# How to start Django

Aug 7, 2018.Xi ynag

Catalog

[How to start Django 1](#_Toc521489176)

[Basic Configure 3](#_Toc521489177)

[Some video urls for learning Django 5](#_Toc521489178)

[Set up a application 5](#_Toc521489179)

[Start a application: Hello world 6](#_Toc521489180)

[Set up a SQL database 8](#_Toc521489181)

[Django`s manage system \* 11](#_Toc521489182)

[A simple blog(An example) 13](#_Toc521489183)

[Home page 13](#_Toc521489184)

[Article page 14](#_Toc521489185)

[Edit page 16](#_Toc521489186)

## 

## Basic Configure

1、use command line :”*pip install django*” to download django (I use windows ,maybe it`s the same in linux.)

2、configure the PATH environment.Get the path of django(My django path is : C:\Users\luo xi yang\Anaconda3\Lib\site-packages\django\bin).As show in figure 1.

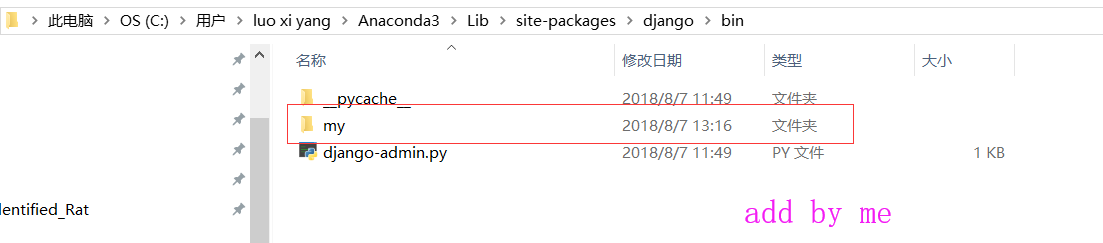


Figure 1

3、Add the PATH into ATH environment. As show in figure 2

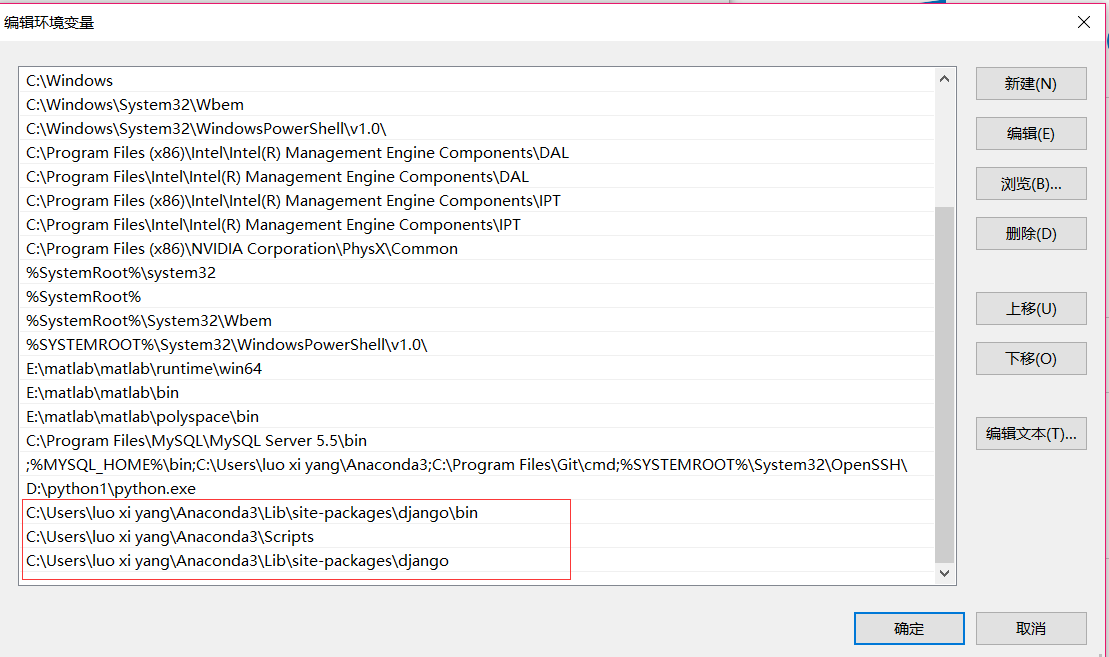


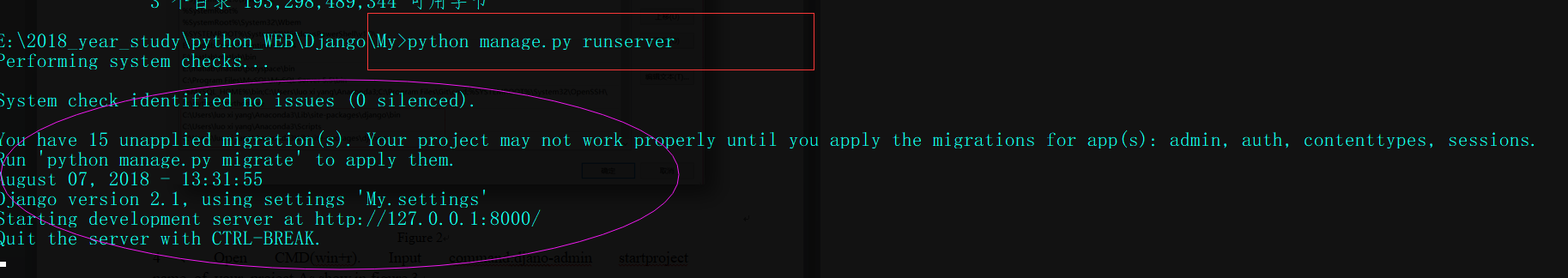
Figure 2

4、Open CMD(win+r). Input command: *djano-admin startproject name\_of\_your\_project.* As show in figure 3

