日期：2017.6.13

|  |  |
| --- | --- |
|  | django-admin startproject 项目名称  python manage.py runserver 运行项目  python manage.py runserver 8080指定运行端口号  python manage.py runserver 0:8000指定外网能访问  python manage.py startapp 新建appq名  URL定义（apps下的urls.py新建）：  from django.conf.urls import url  from . import views  urlpatterns = [  url(r'^$', views.index, name='index'),  ]  URL定义（project下的urls.py修改）  from django.conf.urls import include, url  from django.contrib import admin  urlpatterns = [  url(r'^polls/', include('polls.urls')),  url(r'^admin/', admin.site.urls),  ]  #在app目录下建立 migrations目录，记录下所有的对models.py的更改。  python manage.py makemigrations tousu  #查看数据库管理命令（不实际运行）  python manage.py sqlmigrate tousu 0001  #将改动后的migrations作用到数据库  python manage.py migrate  python manage.py createsuperuser  python manage.py runserver |
|  |

**Models and databases**

日期：2017.6.13

|  |  |
| --- | --- |
| essential 英[ɪˈsenʃl]  美[ɪˈsɛnʃəl]  adj. 必要的; 本质的; 基本的; 精华的;  n. 必需品; 基本要素; 必不可少的东西; | 定义Model  class Person(models.Model):  first\_name = models.CharField(verbose\_name=“投诉类型名称”,max\_length=30,default=”默认值”,help\_text=”说明”,unique=False) #verbose\_name可以省略  last\_name = models.CharField(max\_length=30)  定义字段  DurationField  upload = models.FileField(upload\_to='uploads/')  upload = models.FileField(upload\_to='uploads/%Y/%m/%d/')  def user\_directory\_path(instance, filename):  # file will be uploaded to MEDIA\_ROOT/user\_<id>/<filename>  return 'user\_{0}/{1}'.format(instance.user.id, filename)  class MyModel(models.Model):  upload = models.FileField(upload\_to=user\_directory\_path)  ImageField  ForeignKey 多对一的字段  class Car(models.Model):  manufacturer = models.ForeignKey('Manufacturer',  on\_delete=models.CASCADE,related\_name=’company\_site’)#related\_name的设置可以通过主表查找子表所有内容  # ...  class Manufacturer(models.Model):  # ...  pass  自连接models.ForeignKey('self',on\_delete=models.CASCADE).  ManyToManyField  OneToOneField |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  | 获取指定的对象  Blog.objects  b = Blog(name='Foo', tagline='Bar')  b.objects  all\_entries = Entry.objects.all() #获得所有的数据  Entry.objects.filter(pub\_date\_\_year=2006)  Entry.objects.all().filter(pub\_date\_\_year=2006)  Entry.objects.all()得到所有数据  Entry.objects.filter(pub\_date\_\_year=2006)得到指定条件的数据  Entry.objects.get(pk=1)得到一条数据  Entry.objects.exclude(条件)排除符合条件的内容  Entry.objects.all()[:5]前5条  Entry.objects.annotate(search=SearchVector('blog\_\_tagline', 'body\_text'),).filter(search='cheese')  在多个字段中查找cheese |
|  |

**URL映射**

日期：2017.6.13

|  |  |
| --- | --- |
|  | (?P<name>pattern)  url(r'^articles/(?P<year>[0-9]{4})/(?P<month>[0-9]{2})/$', views.month\_archive), |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |

日期：2017.6.13

|  |  |
| --- | --- |
|  |  |
|  |