

哈尔滨工业大学计算机科学与技术学院
2015年秋季学期《软件工程》

Lab 3：MVC编程与云平台部署

姓名	学号	联系方式
董頔	1130310224	seancasey@sina.cn /15754604434

目 录

1. 实验要求.....	I
2. 开发环境配置.....	I
3. 图书SaaS设计	II
3.1.Settings.py.....	II
3.2.urls.py	V
3.3.View中的各方法	VI
3.4.辅助类	VI
3.5. Templates	VII
3.6.各view/template之间的调用和消息传递关系	VII
4. 图书SaaS核心代码.....	VIII
4.1.按作者查询	VIII
4.2.展示图书详细信息	VIII
4.3.删除图书.....	IX
4.4.新增图书/作者(可选)	X
4.5.更新图书信息(可选)	XI
4.6.数据库连接与访问	XIII
5. 图书SaaS的SAE部署	XIV
5.1.步骤	XIV
5.2.外在访问结果.....	XVII
6. 计划与实际进度.....	XXI
7. 小结.....	XXI

[文档全部完成之后，请更新上述区域]

1. 实验要求

- 掌握MVC架构下SaaS开发的基本流程;
- 开发环境:Python+Django+SAE;
- 在Python编程环境中搭建Django开发环境,开发一个小型SaaS,在 Web页面中对数据库中的数据进行CRUD(增加Create、读取Retrieve、更新Update和删除Delete)操作;
- 部署在SAE上并对外发布

开发要求

❑ 使用MySQL或SQLite建立一个图书数据库“BookDB”,包含两张表: – Book {ISBN (PK), Title, AuthorID (FK), Publisher, PublishDate, Price}

– Author {AuthorID (PK), Name, Age, Country}

❑ 手工输入足够多的若干测试数据;

❑ 开发以下功能:

- 输入作者名字,查询该作者所著的全部图书的题目;
- 当用户点击某本图书的题目时,展示图书详细信息和作者详细信息; – 当用户点击“删除”按钮时,将对应行的图书从数据表中删除;
- (opt)用户可新增一本图书,若该书作者不在库中,还需增加新作者;
- (opt)用户可更新一本图书的作者、出版社、出版日期、价格。

简要复述实验手册中要求达到的实验目标与过程。

2. 开发环境配置

```
Last login: Mon Nov  2 09:24:38 on ttys000
d_d:~ d_d$ brew install mysql
==> Downloading https://homebrew.bintray.com/bottles/mysql-5.6.27.el_capitan.bottle.tar.gz
##### 100.0%
==> Pouring mysql-5.6.27.el_capitan.bottle.tar.gz
==> Caveats
A "/etc/my.cnf" from another install may interfere with a Homebrew-built
server starting up correctly.

To connect:
  mysql -uroot

To have launchd start mysql at login:
  ln -sfv /usr/local/opt/mysql/*.plist ~/Library/LaunchAgents
Then to load mysql now:
  launchctl load ~/Library/LaunchAgents/homebrew.mxcl.mysql.plist
Or, if you don't want/need launchctl, you can just run:
  mysql.server start
==> Summary
📦 /usr/local/Cellar/mysql/5.6.27: 9883 files, 339M
```

MYSQL的安装

```
d_d:d_dlib d_d$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.6.27 Homebrew

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database BookDB;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| BookDB |
| mysql |
| performance_schema |
| test |
+-----+
5 rows in set (0.04 sec)

mysql> exit
Bye
```

数据库BookDB创建;

```
d_d:libman d_d$ pip install django
Requirement already satisfied (use --upgrade to upgrade): django in /Library/Python/2.7/site-packages
d_d:libman d_d$
```

Django 1.8.5 安装

3. 图书SaaS设计

3.1.Settings.py

```
"""
Django settings for d_dlib project.

Generated by 'django-admin startproject' using Django 1.8.5.

For more information on this file, see
https://docs.djangoproject.com/en/1.8/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/1.8/ref/settings/
"""
```

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
import os

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
import os.path
from os import environ
debug = not environ.get("APP_NAME", "")
if debug:
    MYSQL_DB = 'BookDB'
    MYSQL_USER = 'root'
    MYSQL_PASS = ''
    MYSQL_HOST_M = '127.0.0.1'
    MYSQL_HOST_S = '127.0.0.1'
    MYSQL_PORT = '3306'
else:
    #SAE
    import sae.const
    MYSQL_DB = sae.const.MYSQL_DB
    MYSQL_USER = sae.const.MYSQL_USER
    MYSQL_PASS = sae.const.MYSQL_PASS
    MYSQL_HOST_M = sae.const.MYSQL_HOST
    MYSQL_HOST_S = sae.const.MYSQL_HOST_S
    MYSQL_PORT = sae.const.MYSQL_PORT

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.8/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '0g6fd-i8*y@7xk!bvvszwmw86ebn8l8tw%s!ladcdmi%^e462sp'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = False

ALLOWED_HOSTS = ['*']

# Application definition

INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'libman',
)

MIDDLEWARE_CLASSES = (
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.auth.middleware.SessionAuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'django.middleware.security.SecurityMiddleware',
)

ROOT_URLCONF = 'd_dlib.urls'

TEMPLATES = [
```

```

    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

```

```
WSGI_APPLICATION = 'd_dlib.wsgi.application'
```

```

# Database
# https://docs.djangoproject.com/en/1.8/ref/settings/#databases

```

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': MYSQL_DB,
        'USER': MYSQL_USER,
        'PASSWORD': MYSQL_PASS,
        'HOST': MYSQL_HOST_M,
        'PORT': MYSQL_PORT,
    }
}

```

```

# Internationalization
# https://docs.djangoproject.com/en/1.8/topics/i18n/

```

```
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

```

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.8/howto/static-files/
STATIC_ROOT=os.path.join(os.path.dirname(__file__),'../static')
STATIC_URL = '/static/'

```

源代码(py)

附加关键部分的说明

已经更改为部署在SAE上的Settings，数据库的连接有文件开始的if条件语句控制。

3.2.urls.py

共有两个urls.py

这是总的：

```
"""d_dlib URL Configuration
```

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/1.8/topics/http/urls/>

Examples:

Function views

1. Add an import: from my_app import views
2. Add a URL to urlpatterns: url(r'^\$', views.home, name='home')

Class-based views

1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: url(r'^\$', Home.as_view(), name='home')

Including another URLconf

1. Add an import: from blog import urls as blog_urls
2. Add a URL to urlpatterns: url(r'^blog/', include(blog_urls))

```
"""
```

```
from django.conf.urls import include, url
from django.contrib import admin
```

```
urlpatterns = [
    url(r'',include('libman.urls',namespace='libman')),
    url(r'^admin/', include(admin.site.urls)),
]
```

上面的这个urlpatterns包含着下面这个文件。这是App内部的：

