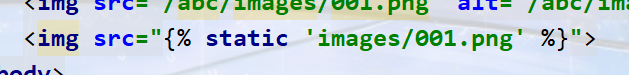


可以使用{% load static%}标签来自动引入静态文件

导入文件的时候，使用{% static ‘对应路径’ %}来引入静态文件

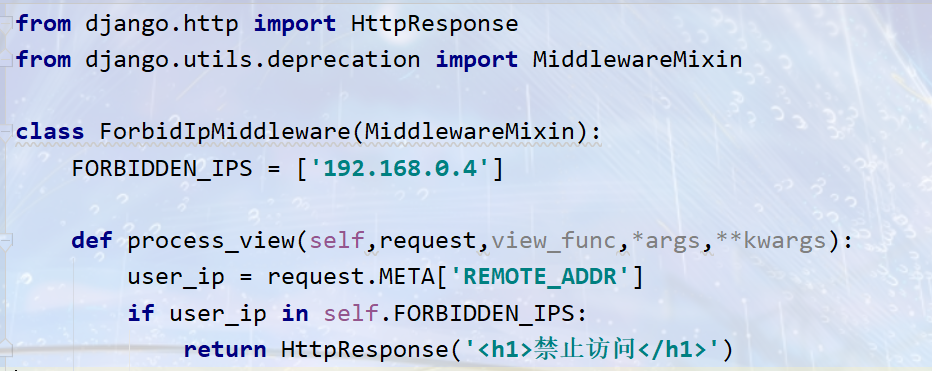


使用中间件（以下是django1.x版本的方法）

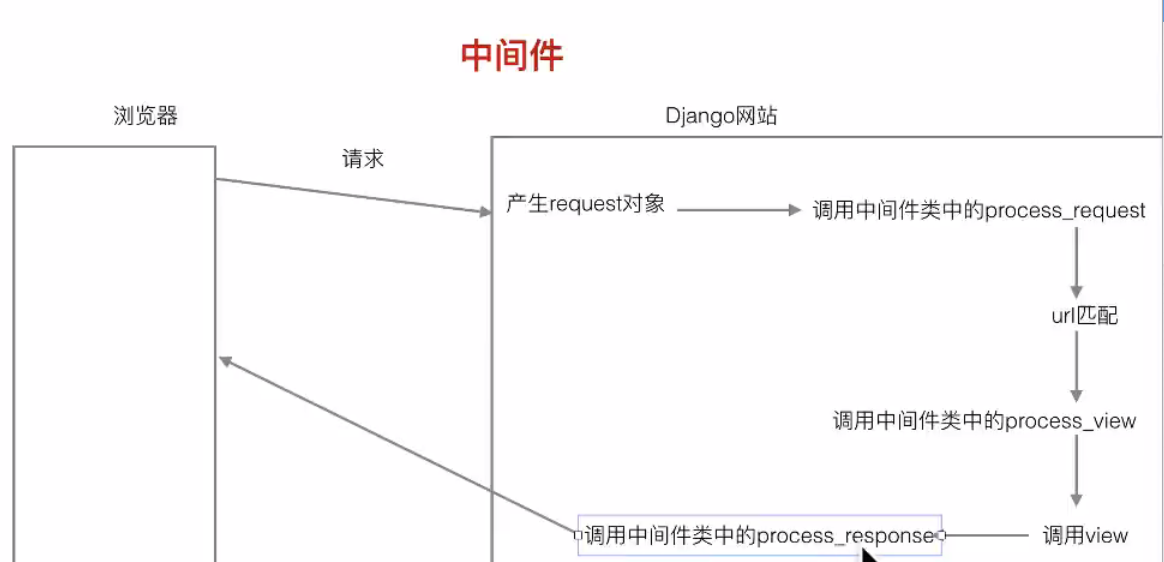
1. 在app路径下创建一个py文件，一般命名为middleware
2. 定义一个中间件类
3. 在中间件类里面定义中间件函数，注意，这个中间件函数只能是process\_view，这个是django预留的，这个函数一共有五个参数，第一个为self，第二个为request，第三个参数为view\_func，这个view\_func就是要调用的视图函数是谁，第四五个分别为\*args和\*\*kwargs
4. 在setting文件里面注册中间件类