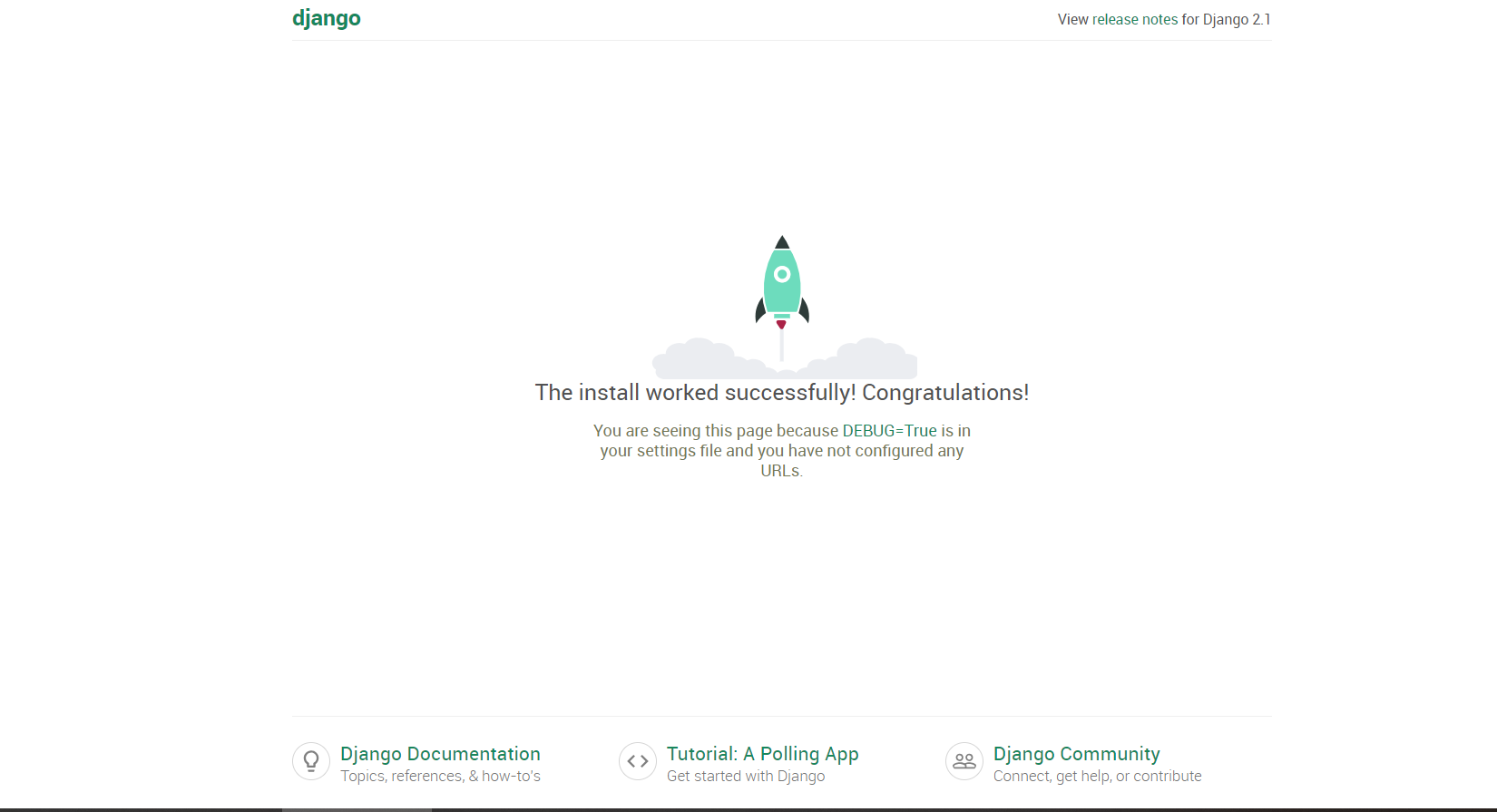


Figure 3

5、Use command :*python manage.py runserver* to start django .As show in figure 4.

 Figure 4

6、You can input http://127.0.0.1:8000 in browser , so you can see the sentence that django worked successfully! As show in figure 5.

