

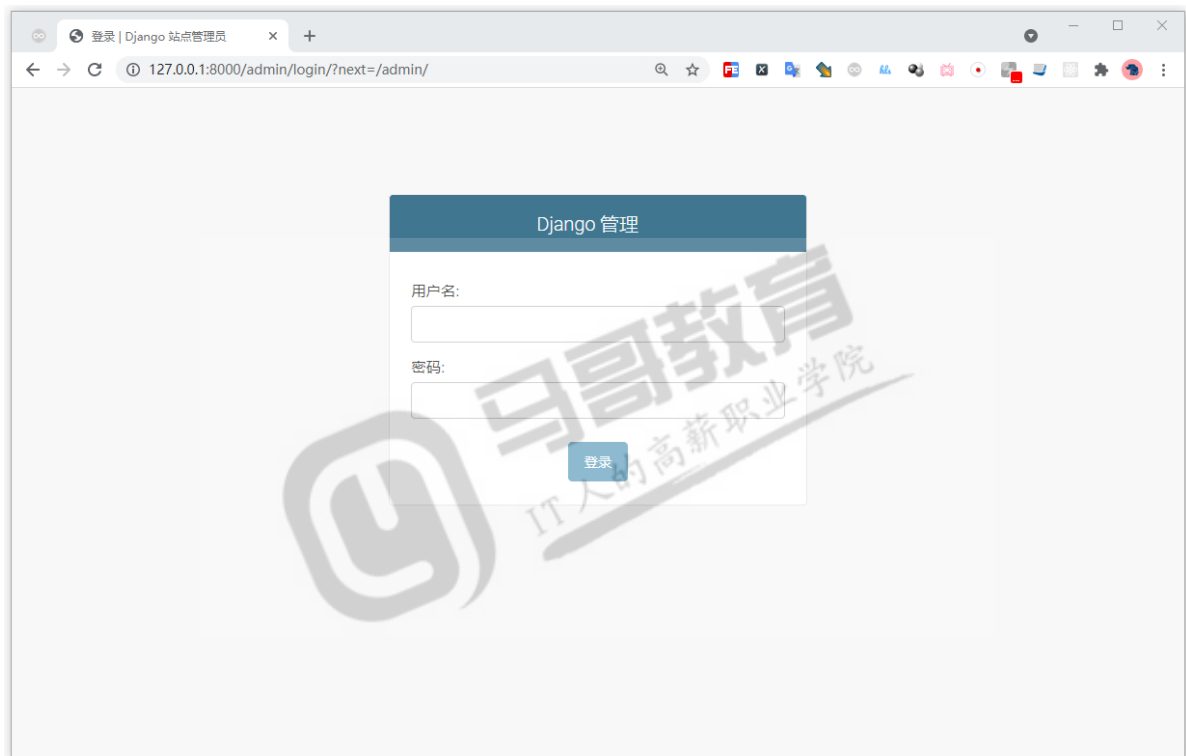
后台管理

站点管理

如果要使用Django，必须提供一个管理员用户，目前用户表auth_user表没有记录。

```
1 $ python manage.py createsuperuser
2
3 $ python manage.py runserver
```

浏览器中访问 <http://127.0.0.1:8000/admin/> 即可看到登录界面。



登录后，并没有看到自己建立的Model类对应的表。这是因为有2步必须做

1、注册应用

```
1 INSTALLED_APPS = [
2     'django.contrib.admin',
3     'django.contrib.auth',
4     'django.contrib.contenttypes',
5     'django.contrib.sessions',
6     'django.contrib.messages',
7     'django.contrib.staticfiles',
8     'employee',
9 ]
```

2、注册Model类

```
1 from django.contrib import admin
2 from .models import Employee, Department
3
4 admin.site.register(Employee)
5 admin.site.register(Department)
```

路由

按照WSGI原理代码，所有请求都交给一个app处理，如何做到不同请求对应不同的处理？

最简单的方式就是，进入app后，不同的URL做不同的处理。也就是，建立URL和处理函数之间的映射。

路由：根据访问的路径Path不同调度不同的处理函数。可以对路径Path采用模式匹配。

Django 3.x使用path、re_path定义，这和2.x略有区别。

主路由

path参考 <https://docs.djangoproject.com/en/3.2/topics/http/urls/#example>

主目录中的urls.py是路径匹配的入口，它定义在 settings.py 中 ROOT_URLCONF = 'salary.urls'。

约定，称主目录中的路由配置文件urls.py为主路由 或 一级路由

```
1 from django.contrib import admin
2 from django.urls import path
3 from django.http import HttpResponse
4
5 # 为了测试方便，写了这个path_test视图函数，以后不要写在这里
6 def path_test(request, *args, **kwargs):
7     print('=' * 30)
8     print(request)
9     print('args', [(type(x), x) for x in args])
10    print('kwargs', [(type(k), k, type(v), v) for k, v in kwargs.items()])
11    print('=' * 30)
12    return HttpResponse()
13
14
15 # urlpatterns是一个列表，匹配从上往下，匹配上了就不会继续向下匹配
16 urlpatterns = [
17     path('admin/', admin.site.urls),
18     path('test/<course>/<year>/', path_test), # 使用缺省转换器
19     path('test/<str:course>/<int:year>/', path_test), # 指定转换器类型
20 ]
```

路径转换器

- str：匹配任一非空字符串，但不能匹配路径分隔符 /，这是缺省转换器
- int：匹配0或任一正整数，并转换为int

URL	path模式	参数解析
/test/python/2008 /test/python/2008/	test/<course>/<year>/	默认转换器类型为str ['course', <class 'str'>, 'python'] ['year', <class 'str'>, '2008']
/test/python/2008 /test/python/2008/	test/<str:course>/<int:year>/	['course', <class 'str'>, 'python'] ['year', <class 'int'>, 2008]

从测试结果看，如果和匹配上了，采用的是**关键字传参**。

URL	path模式	
/test/python/2008	test/<course>/<year>	匹配
/test/python/2008/	test/<course>/<year>	失败
/test/python/2008	test/<course>/<year>/	301跳转到/test/python/2008/
/test/python/2008/	test/<course>/<year>/	匹配

由于之后所有的链接都是程序中提取写好的，所以，一定可以保证程序员写的URL是正确的。建议，定义path模式时，使用 `xxx/`，以斜杠 / 结尾。在Django官网例子中就是这样定义的。

re_path支持正则表达式，参考 <https://docs.djangoproject.com/en/3.2/topics/http/urls/#using-regular-expressions>

二级路由

如果所有路由都定义在主路由文件中，太乱太臃肿，解决的办法就是二级路由，即路由到应用的路有文件。但是脚手架建立的应用文件中缺少urls.py，手动在应用目录建立一个。

1、主路由文件

```

1 from django.contrib import admin
2 from django.urls import path, include
3
4
5 urlpatterns = [
6     path('admin/', admin.site.urls),
7     path('emp/', include('employee.urls')), # 指向应用路由文件
8 ]

```

2、应用路由文件

在employee目录下新建urls.py。

```
1 from django.urls import path
2 from .views import test_index
3
4 urlpatterns = [
5     path('', test_index), # 二级路由包含前缀emp/, 对应URL是/emp/
6 ]
```

在employee/view.py中增加index视图函数

```
1 from django.http import HttpRequest, HttpResponse
2
3 def test_index(request:HttpRequest):
4     return HttpResponse(b'magedu.com')
```

访问 <http://127.0.0.1:8000/emp/> 即可。

