```
1. """
2. time: 22/6
3. item: Book CURD
4. """
5. def book_table(request):
6.     try:
7.         all_book=Book.objects.all().order_by('-price')
8.         if not all_book:
9.             return HttpResponse('书籍信息表为空，请录入！')
10.    except Exception as e:
11.        print(e)
12.    return render(request, 'index/book_table.html', locals())
```

并配置路由映射关系，最后访问 http://127.0.0.1:8000/index/all_book/ 可得如下界面显示：



| id | title | price | retail_price | public | option |
|----|---------------|-------|--------------|----------|---------------------------------------|
| 4 | Django | 65.00 | 145.00 | c语言中文网出版 | 更新 删除 |
| 1 | Python Django | 59.00 | 139.00 | c语言中文网出版 | 更新 删除 |
| 3 | Java | 45.00 | 125.00 | c语言中文网出版 | 更新 删除 |
| 5 | Flask | 45.00 | 125.00 | c语言中文网出版 | 更新 删除 |
| 2 | Redis | 25.00 | 105.00 | c语言中文网出版 | 更新 删除 |

图1：Django实例书籍信息表

2. 实现添加书籍信息

接下来实现书籍信息的添加功能，编写 add_book.html 页面，代码如下所示：

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <title>添加书籍信息</title>
6. </head>
7. <body>
8.     <form action="/index/add_book/" method="post">
9.     {% csrf_token %}
10.     <p>
11.         title : <input type="text" name="title">
12.     </p>
13.     <p>
14.         price : <input type="text" name="price">
15.     </p>
16.     <p>
17.         retail_price : <input type="text" name="retail_price">
18.     </p>
19.     <p>
20.         public : <input type="text" name="pub">
21.     </p>
22.     <p>
23.         <!--按钮的值提交--> <input type="submit" value="提交">
24.     </p>
25. </form>
26. </body>
27. </html>
```

上述我们编写完成了添加书籍信息的页面，然后需要在视图层 index¥views.py 中编写提交书籍信息的逻辑代码。如下所示：

```
1. def add_book(request):
2.     if request.method == 'GET':
3.         return render(request, 'index/add_book.html')
4.     elif request.method == 'POST':
5.         #添加书籍
6.         title = request.POST.get('title')
7.         if not title:
8.             return HttpResponse('请给出一个正确的title')
9.         pub = request.POST.get('pub')
10.        price = float(request.POST.get('price', '999.99'))
11.        if not price:
12.            return HttpResponse('请输入价格')
13.        try:
14.            retail_price = float(request.POST.get('retail_price'))
```