 Figure 5

## Some video urls for learning Django

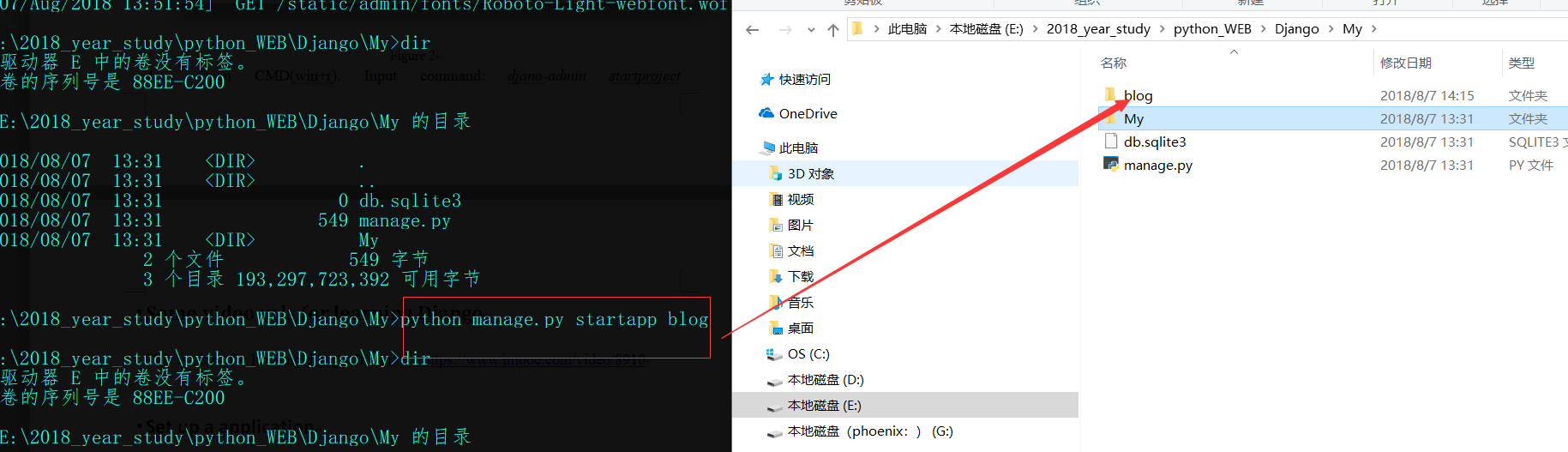
This is video for learning basic django. It`s url is :

<https://www.imooc.com/video/8910>

This is video for learning django. It`s url is : <https://www.imooc.com/video/13966>

## Set up a application

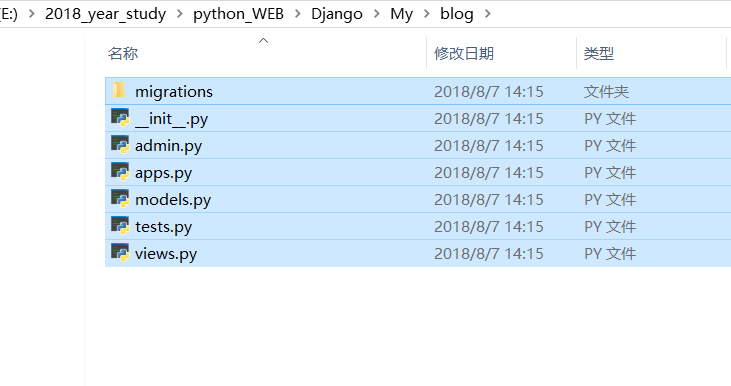
1、Typing command: *python manage.py startapp your app name* at the command line.Ok, a folder has been created in current folder. As shown in figure 6.

 Figure 6

2、Find settings.py and open it.Add your application into INSTALLED\_APPS.

 Figure 7

OK, application has been created.Some files in your application folder. As show in figure 8.

 Figure 8

## Start a application: Hello world

1、open file blog/views.py and input code. the code show in table 1.

Table 1 views.py

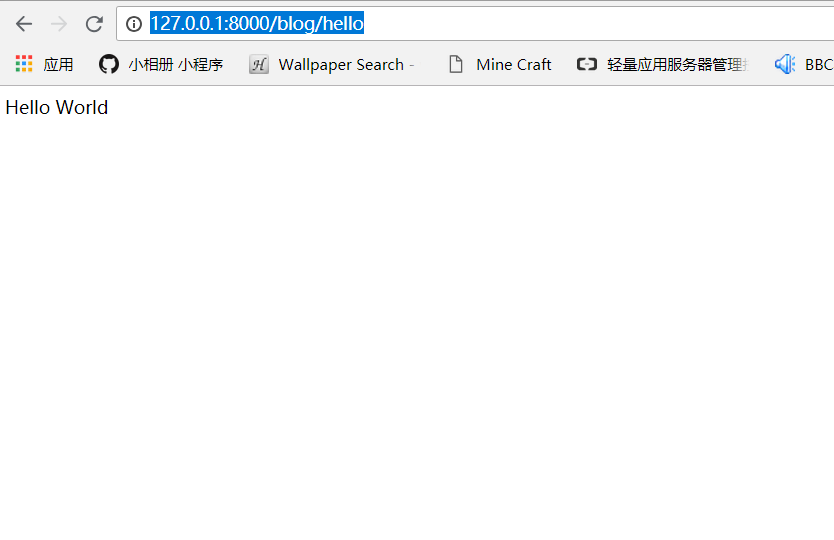
|  |
| --- |
| views.py |
| from django.shortcuts import render  from django.http import HttpResponse  # Create your views here.  def Hello(request):  return HttpResponse('<html>Hello World</html>') |

2、open file My/urls.py and input code, as show in table 2.