```
from django.conf.urls import url
from . import views
urlpatterns=[
    url(r'^$',views.index, name='index'),
    url(r'^(?P<bookisbn>[0-9-]+)/$',views.details, name='details'),
    url(r'^delete/$',views.delete,name='delete'),
    url(r'^edit/$',views.edit,name='edit'),
    url(r'^add_book/$',views.add_book,name='add_book'),
    url(r'^add_author/$',views.add_author,name='add_author'),
]
```

源代码(py)

附加关键部分的说明

功能包括展示主页，得到图书具体信息，删掉图书，编辑，加书，加作者。

3.3.View中的各方法

序号	方法名称	作用
1	index(request)	显示主页，先是搜索结果，完成搜索功能
2	edit(request)	编辑书目信息
3	details(request, bookisbn)	查看书目信息
4	delete(request)	删掉一本书
5	add_book(request)	增加一本书
6	add_author(request)	增加一个作者

3.4.辅助类

序号	方法名称	作用
无	无	无

3.5. Templates

序号	页面名	作用	页面核心元素 (form)	Form对应的 action name	Form中提交的数据项
1	index.html	主页	删除	delete/	delete_book_ISBN
			编辑	edit/	update_book_ISBN
			按作者搜索	.	author_name
			增加一本书	add_book/	add_book_isbn
					add_book_title
					add_book_authorid
					add_book_publisher
					add_book_publishdate
					add_book_price
2	edit.html	编辑书目信息	编辑书目信息	.	update_book_isbn
					update_book_title
					update_book_authorid
					update_book_publisher
					update_book_publishdate
					update_book_price
3	details.html	书目详细信息	无	无	无
4	add_author.html	添加作者	作者信息	.	AuthorID
					Name
					Age
					Country

3.6. 各view/template之间的调用和消息传递关系

给出一张图示描述彼此之间的数据传递和调用关系。

4. 图书SaaS核心代码

针对下列功能，分别给出view类和template页面中的核心代码。

4.1.按作者查询

```

6 def index(request):
7     book_list=Book.objects.all()
8     author_list=Author.objects.all()
9     if request.method == 'POST':
10         search_name = request.POST.get('author_name','')
11         try:
12             author = Author.objects.get(Name=search_name)
13             book_list_of_author = Book.objects.filter(AuthorID=author.AuthorID)
14             context = {'book_list': book_list, 'author_lib':book_list_of_author, 'author_name':author.Name, 'author_list':author_list}
15         except:
16             book_list_of_author = ['empty']
17             context = {'book_list': book_list, 'author_empty':book_list_of_author, 'author_name':search_name, 'author_list':author_list}
18     else:
19         context = {'book_list': book_list, 'author_list':author_list}
20     if add[0] and add[1]:
21         context['bad_parameters']=['not_empty']
22         add[1]=False
23     return render(request, 'libman/index.html', context)

```

9-19行为View中的核心：

```

38 <div id="content" class="colM">
39     <div id="content-main" style='margin-bottom:20px;'>
40         <form action="." method='POST' id='author_search'>
41             {% csrf_token %}
42             <input type='text' name='author_name' />
43             <input type='submit' value='Author Search' />
44         </form>
45         <label>
46             Exist Authors: {% for author in author_list %} {{author.Name}} {% endfor %}
47         </label>
48         {% if author_lib %}
49         <p>Author {{author_name}}'s book(s)</p>
50         <ul>
51             {% for j in author_lib %}
52             <li><a href="{% url 'libman:details' j.ISBN %}">{{ j.Title }}</a></li>
53             {% endfor %}
54         </ul>
55         {% endif %} {% if author_empty %}
56         <p>Author {{author_name}} not found.</p>
57         {% endif %}
58     </div>
59 </div>

```

上图为index.html作者查询部分核心代码

4.2.展示图书详细信息

```

47 def details(request, bookisbn):
48     book=Book.objects.get(ISBN=bookisbn)
49     context={'book':book}
50     return render(request, 'libman/details.html', context)

```

上图为view中展示图书详细信息代码

```

9 <body>
10 {% if book %}
11 <div id="content" class="colM">
12 <div id="content-main">
13 <form action="update" method="post" id="book_form" novalidate>
14 <input type="hidden" name="csrfmiddlewaretoken" value='ruGXkx0X0knTGEVHiHgM40zPXj2RGojf' />
15 <div>
16 <fieldset class="module aligned">
17 <div class="form-row field-ISBN">
18 <div>
19 <label class="required" for="id_ISBN">ISBN:</label>
20 {{ book.ISBN }}
21 </div>
22 </div>
23 <div class="form-row field-Title">
24 <div>
25 <label class="required" for="id_Title">Title:</label>
26 {{ book.Title }}
27 </div>
28 </div>
29 <div class="form-row field-AuthorID">
30 <div>
31 <label class="required" for="id_AuthorID">AuthorID:</label>
32 <div class="related-widget-wrapper">
33 {{book.AuthorID}}
34 </div>
35 </div>
36 </div>
37 <div class="form-row field-Publisher">
38 <div>
39 <label class="required" for="id_Publisher">Publisher:</label>
40 {{ book.Publisher }}
41 </div>
42 </div>
43 <div class="form-row field-PublishDate">
44 <div>
45 <label class="required" for="id_PublishDate">PublishDate:</label>
46 {{ book.PublishDate }}
47 </div>
48 </div>
49 <div class="form-row field-Price">
50 <div>
51 <label class="required" for="id_Price">Price:</label>
52 {{ book.Price }}
53 </div>
54 </div>
55 </fieldset>

```

上图为details核心代码。

4.3.删除图书

```

52 def delete(request):
53     try:
54         Book.objects.get(ISBN=request.GET.get('delete_book_ISBN')).delete()
55     except:
56         pass
57     return HttpResponseRedirect('/')

```

如图为删除部分；

```

12 <table>
13     {% for i in book_list %}
14     <tr>
15         <td>
16             <ul>
17                 <li><a href="{% url 'libman:details' i.ISBN %}">{{ i.Title }}</a></li>
18             </ul>
19         </td>
20         <td>
21             <form action='delete/' method='GET'>
22                 <input type='hidden' name='delete_book_ISBN' value={{ i.ISBN }} />
23                 <input type='submit' value='Delete' />
24             </form>
25         </td>
26         <td>
27             <form action='edit/' method='GET'>
28                 <input type='hidden' name='update_book_ISBN' value={{ i.ISBN }} />
29                 <input type='submit' value='Edit' />
30             </form>
31         </td>
32     </tr>
33     {% endfor %}
34 </table>
35 {% else %}
36 <p>No books are available.</p>
37 {% endif %}

```

图中21-24行为index.html中的删除部分：

4.4.新增图书/作者(可选)

```

59 def add_book(request):
60     add[0]=True
61     get=request.GET
62     try:
63         new_book=Book(
64             ISBN=get['add_book_isbn'],
65             Title=get['add_book_title'],
66             AuthorID=Author.objects.get(AuthorID=get['add_book_authorid']),
67             Publisher=get['add_book_publisher'],
68             PublishDate=get['add_book_publishdate'],
69             Price=get['add_book_price']
70         )
71         new_book.save()
72         add[1]=False
73         return HttpResponseRedirect('/')
74     except:
75         add[1]=True
76         book_list=Book.objects.all()
77         author_list=Author.objects.all()
78         return HttpResponseRedirect('/')

```

图中为views中的增加一本书的代码：

