1. pip Django==1.11.4
2. Django-admin startproject project
3. Tree . /F 查看目录层级
4. Settings.py 配置数据库 DataBases

Python3 安装 pymysql;

\_\_init\_\_.py 添加 import pymsql pymsql.install\_As\_DB()

1. 创建应用

python manage.py startapp myapp

1. 配置应用 settings INSTALLED\_APPS myapp
2. Python manage.py makemigrations myapp 迁移文件

Python manage.py migrate执行迁移

1. Python manage.py shell
2. Grade.students\_set.create()
3. Stu = Students()

Stu.name=”shuai” stu.save()

Garde=Gardes.objects.get(pk=1)

Grade.delete()

1. Python manage.py runserver ip:port 启动服务 port 默认8000，ip可以不写 轻量级 仅开发测试
2. python manage.py createsuperuser
3. 站点管理 <http://127.0.0.1:8000/admin> 汉化 zh-Hans Asia/Shanghai
4. 管理数据表 myapp 下的 admin.py

Admin.site.register(表名类,BanjiAdmin)

1. 自定义管理类

Class BanjiAdmin(admin.ModelAdmin)

Inlines=[StudengtsIfo]//创建班级时新加2个学生

列表页属性:

List\_display=[pk,name,age,sex,banNo\_id, gender(函数)]

List\_filter=[]过滤字段

Search\_fields=[]搜索字段

List\_per\_page=分页 5

添加/修改属性

Fields=[] 属性的先后顺序

Fieldsets=[(num,{“fields”:[name,age]}),(“base”,{“fields”：[sex,banNo\_id]})]分组

Def gender(self):

If self.gender:

Return “男”

1. 关联对象

Class StudentsIfo(admin.TabularInline):

Model =Students

Extra=2

1. Bool显示问题

Name.short\_description=”性别”

1. 执行动作位置

Actions\_on\_bottom=True

1. 实际注册 使用装饰器

@admin.register(Students)

1. MTV模式
2. Views views.py

From Django.http import HttpResponse

Def index(request):

Return HttpResponse(str)

1. 配置 url urls.py url(r’^’,include(myapp.urls))

在myapp内创建一个 urls.py文件

From Django.conf.urls import url

From . import views

Urlpatterns=[url(‘^$’,,views.index),url(r’^(\d+)/$’,views.detail)]

1. 模板

与project 同目录创建模板目录 templates

Settings.py templates属性 ‘DIRS’:[os.path.join(basedir,’template’)]

Html格式:

{%for grade in grades%}

<li>

<a href=”#”>{{grade.name}}</a>

</li>

{%endfor%}

取数据:

List=Grades.objects.all()

Return render(request,’myApp/grades.html’,{“grades”:list}) 传递数据到html

Grade.students\_set.all()获取班级下的所有学生