Table 2 urls.py

|  |
| --- |
| urls.py |
| """My URL Configuration  The `urlpatterns` list routes URLs to views. For more information please see:  https://docs.djangoproject.com/en/2.1/topics/http/urls/  Examples:  Function views  1. Add an import: from my\_app import views  2. Add a URL to urlpatterns: path('', views.home, name='home')  Class-based views  1. Add an import: from other\_app.views import Home  2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')  Including another URLconf  1. Import the include() function: from django.urls import include, path  2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))  """  from django.contrib import admin  from django.urls import path  from django.conf.urls import include,url  from blog.views import Hello  """  urlpatterns = [  path('admin/', admin.site.urls),  url(r'helloworld','blog.views.Hello')  ]  """  urlpatterns = [  url(r'^admin/',admin.site.urls),  url(r'^blog/hello',Hello)  ] |

Then use command to start the server. The command is “*python manage.py runserver*”.And you input url “*http://127.0.0.1:8000/blog/hello*” in browser.You will get the sentence, as show in figure 9.

Figure 9

## Set up a SQL database

This is video for learning django. It`s url is : <https://www.imooc.com/video/13966>.

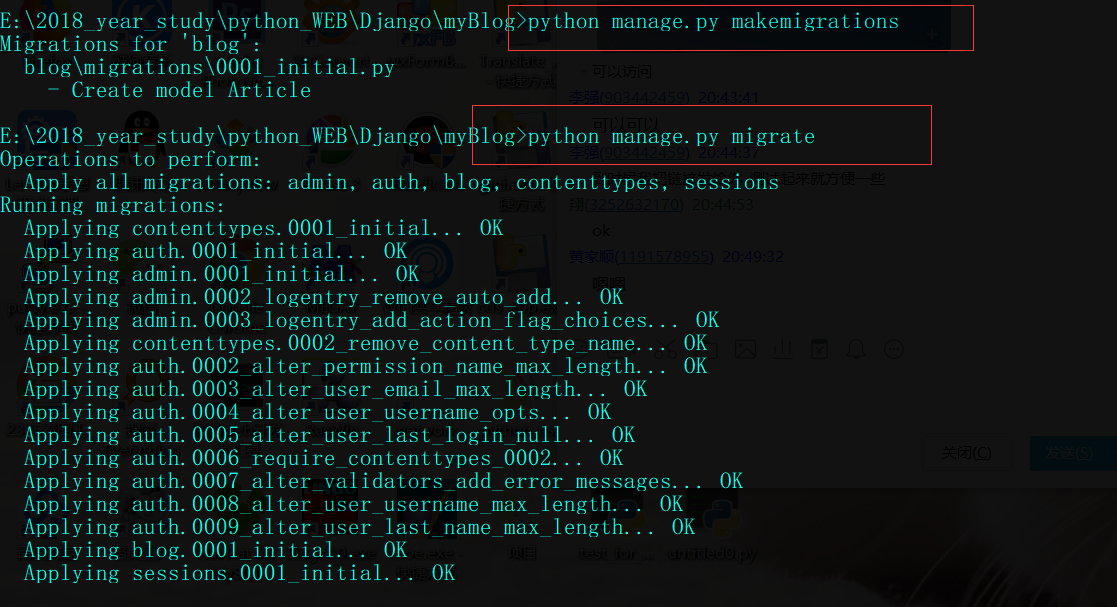
1、build a SQL (default SQL from Django, don`t change the setting.py). Write the codes into models.py where is in application folder. As show in table 3. You can learn the SQL of Django from this url :

<https://docs.djangoproject.com/en/2.1/ref/settings/#databases>

table 3

|  |
| --- |
| models.py |
| from django.db import models  # Create your models here.  class Article(models.Model):  title = models.CharField(max\_length=32,default="Title")  content =models.TextField(null=True)  def \_\_unicode\_\_(self):  return self.title |

2、We can use command :*python manage.py makemigrations* and *python manage.py migrate* to establish the database.The process show in figure 10

 Figure 10

After built the SQL database,There is a file named db.sqlite3 has been created in current folder. As show in figure 11.

