

Django 开发实战 - 第四章



扫码试看/订阅

《 Django 快速开发实战 》视频课程

更美观的管理后台：替换Django admin的主题风格

- 安装 django-grappelli 风格
 - `pip install django-grappelli`
- settings.py 中配置
 - `INSTALLED_APPS = ('grappelli', 'django.contrib.admin',)`
- urls.py 中添加 URL 映射

```
urlpatterns = [  
  
    path('grappelli/', include('grappelli.urls')),  
  
    path('admin/', admin.site.urls), # admin site  
  
]
```

更美观的管理后台：替换Django admin的主题风格

127.0.0.1:8000/admin/interview/candidate/

匠果科技招聘管理系统

admin

View site

Home > Interview > 应聘者

应聘者

+ Add 应聘者

11 total

Filter

<input type="checkbox"/>	姓名	城市	本科学校	初试分	初试结果	3 ^	面试官	专业复试得分	专业复试结果	2 ^	二面面试官	HR复试综合等级	HR复试结果	1 ^	HR面试官	
<input type="checkbox"/>	李冀	成都	电子科技大学	-	-		<div> davidullua ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	刘力	上海	北京师范大学	-	-		<div> TomYang ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	王小小	深圳	华中科技大学	-	-		<div> TomYang ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	李爽	北京	南京大学	-	-		<div> davidullua ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	王五	上海	东南大学	-	-		<div> davidullua ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	李四	南京	电子科技大学	-	-		<div> TomYang ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	李小龙	深圳	东南大学	-	-		<div> zhaoganglv ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	陈欣欣	北京	北京师范大学	-	-		<div> ----- ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	刘倩	北京	北京航空航天	-	-		<div> ----- ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	张三	北京	北京航空航天	-	-		<div> ----- ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	
<input type="checkbox"/>	刘德华	上海	复旦大学	4.0	建议复试		<div> ----- ▾ → + </div>	-	-		<div> ----- ▾ → + </div>	-	-		-	

11 total

定制面试官权限

- 数据权限
 - 专业面试官仅能评估自己负责的环节
- 数据集权限 (QuerySet)
 - 专业面试官只能看到分到自己的候选人
- 功能权限（菜单/按钮）
 - 数据导出权限仅 HR 和超级管理员可用



定制面试官权限

- **数据权限：** 一面面试官仅填写一面反馈， 二面面试官可以填写二面反馈
`def get_fieldsets(self, request, obj=None):`
- **数据集权限：** 对于面试官， 获取自己是一面面试官或者二面面试官的候选人集合
`def get_queryset(self, request):`
- **功能/菜单权限**
 - 自定义权限： 在 Model 类的 Meta 中定义自定义的 permissions
 - 同步数据库： `./manage.py makemigrations && ./manage.py migrate`
 - 在 action 上限制权限： `export_model_as_csv.allowed_permissions = ('export',)`
 - 在 Admin 上检查权限： `def has_export_permission(self, request)`

发送通知：钉钉群消息集成

- 为什么不使用 Email/SMS 通知
 - 由于邮件、短信没有限制，可以给任何人发；网络上对于 API 调用有了各种限制
 - 阿里云封禁 25 端口
- 为什么使用钉钉群消息
 - 可以使用 Web Hook 直接发送，简单易用
 - 低成本
- 其他推荐消息方式
 - Slack 消息
 - 企业微信消息

测试钉钉群消息

- 安装钉钉聊天机器人: `pip install DingtalkChatbot`
- 测试群消息
 - `python ./manage.py shell --settings=settings.local`
 - `from interview import dingtalk`
 - `dingtalk.send("秋季招聘面试启动通知, 自 2020/09/01 开始秋季招聘")`

定制管理后台的操作按钮：通知面试官准备面试

- 定义 通知面试官的方法

```
def notify_interviewer(modeladmin, request, queryset)
```

- 注册到 modeladmin 中

```
actions = (export_model_as_csv, notify_interviewer, )
```

集成 registration, 允许候选人注册登陆

- 允许注册：安装 registration

```
pip install django-registration-redux
```

- 添加到 apps 中
- 同步数据库
- 添加登陆，退出链接到页面中

创建简历 Model （为投递简历做准备）

- 创建 Model
- 注册 Model 到 Admin 中，设置展示字段
- 同步数据库
- 授予管理权限到 HR

职位详情页：候选人简历投递

- 目标
 - 注册的用户可以提交简历
 - 简历跟当前用户关联
 - 能够追溯到谁投递的简历
- 步骤
 - 定义简历创建 View (继承自通用的CreateView)
 - 定义简历创建页面的表单模板
 - 关联“申请职位”按钮的点击事件到简历提交页
 - 进一步完善， 可以带参数跳转 && 关联登陆用户到简历

使用 Bootstrap 来定制页面样式

- 安装依赖包: `pip install django-bootstrap4`
- 添加到 apps 中: **bootstrap4**
- 模板里面使用 bootstrap 标签

简历初评，安排一面面试官

- 目标：打通简历投递与面试流程，让**简历实体** (Resume) 流转**到候选人实体** (Candidate)
- 添加一个数据操作菜单 “进入面试流程”
- 定义 enter_interview_process 方法

```
def enter_interview_process(modeladmin, request, queryset)
```

方法逻辑： 拷贝 Resume 对象中的属性 到 Candidate, 保存 Candidate, 消息提示成功保存

- 注册到 modeladmin 中

```
actions = (export_model_as_csv, notify_interviewer, enter_interview_process, )
```

定制列表字段，查看简历详情

- 添加 ResumeDetailView 的详情页视图，使用 Django 的通用视图，继承自 DetailView
- 添加 Detail 页模板：resume_detail.html
- 候选人列表页，对于每一行来自简历投递的数据，添加一个“查看简历”的链接：
 - 列表页，使用 函数名称 作为 list_display 中的字段
 - 定义一个函数，获取 简历详情页链接

参考资料

- 通用视图的使用:

https://developer.mozilla.org/zh-CN/docs/learn/Server-side/Django/Generic_views

<https://docs.djangoproject.com/en/3.1/topics/class-based-views/generic-display/>

<https://docs.djangoproject.com/en/3.1/ref/class-based-views/generic-editing>

- 表单的使用

<https://docs.djangoproject.com/en/3.1/topics/forms/modelforms/>

- Registration组件

<https://django-registration-redux.readthedocs.io/en/latest/simple-backend.html>



扫码试看/订阅

《 Django 快速开发实战 》视频课程