```
15.         if not retail_price:
16.             return HttpResponseRedirect(' 请输入市场价')
17.     except Exception as e:
18.         print(e)
19.
20.     #判断title是不是已经存在了
21.     old_book = Book.objects.filter(title=title)
22.     if old_book:
23.         return HttpResponseRedirect(' 你输入的书籍系统已经存在 !')
24.     try:
25.         pub1=PubName.objects.get(pubname=str(pub))
26.         Book.objects.create(title=title,price=price,retail_price=retail_price,pub=pub1)
27.     except Exception as e:
28.         print(' Add ErrorReason is %s'%(e))
29.         return HttpResponseRedirect('/index/all_book')
30.     return HttpResponseRedirect(' 请使用正确Http请求方法 !')
```

最后将路由映射关系配置完成，通过上面的代码我们就完成书籍信息的添加的功能，访问 127.0.0.1/index/add_book,可得如下界面：



添加书籍信息

书籍名: C语言入门到精通

出版社: c语言中文网出版

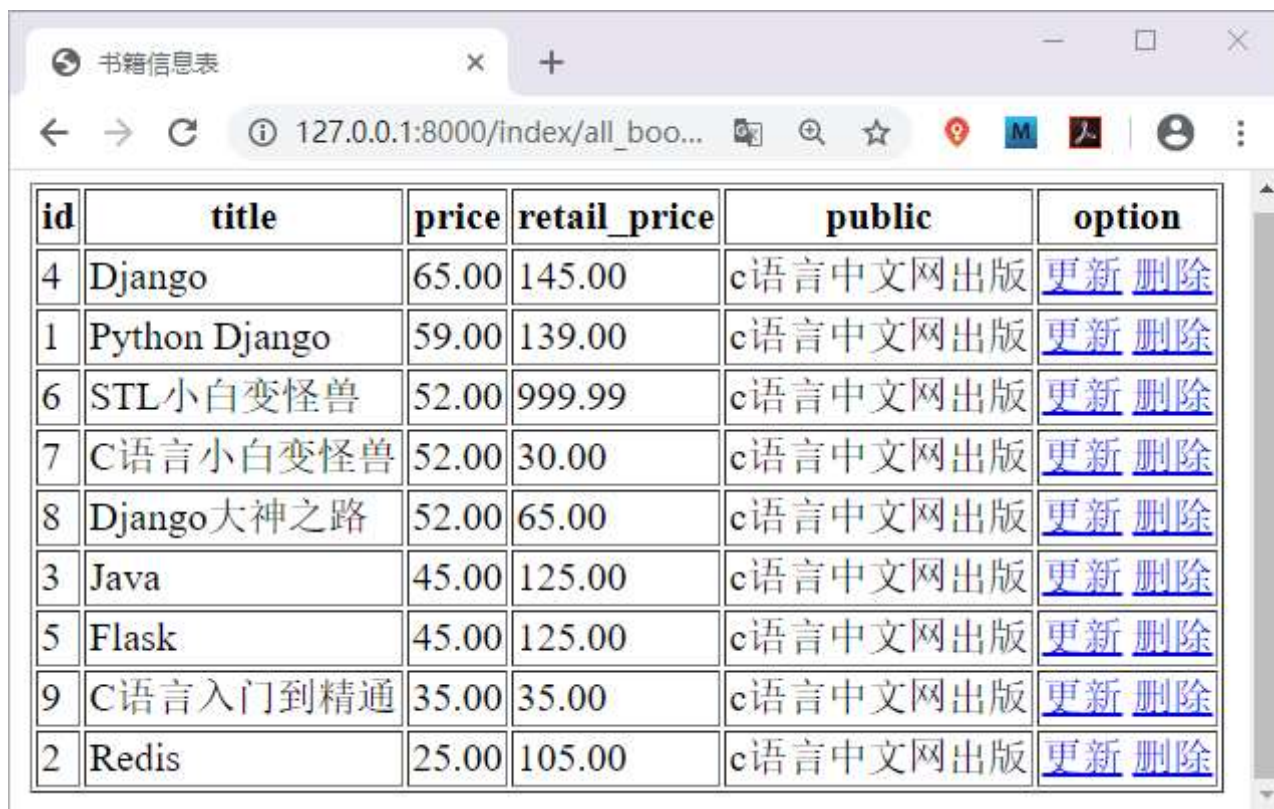
零售价: 35.00

市场价: 35.00

提交

图2：Django实例图书添加功能

输入完上述信息点击添加按钮，会自动跳转到图书浏览目录页面，如下所示：



| id | title | price | retail_price | public | option |
|----|---------------|-------|--------------|----------|---------------------------------------|
| 4 | Django | 65.00 | 145.00 | c语言中文网出版 | 更新 删除 |
| 1 | Python Django | 59.00 | 139.00 | c语言中文网出版 | 更新 删除 |
| 6 | STL小白变怪兽 | 52.00 | 999.99 | c语言中文网出版 | 更新 删除 |
| 7 | C语言小白变怪兽 | 52.00 | 30.00 | c语言中文网出版 | 更新 删除 |
| 8 | Django大神之路 | 52.00 | 65.00 | c语言中文网出版 | 更新 删除 |
| 3 | Java | 45.00 | 125.00 | c语言中文网出版 | 更新 删除 |
| 5 | Flask | 45.00 | 125.00 | c语言中文网出版 | 更新 删除 |
| 9 | C语言入门到精通 | 35.00 | 35.00 | c语言中文网出版 | 更新 删除 |
| 2 | Redis | 25.00 | 105.00 | c语言中文网出版 | 更新 删除 |

图3：Django实例图书添加功能

在本节中，我们实现了书籍的全量展示功能以及图书的添加功能。上述代码理解起来会比较的直观简单，但是实际编写起来，会有很多需要注意的细节，需要开发者做到思维缜密，当然这个能力并非一蹴而就，需要大家不断的练习与自我提升才可以。在下一节中，我们将继续讲解 Django 项目的实例应用，希望大家通过讲解能够熟练的使用Django 并将其作为 Web 开发的工具。