```

63 <div id="content" class="colM">
64     <div id="content-main">
65         <form action="add_book/" method="GET" id="new_book_form" novalidate>
66             <input type="hidden" name="csrfmiddlewaretoken" value="ruGXkx0X0knTGEVHiHgM40zPXj2RGojf" />
67             <div>
68                 <fieldset class="module aligned">
69                     <div class="form-row field-ISBN">
70                         <div>
71                             <label class="required" for="id_ISBN">ISBN:</label>
72                             <input name="add_book_isbn" type="text" />
73                         </div>
74                     </div>
75                     <div class="form-row field-Title">
76                         <div>
77                             <label class="required" for="id_Title">Title:</label>
78                             <input name="add_book_title" type="text" />
79                         </div>
80                     </div>
81                     <div class="form-row field-AuthorID">
82                         <div>
83                             <label class="required" for="id_AuthorID">AuthorID:</label>
84                             <div class="related-widget-wrapper">
85                                 <select name="add_book_authorid">
86                                     {% for author in author_list %}
87                                     <option value={{author.AuthorID}}>{{author.AuthorID}}: {{author.Name}}</option>
88                                     {% endfor %}
89                                 </select>
90                                 <a class="related-widget-wrapper-link add-related" id="add_author" href="add_author/" title="Add another author">
91                                     
92                                 </a>
93                             </div>
94                         </div>
95                     </div>
96                     <div class="form-row field-Publisher">
97                         <div>
98                             <label class="required" for="id_Publisher">Publisher:</label>
99                             <input name="add_book_publisher" type="text" />
100                         </div>
101                     </div>
102                     <div class="form-row field-PublishDate">
103                         <div>
104                             <label class="required" for="id_PublishDate">PublishDate:</label>
105                             <input name="add_book_publishdate" type="text" placeholder="Format: yyyy-mm-dd" />
106                         </div>
107                     </div>
108                     <div class="form-row field-Price">
109                         <div>

```

```

100         </div>
101     </div>
102     <div class="form-row field-PublishDate">
103         <div>
104             <label class="required" for="id_PublishDate">PublishDate:</label>
105             <input name='add_book_publishdate' type='text' placeholder='Format: yyyy-mm-dd' />
106         </div>
107     </div>
108     <div class="form-row field-Price">
109         <div>
110             <label class="required" for="id_Price">Price:</label>
111             <input name='add_book_price' type='text' />
112         </div>
113     </div>
114 </fieldset>
115 <div class="submit-row">
116     <input type="submit" value="Add Book" class="default" name="add">
117 </div>
118 </div>
119 </form>
120 </div>

```

上面两张图为index.html中增加一本书的html代码。

4.5.更新图书信息(可选)

图中为编辑一本书的代码：

```

25 update_book_list=[]
26 def edit(request):
27     if request.method=='GET':
28         pass
29         book=Book.objects.get(ISBN=request.GET.get('update_book_ISBN'))
30         update_book_list.append(book)
31         author_list=Author.objects.all()
32         context={'book':book,'author_list':author_list}
33         return render(request,'libman/edit.html',context)
34     else:
35         try:
36             book=update_book_list[-1]
37             author=Author.objects.get(AuthorID=request.POST.get('update_book_authorid'))
38             book.AuthorID = author
39             book.Publisher = request.POST.get('update_book_publisher')
40             book.PublishDate = request.POST.get('update_book_publishdate')
41             book.Price = request.POST.get('update_book_price')
42             book.save()
43             return HttpResponseRedirect('/')
44         except:
45             return render(request,'libman/edit.html',{'bad_parameters':['not empty']})

```

```
add_author.html x details.html x views.py x edit.html x index.html x
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <link rel="stylesheet" type="text/css" href="/static/admin/css/base.css" />
6 <link rel="stylesheet" type="text/css" href="/static/admin/css/forms.css" />
7 </head>
8
9 <body>
10 {% if bad_parameters %}
11 <h1 style='margin-left:20px;color:red;'>Your book's info is not well formatted. Please try again!</h1>
12 <a style='margin-left:20px;' href="/" class="default">Back</a>
13 {% endif %}
14 {% if book %}
15 <div id="content" class="colM">
16 <div id="content-main">
17 <form action="." method="post" id="book_form" novalidate>
18 <input type="hidden" name="csrfmiddlewaretoken" value="ruGXlx0X0knTGEVHjHgM40zPXj2RGojf" />
19 <div>
20 <fieldset class="module aligned">
21 <div class="form-row field-ISBN">
22 <div>
23 <label class="required" for="id_ISBN">ISBN:</label>
24 {{ book.ISBN }}
25 </div>
26 </div>
27 <div class="form-row field-Title">
28 <div>
29 <label class="required" for="id_Title">Title:</label>
30 {{ book.Title }}
31 </div>
32 </div>
33 <div class="form-row field-AuthorID">
34 <div>
35 <label class="required" for="id_AuthorID">AuthorID:</label>
36 <div class="related-widget-wrapper">
37 {{book.AuthorID}}
38 <select name='update_book_authorid'>
39 {% for author in author_list %}
40 <option value={{author.AuthorID}}>{{author.AuthorID}}: {{author.Name}}</option>
41 {% endfor %}
42 </select>
43 </div>
44 </div>
45 </div>
46 <div class="form-row field-Publisher">
47 <div>
48 <label class="required" for="id_Publisher">Publisher:</label>
49 <input name='update_book_publisher' type='text' value='{{ book.Publisher }}' />
50 </div>

```

Line 45, Column 31

Spaces: 4

HTML