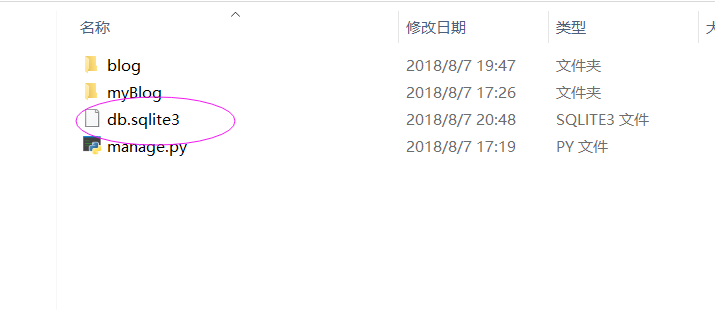
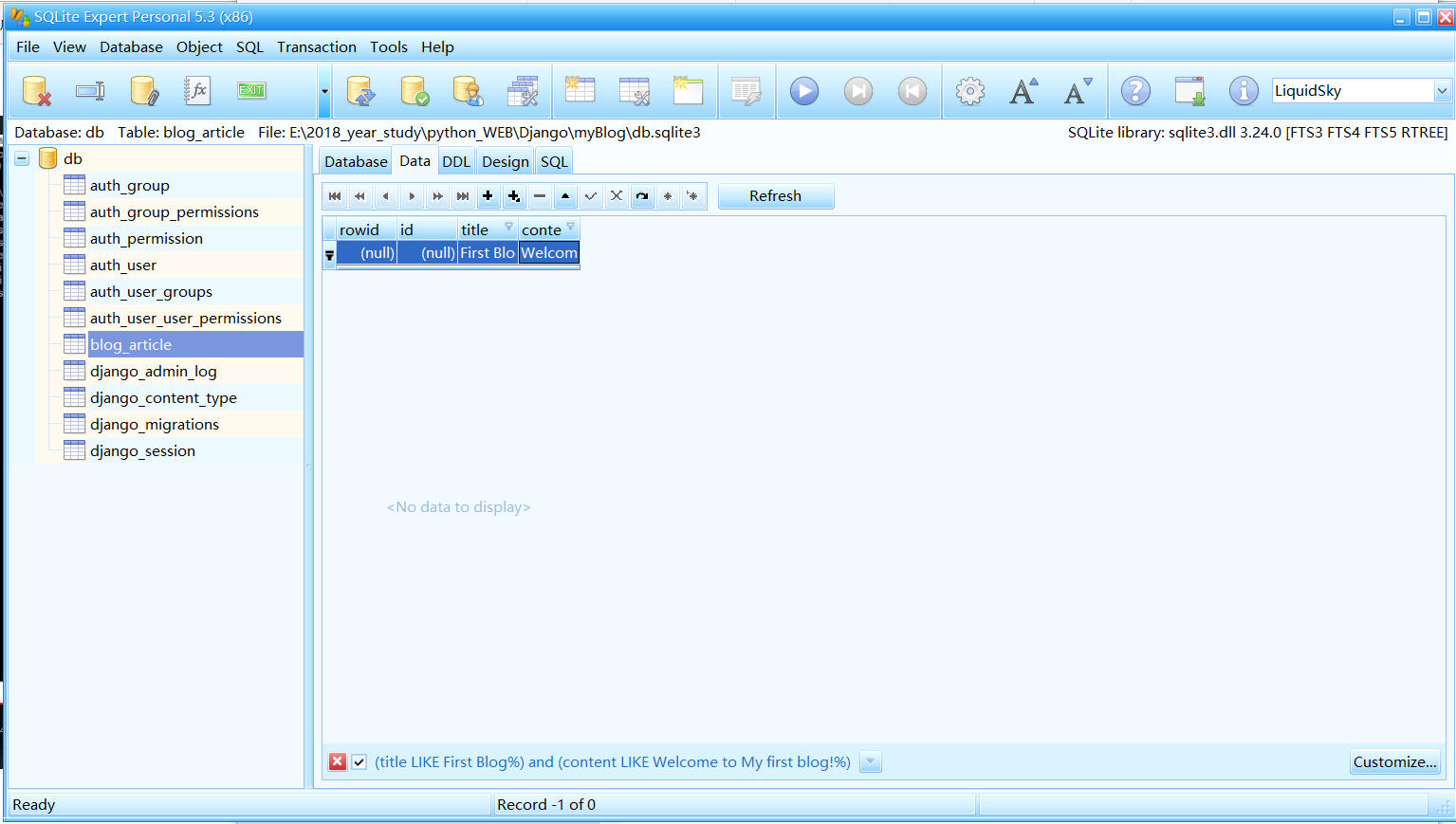


Figure 11

OK, you have built a SQL database. You can use a software named *SQLite expert personal* to manage the database, course. This is the url of software:

<http://www.softpedia.com/get/Internet/Servers/Database-Utils/?utm_source=spd&utm_campaign=postdl_redir>

You can download the software here. The software like this(figure 12).

 Figure 12

3、Use views.py to transmit data from SQL.

table 4

|  |
| --- |
| Views.py |
| from django.shortcuts import render  from django.http import HttpResponse  from . import models  # Create your views here.  def index(request):  # return HttpResponse('hello xiyang')  article = models.Article.objects.get(pk=1)  # if you want get all datas from database,use the code:  # article = models.Article.objects.all() return a list.  return render(request,"blog/index.html",{"article":article}) |

4、Of course, You must write a html to show out. Table 5 show the code. Command : *<h2>{{article.title}}</h2>* can obtain the data from database.

table 5

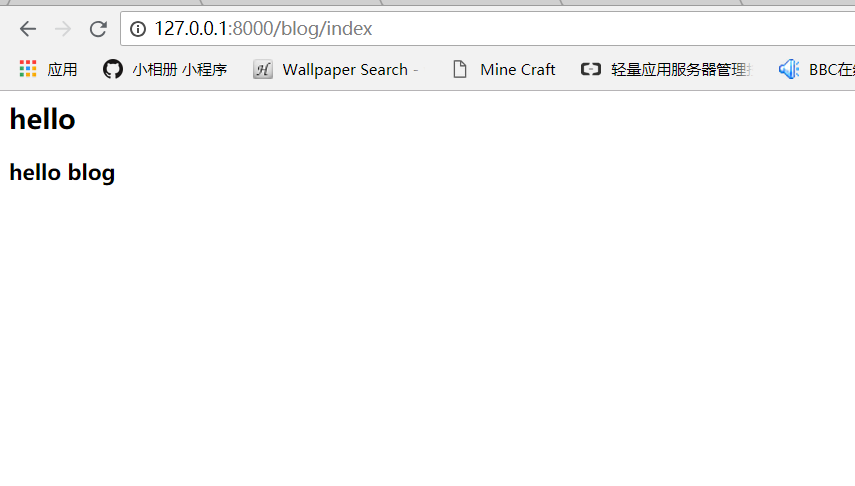
|  |
| --- |
| Index.html |
| <html>  <head>  <title>index</title>  </head>  <body>  <h2>{{article.title}}</h2>  <h3>{{article.content}}</h3>  </body>  </html> |