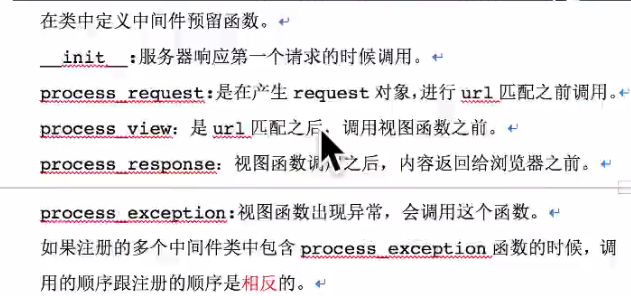
如果使用django3的话

1. 要导入**from** django.utils.deprecation **import** MiddlewareMixin
2. 然后创建的中间类要继承MiddlewareMixin，其余步骤与django1.x相同



中间件的操作步骤如下





对应的方法参数如下





注意：process\_response必须返回response，否则页面会报错







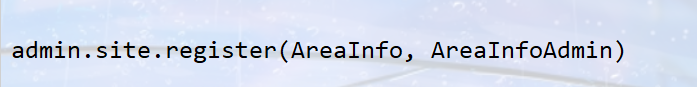
注意：process\_template\_response必须返回response，否则页面会报错, 执行 process\_template\_response 函数有一个前提条件，那就是视图函数返回的对象要有一个 render() 方法（或者表明该对象是一个 TemplateResponse 对象或等价方法）

中间件的执行顺序为process\_request->process\_view->

process\_template\_response->process\_exception->

process\_response

admin后台管理

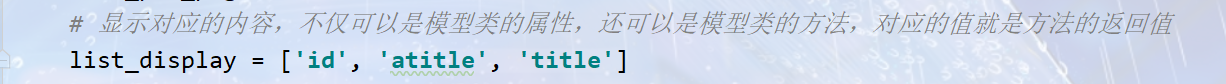


通过admin.site.register(模型类)来注册

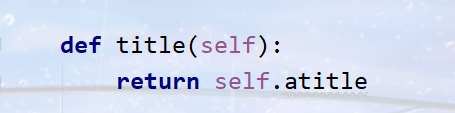


还可以创建一个对应的类来自定义对应的表，该类有以下几种属性



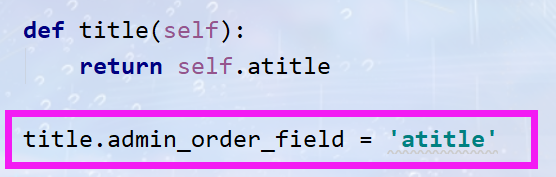


在对应的模型类里加上对应的方法，就可以显示对应方法的内容



如果按照对应的字段显示内容的话，在对应的字段顶部点击是可以排序的，但是如果现实的是方法所展示的内容的话，需要人为的设置

加上框住部分，就可以根据atitle来排序



还可以通过以下方法给对应的方法取别名以便更好看



在模型类中给定verbose\_name可以给定名称，这样在后台现实的就是这个名称

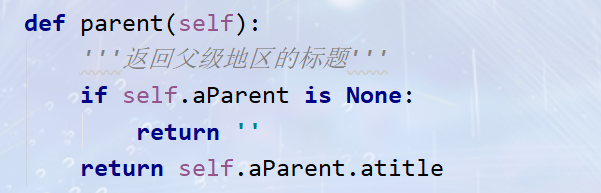




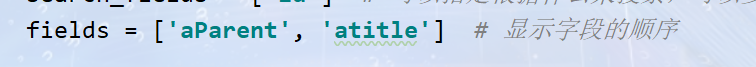




可以给对应的模型类以下方法，来返回父级地区的标题



如果不判空的话，省级是没有上级地区的，这样就会报错



注意：fields和fieldsets二者不可以同时使用