```
add_author.html x details.html x views.py x edit.html x index.html x
30 {{ book.Title }}
31 </div>
32 </div>
33 <div class="form-row field-AuthorID">
34 <div>
35 <label class="required" for="id_AuthorID">AuthorID:</label>
36 <div class="related-widget-wrapper">
37 {{book.AuthorID}}
38 <select name='update_book_authorid'>
39 {% for author in author_list %}
40 <option value={{author.AuthorID}}>{{author.AuthorID}}: {{author.Name}}</option>
41 {% endfor %}
42 </select>
43 </div>
44 </div>
45 </div>
46 <div class="form-row field-Publisher">
47 <div>
48 <label class="required" for="id_Publisher">Publisher:</label>
49 <input name='update_book_publisher' type='text' value='{{ book.Publisher }}' />
50 </div>
51 </div>
52 <div class="form-row field-PublishDate">
53 <div>
54 <label class="required" for="id_PublishDate">PublishDate:</label>
55 {{ book.PublishDate }}
56 <input name='update_book_publishdate' type='text' placeholder='Format: yyyy-mm-dd' />
57 </div>
58 </div>
59 <div class="form-row field-Price">
60 <div>
61 <label class="required" for="id_Price">Price:</label>
62 <input name='update_book_price' type='text' value='{{ book.Price }}' />
63 </div>
64 </div>
65 </fieldset>
66 <div class="submit-row">
67 <input type="submit" value="Save" class="default" name="save">
68 <a href="/" class="default">Cancel</a>
69 </div>
70 </div>
71 </form>
72 </div>
73 <br class="clear" />
74 </div>
75 </div>
76 {% endif %}
77 </body>
78 </html>

```

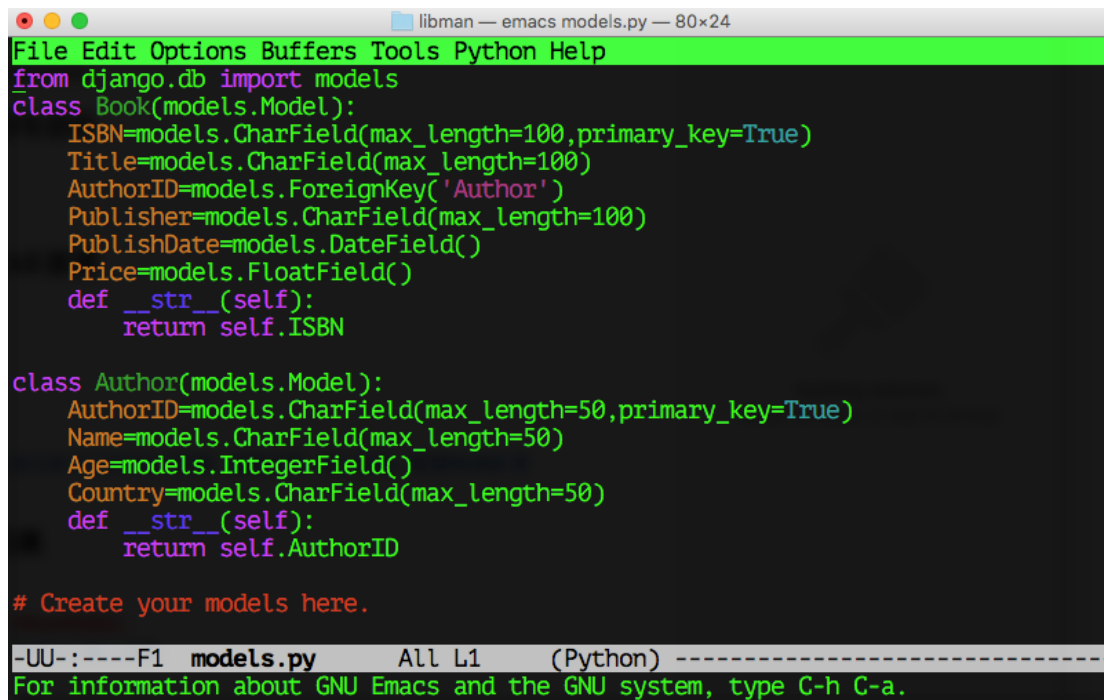
Line 45, Column 31

Spaces: 4

HTML

上面两图是编辑的html代码。

4.6.数据库连接与访问



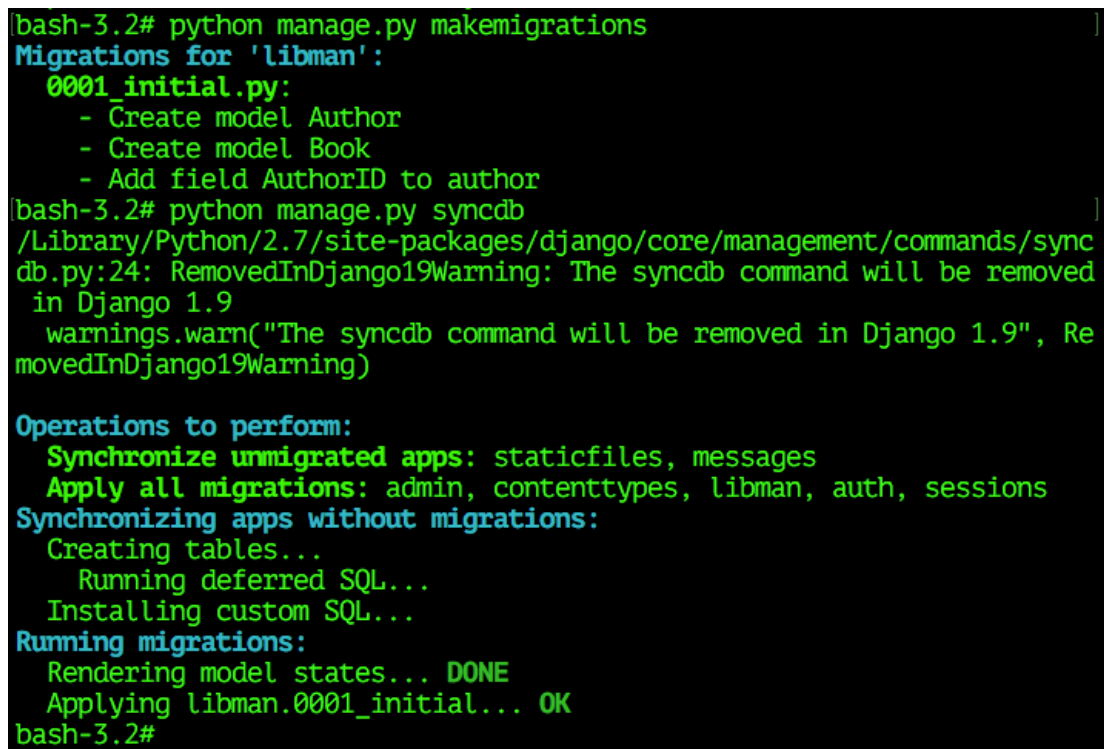
```
libman — emacs models.py — 80x24
File Edit Options Buffers Tools Python Help
from django.db import models
class Book(models.Model):
    ISBN=models.CharField(max_length=100,primary_key=True)
    Title=models.CharField(max_length=100)
    AuthorID=models.ForeignKey('Author')
    Publisher=models.CharField(max_length=100)
    PublishDate=models.DateField()
    Price=models.FloatField()
    def __str__(self):
        return self.ISBN

class Author(models.Model):
    AuthorID=models.CharField(max_length=50,primary_key=True)
    Name=models.CharField(max_length=50)
    Age=models.IntegerField()
    Country=models.CharField(max_length=50)
    def __str__(self):
        return self.AuthorID

# Create your models here.

-UU-:----F1 models.py All L1 (Python) -----
For information about GNU Emacs and the GNU system, type C-h C-a.
```

以上是Models里的两个基类。



```
bash-3.2# python manage.py makemigrations
Migrations for 'libman':
  0001_initial.py:
    - Create model Author
    - Create model Book
    - Add field AuthorID to author
bash-3.2# python manage.py syncdb
/Library/Python/2.7/site-packages/django/core/management/commands/syncdb.py:24: RemovedInDjango19Warning: The syncdb command will be removed in Django 1.9
  warnings.warn("The syncdb command will be removed in Django 1.9", RemovedInDjango19Warning)

Operations to perform:
  Synchronize unmigrated apps: staticfiles, messages
  Apply all migrations: admin, contenttypes, libman, auth, sessions
Synchronizing apps without migrations:
  Creating tables...
  Running deferred SQL...
  Installing custom SQL...
Running migrations:
  Rendering model states... DONE
  Applying libman.0001_initial... OK
bash-3.2#
```

迁移以及同步数据库；