Lastly you need to configure the url .like table 6.

table 6

|  |
| --- |
| urls.py |
| from django.contrib import admin  from django.urls import path  from django.conf.urls import url,include  from blog.views import index#This is must.  #from blog.views import index  urlpatterns = [  url(r'^admin/', admin.site.urls),  url(r'^blog/index',index),  # url(r'^blog/',include('blog.urls')),  ] |

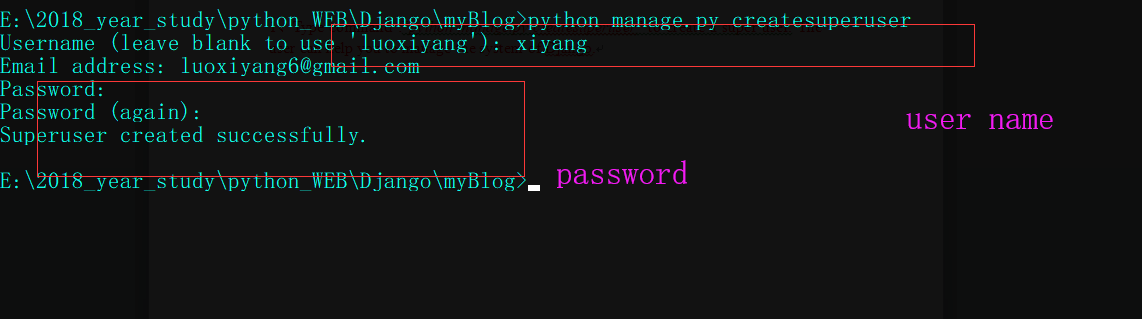
Well, type the command :*python manage.py runserver* to start server. And input the <url:http:127.0.0.1:8000/blog/index> in browser. It will return to the page like figure 13.

 Figure 13

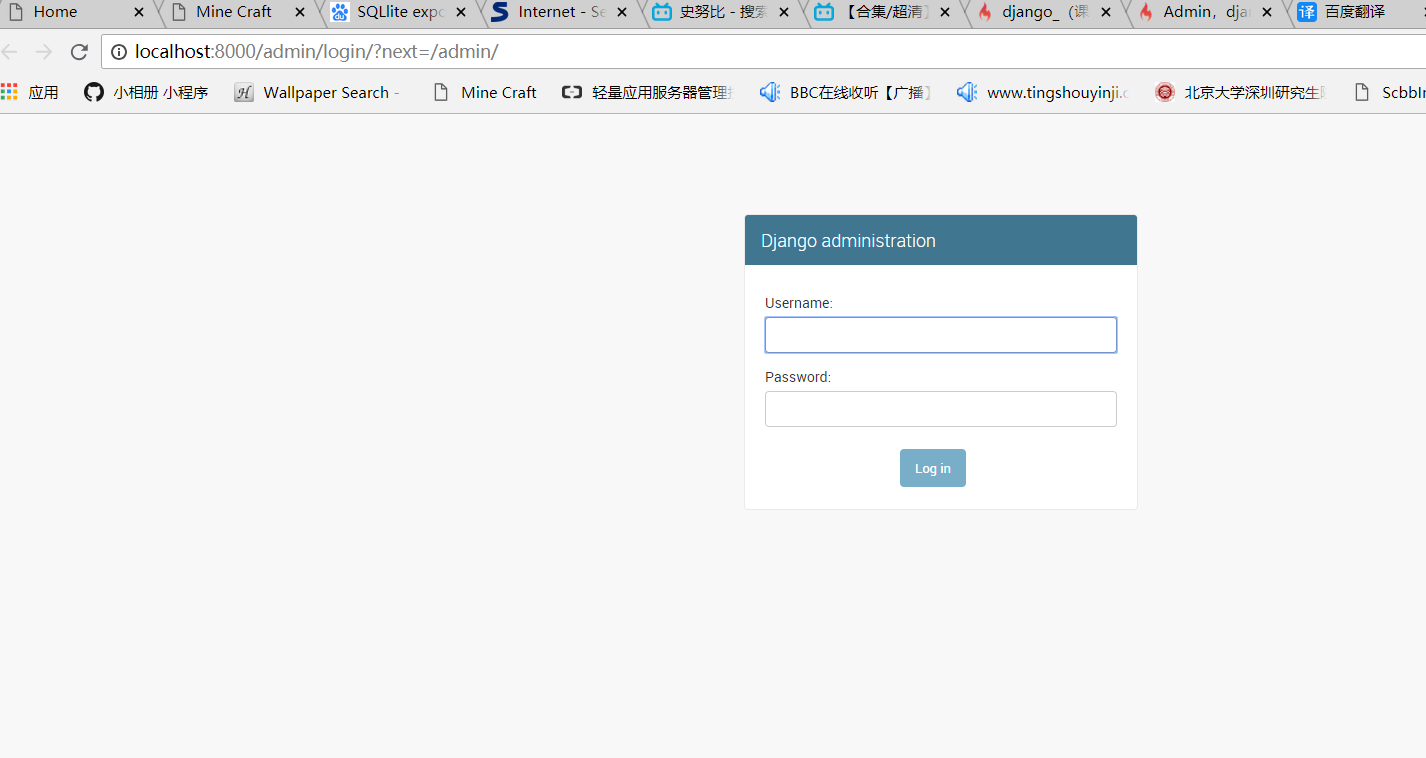
Well, you have built a your first blog.

