



链滴

使用 Django 框架快速开发

作者: [darkleo](#)

原文链接: <https://ld246.com/article/1540370611000>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

整体流程: 基础准备 -> 数据库 -> 服务 -> 页面

注:

1. 不做深入介绍,只基于使用, 需自行进行深入研究。
2. 基于python3.7,django2.1,sqlite3
3. 参考文档: <https://docs.djangoproject.com/zh-hans/2.1/>

一, 基础准备

1. 环境搭建, 请百度
2. 建立项目

```
python -m django startproject django-demo
```

3. 建立完项目后, 进入其目录, 通过 `manage.py`来建立模块

```
cd django-demo  
python manage.py startapp person_module
```

4. 将项目导入到 PyCharm中。

参考链接: <https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial01/>

二, 数据库

1. 设置数据库链接

修改配置文件`settings.py`。找到配置选项`DATABASES`,将其内容修改为自己使用的数据库, 默认为sqlite3,以下配置为mysql配置信息

```
DATABASES = { 'default': { 'ENGINE': 'django.db.backends.d', 'NAME': 'mydatabase', 'USER': 'm  
databaseuser', 'PASSWORD': 'mypassword', 'HOST': '127.0.0.1', 'PORT': '5432', } }
```

注: 如果使用mysql数据库里, 需要进行如下操作:

- 1)安装pymysql模块 `pip3 install pymysql`
- 2)添加

```
import pymysql  
pymysql.install_as_MySQLdb()
```

到模块下的`__init__.py`文件中

2. 定义表结构

在`models.py`中添加表结构类对象.例如下内容:

```
...  
class Person(models.Model):  
    id = models.IntegerField(primary_key=True)  
    name = models.CharField(max_length=64)  
    age = models.IntegerField(default=-1)  
    sex = models.IntegerField(default=0)  
    hobby = models.CharField(max_length=255)
```

```
height = models.IntegerField(default=-1)
weight = models.IntegerField(default=-1)
married = models.IntegerField(default=-1)
status = models.IntegerField(default=0)
```

...

3. 使用django自动创建表

修改程序配置文件`settings.py`, 找到`INSTALLED_APPS`配置选项。添加如下内容。

```
'person_module.apps.PersonModuleConfig'
```

4. 生成迁移文件

```
python manage.py makemigrations person_module
```

...

5. 执行迁移

```
python manage.py migrate person_module
```

...

6. 查看表是否已经创建成功。

注：如果想重新创建表时，需要将表`django_migrations`表中相关数据删除，否则不会再重新创建的。

参考链接：<https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial02/>

三，创建服务

1. 在模块下的 `views.py`文件中添加方法，然后将其方法发布为一个服务。添加的内容如下：

...

```
from person_module.models import Person
```

```
def list_person(request):
    person_list = Person.objects.all() #django提供的方法
    return render(request, "person/list_person.html", {"person_list": person_list})
```

...

2. 配置

1)在`person_module`模块下新建立`urls.py`文件，文件名可以自己定义。

2)在文件中添加此`person_module`模块中的服务。

```
from django.urls import path
from person_module import views
```

```
app_name = 'person' #在页面模板中使用
```

```
urlpatterns = [  
    path('list_person', views.list_person, name='list_person'),  
]
```

3)配置到全局的服务发布文件中`urls.py`文件中。添加如下内容:

```
from django.contrib import admin  
from django.urls import path  
from django.urls import include #添加的内容  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('site/', include('website_one.urls')), #添加的内容  
]
```

3. 服务启动,默认监听端口为8000

```
python manage.py runserver
```

访问地址为: http://127.0.0.1/person/list_person 但会报错, 因为没有创建模板页面.

参考链接:

服务配置: <https://docs.djangoproject.com/zh-hans/2.1/intro/tutorial01/>

数据库查询: <https://docs.djangoproject.com/zh-hans/2.1/topics/db/queries/>

四, 创建页面

1. 在模块下建立文件夹名为 `templates`

2. 在 `templates`目录下建立`person`文件夹, 然后在建立`list_person.html`文件。

3. 编写内容, 使用了django的模板语法列表内容如下:

```
<table>  
    {% for person in person_list %}  
        <tr>  
            <td>{{ person.name }}</td>  
            <td>{{ person.age }}</td>  
        </tr>  
    {% endfor %}  
</table>
```

4. 在数据库中手动添加几条数据, 启动服务, 进行测试。

5. 静态文件的引入

1)在模块下建立文件夹`statics`

2)将需要引入的文件导入到`statics`目录下

3)页面上的写法如下:

```
<link rel="stylesheet" href="{% static 'bootstrap/css/bootstrap.min.css' %}">
```

参考链接:

django模板: <https://docs.djangoproject.com/zh-hans/2.1/topics/templates/>

静态资源引用: <https://docs.djangoproject.com/zh-hans/2.1/howto/static-files/>