```
mysql> show tables;
+-----+
| Tables_in_bookdb |
+-----+
| auth_group        |
| auth_group_permissions |
| auth_permission   |
| auth_user         |
| auth_user_groups  |
| auth_user_user_permissions |
| django_admin_log  |
| django_content_type |
| django_migrations |
| django_session    |
| libman_author     |
| libman_book       |
+-----+
12 rows in set (0.00 sec)

mysql> _
```

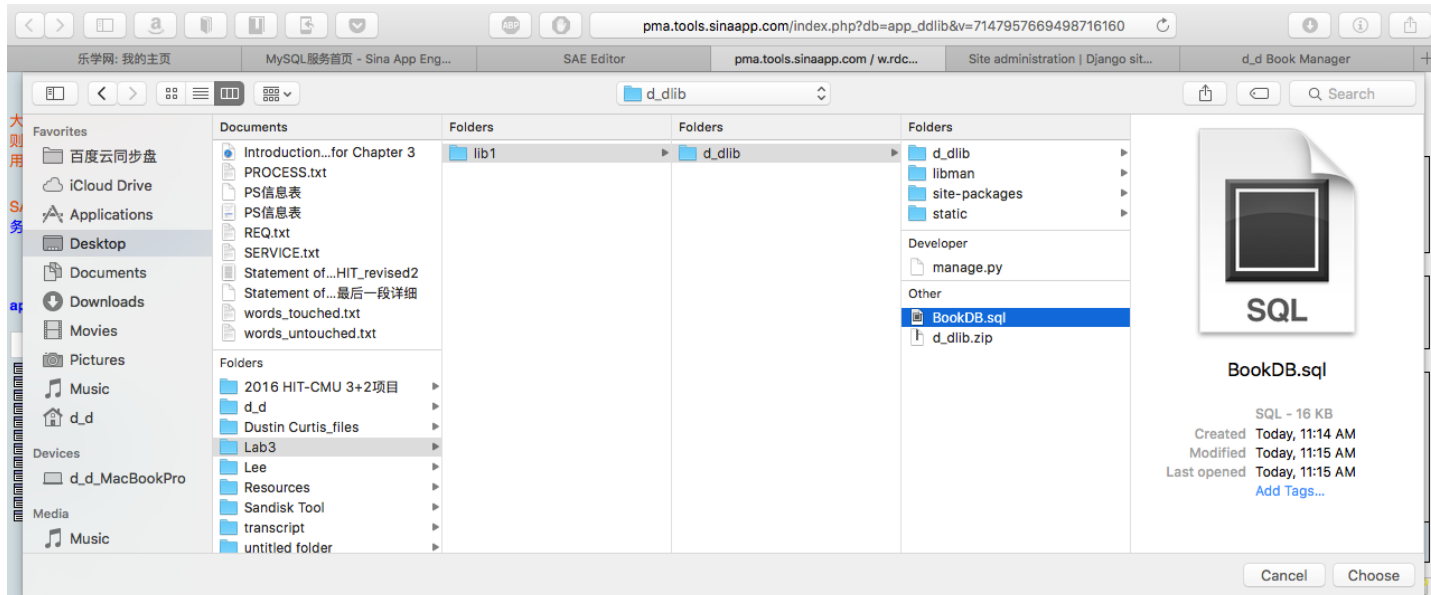
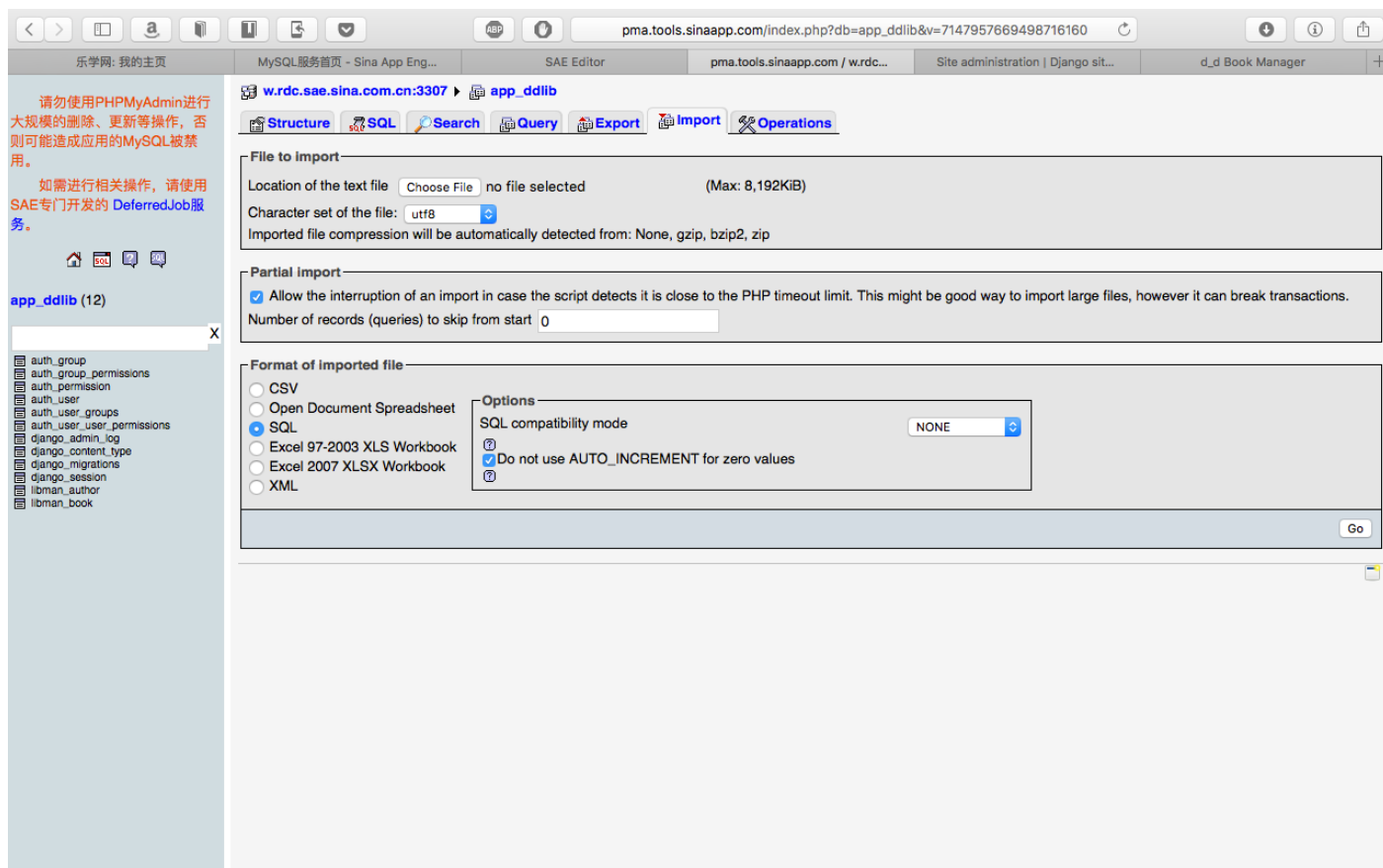
以上是已经建立好的表libman_author, libman_book。

5. 图书SaaS的SAE部署

5.1. 步骤

采用屏幕截图的方式给出你自己在SAE上注册、创建应用、导入应用、配置MySQL等步骤的过程。

由于早已注册，所以注册创建部分已经没有截图。



请勿使用PHPMyAdmin进行大规模的删除、更新等操作，否则可能造成应用的MySQL被禁用。

如需进行相关操作，请使用SAE专门开发的 [DeferredJob服务](#)。

app_ddlib (12)

- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_migrations
- django_session
- libman_author
- libman_book

Structure SQL Search Query Export Import Operations

Import has been successfully finished, 111 queries executed.

File to import

Location of the text file: no file selected (Max: 8,192KiB)

Character set of the file:

Imported file compression will be automatically detected from: None, gzip, bzip2, zip