## Django`s manage system \*

1、Type command :”*python manage.py createsuperuser*” to create a super user. The user can help you to manage the system of Django. The process like figure 14

Figure 14

2、Start the server and type “localhost:8000/admin/” in browser to visit the manage system. As show in figure 15.



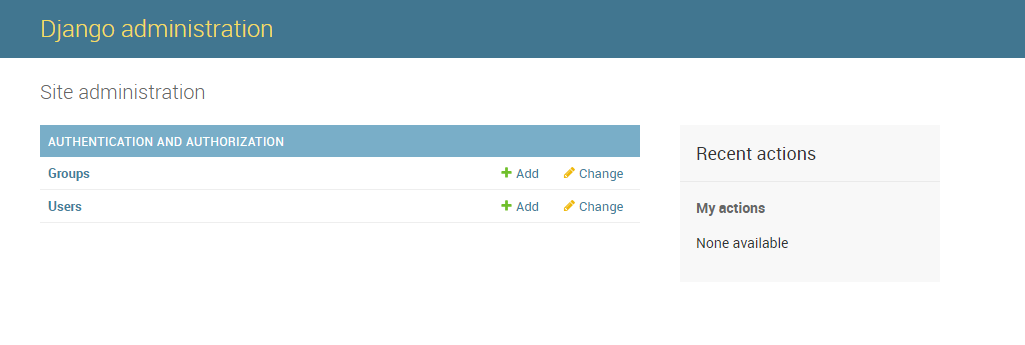


Figure 15

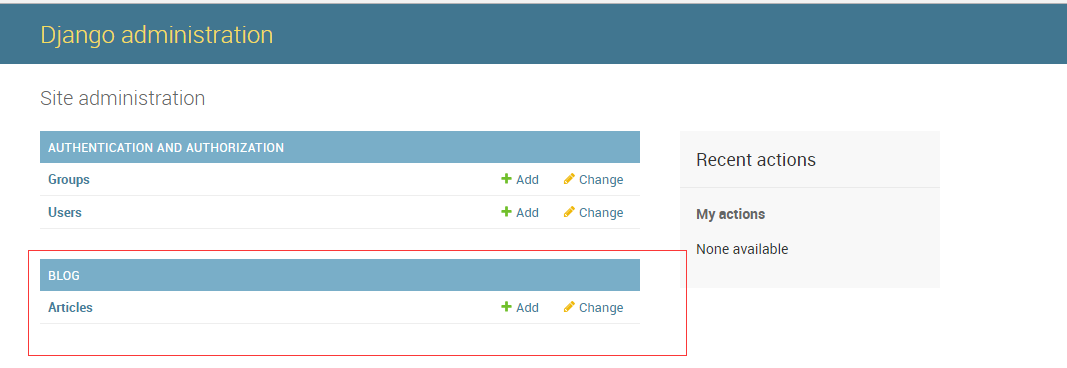
3、open file named admin.py and add code like table 7 into this file. Maybe there are some question here, you need to plus this command: *from . import models.* The page will help you, it`s url is :

<https://stackoverflow.com/questions/45020963/modulenotfounderror-no-module-named-models>

table 7

|  |
| --- |
| admin.py |
| from django.contrib import admin  # Register your models here.  from . import models  from .models import Article  admin.site.register(Article) |

Look the manage system again, you will find that the database has been plus into the manage system. As show in figure 16.

 Figure 16

## A simple blog(An example)

### Home page

1、Obtain the all data from database. You can obtain all data use command:” *article = models.Article.objects.all()*” in views.py. The detail code show in table 8.

table 8

|  |
| --- |
| Views.py |
| from django.shortcuts import render  from django.http import HttpResponse  from . import models  # Create your views here.  def index(request):  # return HttpResponse('hello xiyang')  articles = models.Article.objects.all()  return render(request,"blog/index.html",{"articles":articles}) |

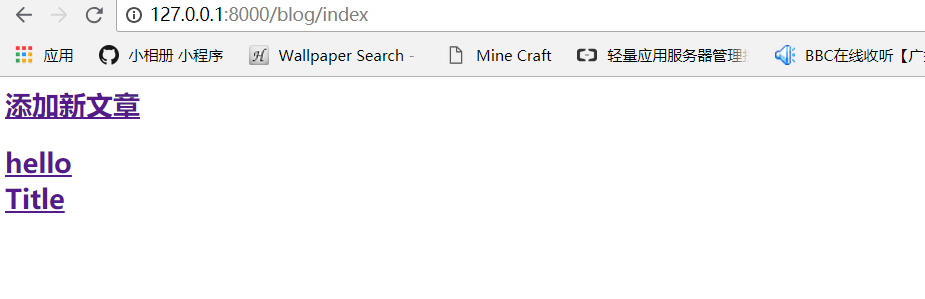
In the html, use a FOR loop to display all titles of article from database. The detail code show in table 9.

table 9

|  |
| --- |
| Index.html |
| <html>  <head>  <title>index</title>  </head>  <body>  <h2><a href="">添加新文章</a><h2>  {% for article in articles%}  <a href="">{{article.title}}</a>  </br>  {%endfor%}  </body>  </html> |

OK, Other documents remain unchanged. Type url:

<http://127.0.0.1:8000/blog/index> in browser will get this page. As show in figure 17.