Partial import

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. This might be good way to import large files, however it can break transactions.

Number of records (queries) to skip from start:

Format of imported file

☐ CSV
☐ Open Document Spreadsheet
☒ SQL
☐ Excel 97-2003 XLS Workbook
☐ Excel 2007 XLSX Workbook
☐ XML

Options

SQL compatibility mode:

☒ Do not use AUTO_INCREMENT for zero values

乐学网: 我的主页 代码管理 - Sina App Engine SAE Editor pma.tools.sinaapp.com / w.rdc... Site administration | Django sit... d_d Book Manager

新浪云 控制台 移动客户端 消费记录 工单 消息 支持中心 普通 Sean19

python **ddlib** 修改应用基本信息

2.7 web应用

代码管理

代码仓库--SVN

SVN仓库地址: <https://svn.sinaapp.com/ddlib/>

免费配额 已用: 16M/5G

提示 代码管理磁盘容量计费请参考[\[磁盘容量价格\]](#)

默认版本	版本创建时间	链接	操作
<input checked="" type="radio"/> 1	2015-11-04 11:21:15	http://1.ddlib.sinaapp.com	编辑代码 上传代码包 删除

该应用尚未启用CDN服务 您可以通过启用CDN，让您的用户获得更快的访问速度。 [\[启动CDN\]](#)

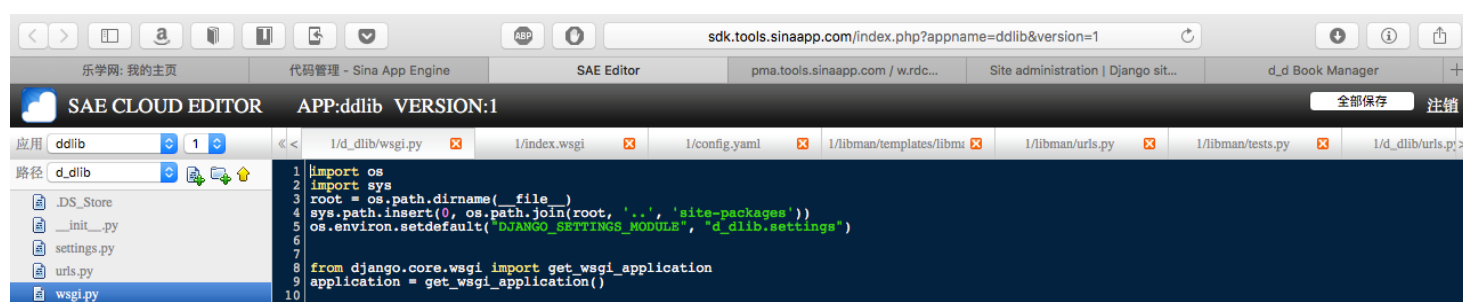
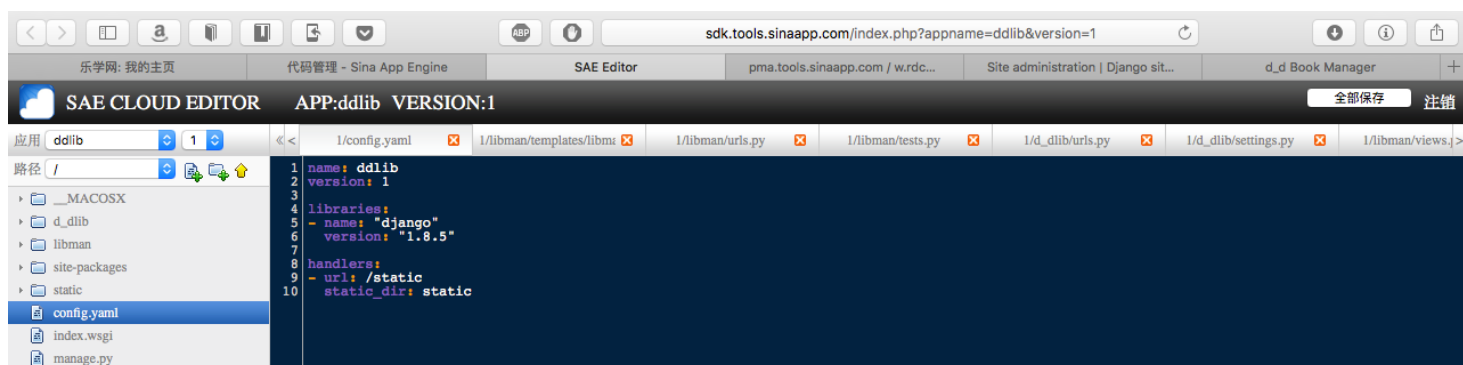
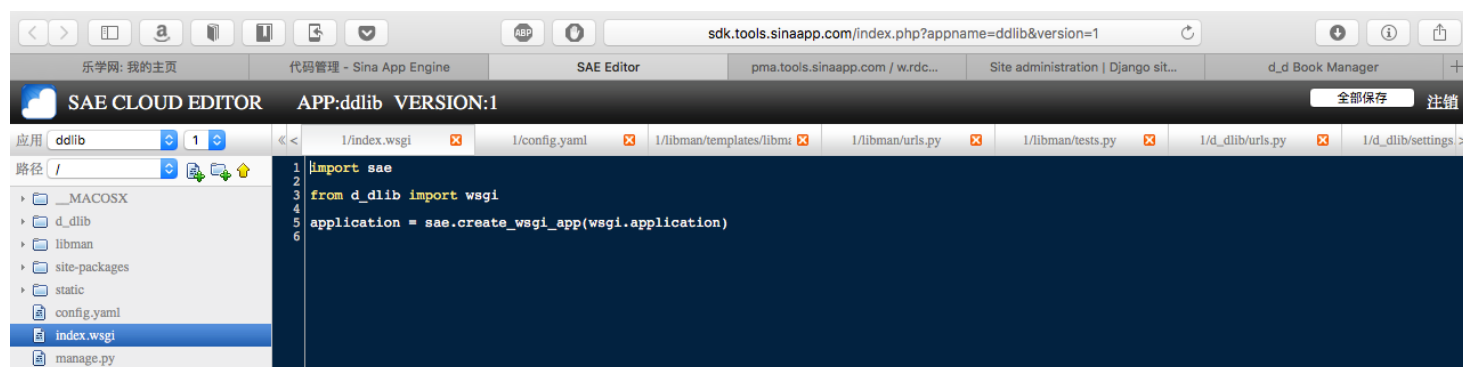
代码部署说明

首先checkout出\$appname的svn仓库。