 Figure 16

### Article page

1、Add a new function in views.py like table 10. The red is new.

table 10

|  |
| --- |
| Views.py |
| from django.shortcuts import render  from django.http import HttpResponse  from . import models  # Create your views here.  def index(request):  # return HttpResponse('hello xiyang')  articles = models.Article.objects.all()  return render(request,"blog/index.html",{"articles":articles})  def article(request,article\_id):  article = models.Article.objects.get(pk=article\_id)  return render(request,"blog/articlePage.html",{"article":article}) |

2、you need a new html file named articlePage.html.

table 11

|  |
| --- |
| articlePage.html |
| <html>  <head>  <title>Article Page</title>  </head>  <body>  <h1>{{article.title}}</h1>  </br>  <h3>{{article.content}}</h3>  </br></br>  <a href="">修改文章</a>  </body>  </html> |

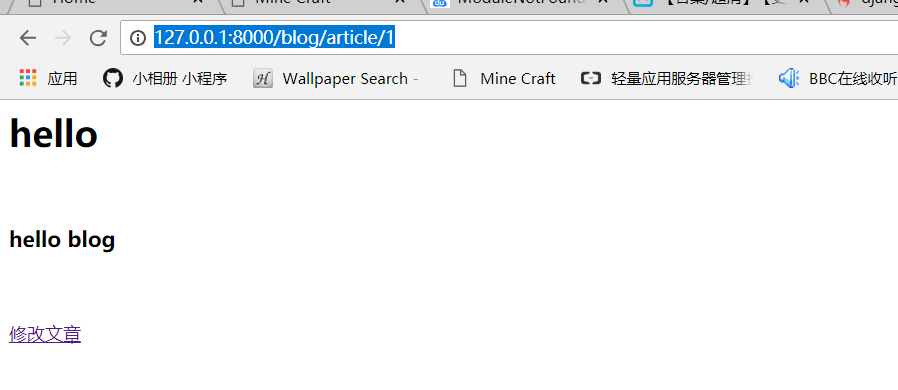
3、Add a url into blog/urls.py. The red is new.

table 12

|  |
| --- |
| Blog/Urls.py |
| from django.conf.urls import url  from . import views  #from blog.views import index  urlpatterns = [  url(r'^index',views.index),  url(r'^article/(?P<article\_id>[0-9]+)$',views.article),  ] |

Attention! The (…) in code:” *article/(?P<article\_id>[0-9]+)$*” is article id from views.py, the value must be the same with parameter in views.py.

Well input url:” <http://127.0.0.1:8000/blog/article/1>” in browser, it will return a page about content of first article as figure 17.

 Figure 17

### Edit page

1、Firstly we need a form to submit article. The form include title of article, content of article and a submit button. In html, we create a form. Attention! The code: *{% csrf\_token %}* is must when method is ‘post’ in form.

table 13

|  |
| --- |
| edit\_page.html |
| <html>  <head>  <title>Edit page</title>  </head>  <body>  <form action="{% url "blog:edit\_action"%}" method="post">  {% csrf\_token %}  <label>文章标题  <input type="text" name="title"/>  <label>  </br>  <label>文章内容  <input type="text" name="content"/>  <label>  </br>  <input type="submit">  </form>  </body>  </html> |

2、Need a render function in views.py to response the request. We edit a esit\_page function. And the function named edit\_action is used in form. As show in table 14.

table 14

|  |
| --- |
| views.py |
| from django.shortcuts import render  from django.http import HttpResponse  from . import models  # Create your views here.  def index(request):  # return HttpResponse('hello xiyang')  articles = models.Article.objects.all()  return render(request,"blog/index.html",{"articles":articles})  def article(request,article\_id):  article = models.Article.objects.get(pk=article\_id)  return render(request,"blog/articlePage.html",{"article":article})  def edit\_page(request):  return render(request,'blog/edit\_page.html')  def edit\_action(request):  title=request.POST.get('title','TITLE')  content = request.POST.get('content','CONTENT')  models.Article.objects.create(title=title,content=content)  articles = models.Article.objects.all()  return render(request,'blog/index.html',{'articles':articles})#After click submit button, plus #current title into index.html. It`s similar to index function. |

3、Add the url into urls.py.

table 15

|  |
| --- |
| blog/urls.py |
| from django.conf.urls import url  from . import views  #from blog.views import index  app\_name='blog' '''https://www.cnblogs.com/demonszz/p/8329846.html解决了这个问题'''  urlpatterns = [  url(r'^index',views.index),  url(r'^article/(?P<article\_id>[0-9]+)$',views.article,name='article\_page'),  url(r'^edit/$',views.edit\_page,name="edit\_page"),  url(r'^edit/action$',views.edit\_action,name='edit\_action'),  ] |

Well, open browser and type this url: <http://127.0.0.1:8000/blog/edit/>.



Figure 18

OK, It`s over. You can get code in github.

https://github.com/luoxi123/How-to-start-Django/