```
$ svn checkout https://svn.sinaapp.com/$appname
```

Go to #CMD:EDIT-CODE on this page

```
d_d:libman d_d$ mysqldump -uroot -p BookDB > /Users/d_d/Desktop/Lab3/lib1/d_dlib
/BookDB.sql
Enter password:
d_d:libman d_d$ ls
__init__.py      admin.pyc        models.pyc       urls.py          views.pyc
__init__.pyc     migrations       templates        urls.pyc
admin.py         models.py        tests.py         views.py
d_d:libman d_d$ cd ..
d_d:d_dlib d_d$ ls
BookDB.sql      d_dlib.zip      manage.py        static
d_dlib          libman          site-packages
```



5.2.外在访问结果

给出访问SAE上的图书SaaS的地址;

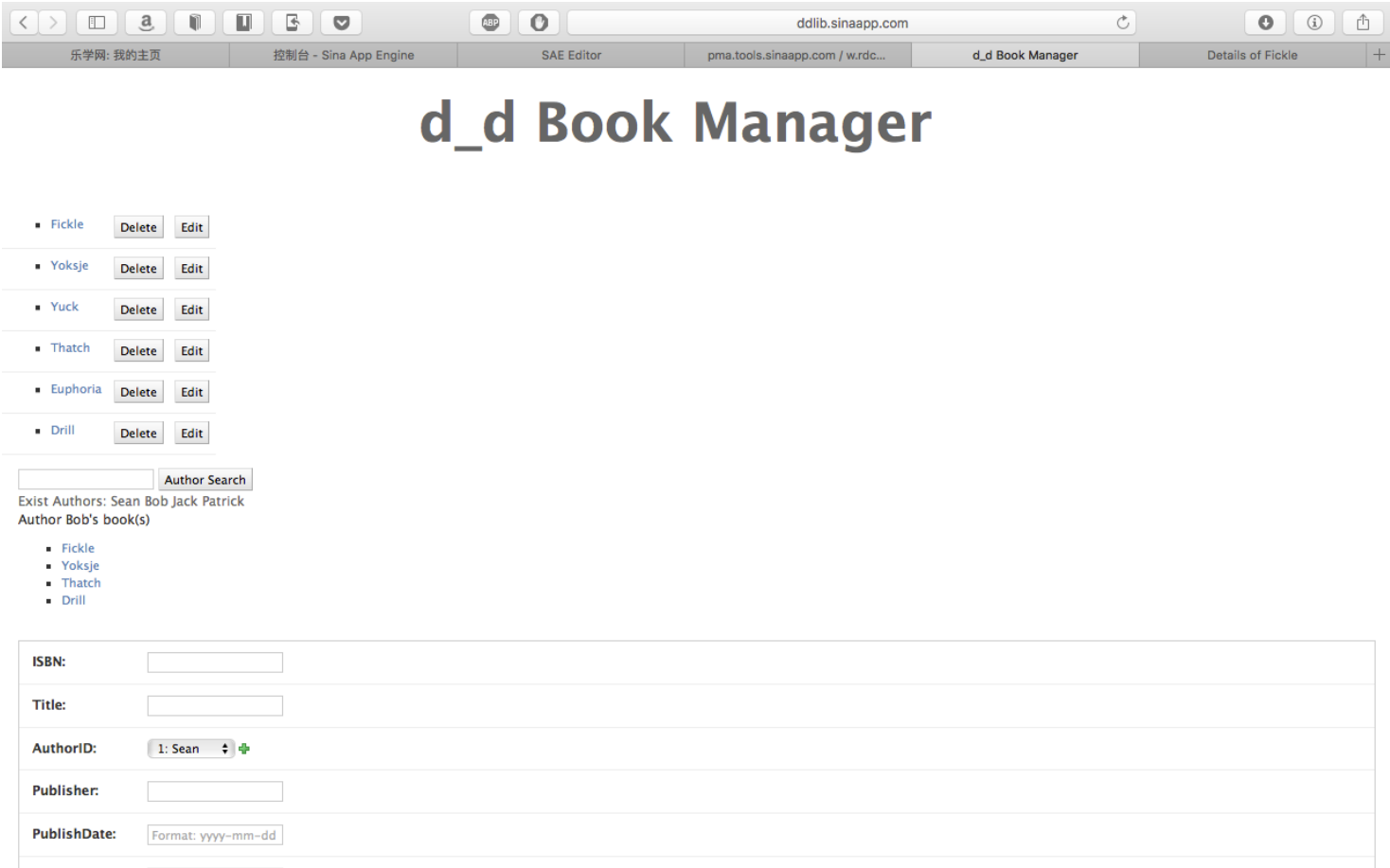
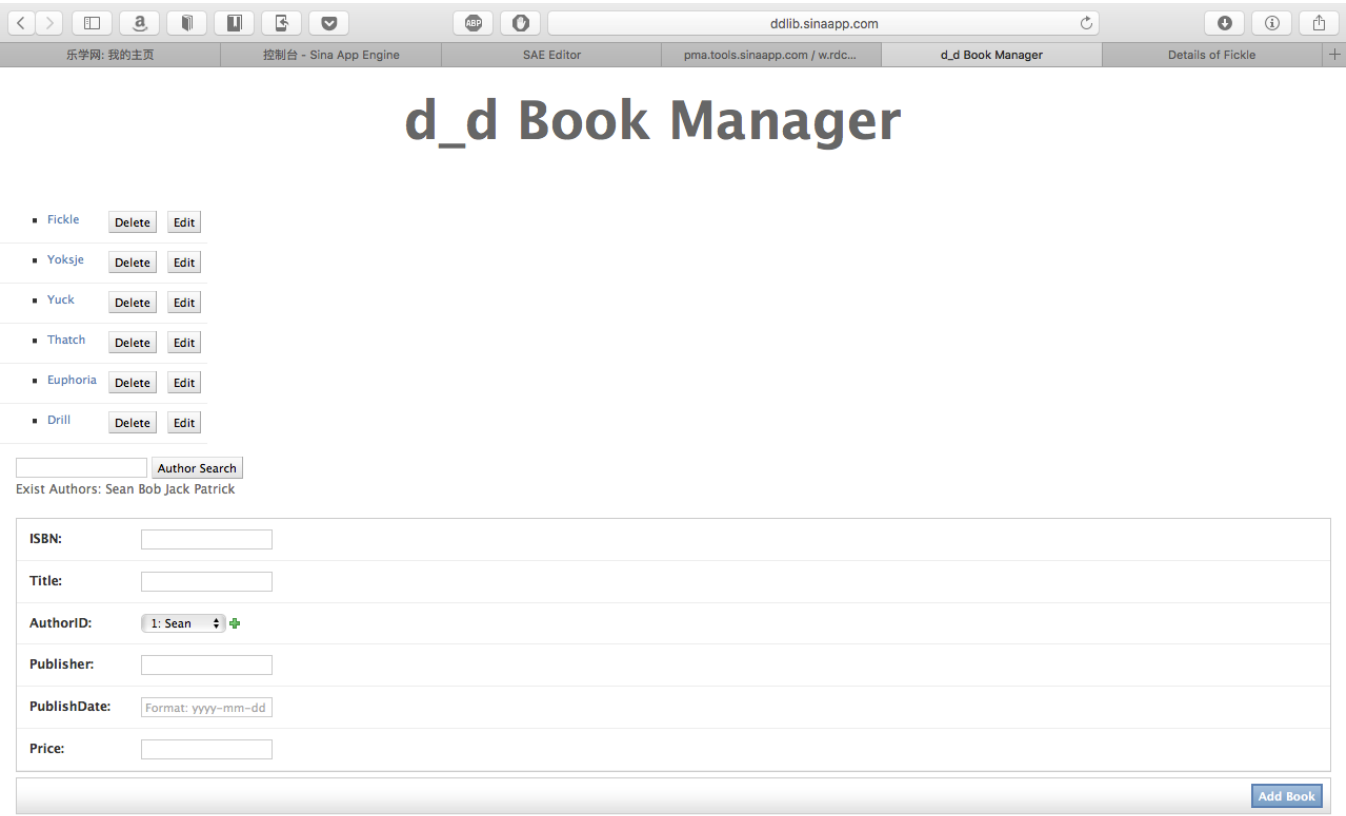
图书SaaS地址:

<http://ddlib.sinaapp.com>

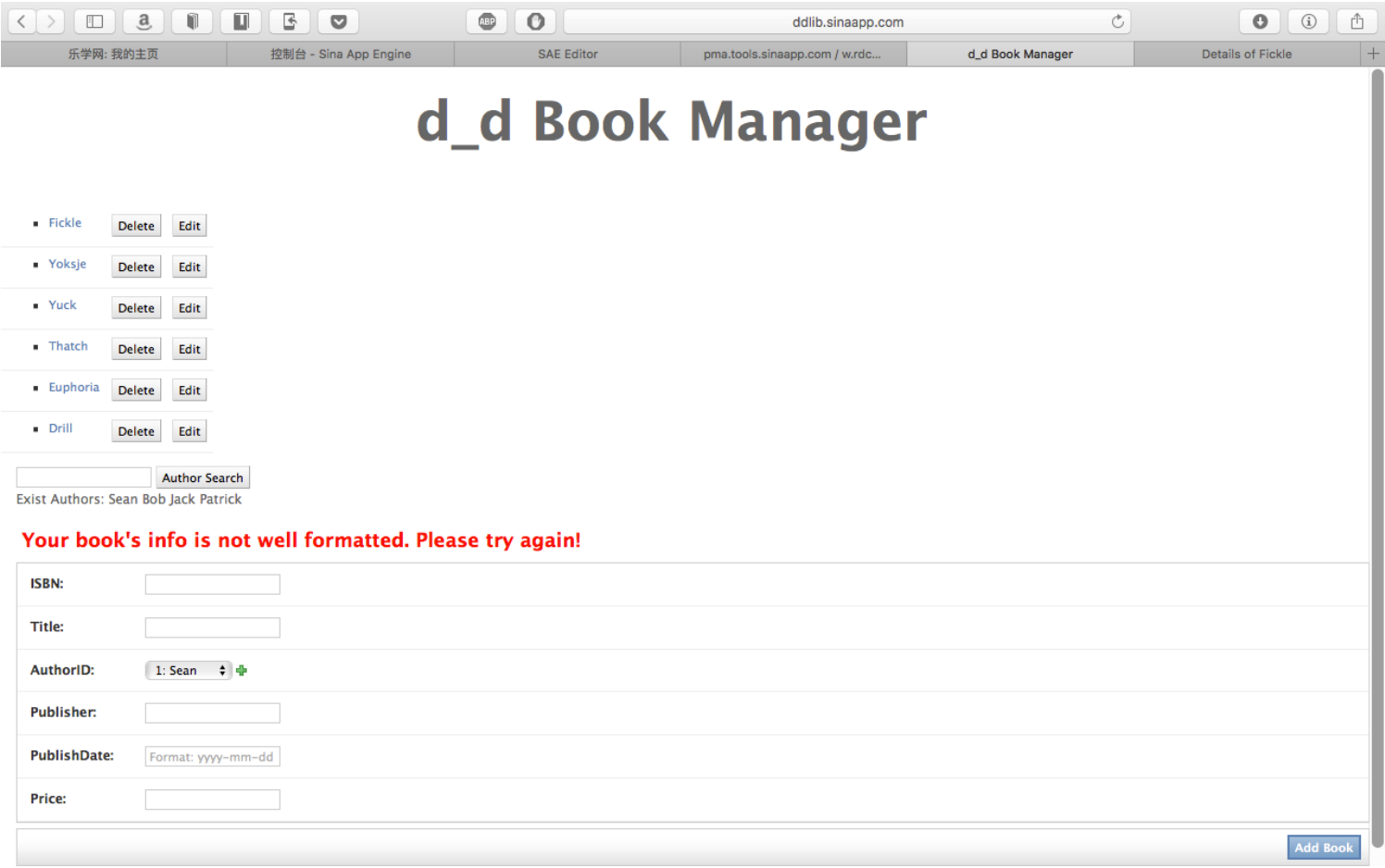
给出屏幕截图结果，要展示出核心功能。

主页：

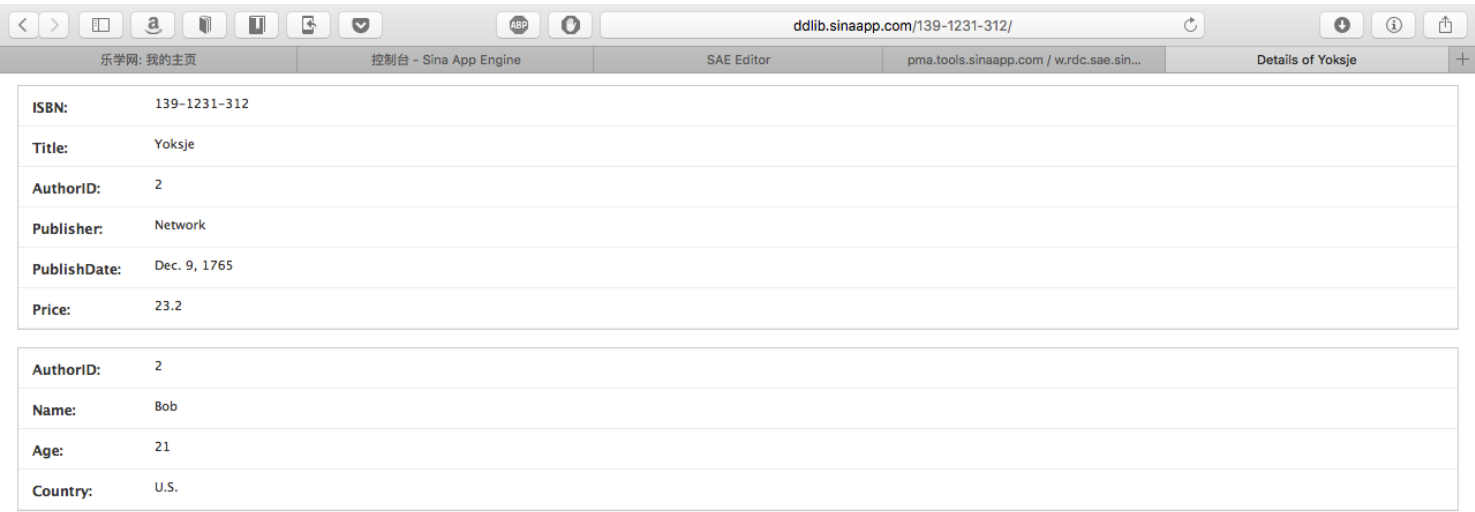
可以搜索，可以添加书，可以点击书名查看详细信息，可以编辑书，可以删除书。



上图为搜索：



此图为添加时的错误处理。



Back

具体功能请到ddlib.sinaapp.com亲自体验。

由于个人近期很忙，没有做过多美化，还请老师见谅。

6. 计划与实际进度

任务编号	计划时间长度（分钟）	实际耗费时间（分钟）	提前或延期的原因分析
1	2个小时	4个小时	没有构思好整体架构，导致写了又改改了又写。
2	3个小时	2个小时	有了思路后写得很快，加上部署只花了不到一个小时

7. 小结

对本次实验过程和结果的思考。

对Django有了更深层次的理解，更加清楚了Django处理请求的机制，也加深了对数据库语句等等的熟悉程度，对部署应用到SAE有了一个大概的了解。

总体来说收获很大，Django框架对于部署小型应用来说真的是很方便，基本就可以让程序员抛下请求网络等等的处理专心在自己的应用上。

因为近期在准备GRE，时间紧，所以只是大致将功能实现，没有写过多的CSS加以装饰，还请老师见谅。