1.django安装与配置

一.安装django所依赖的系统的RPM包

```
yum -y groupinstall "Development Tools"
yum -y install python-devel python3-devel sqlite-devel libffi-devel zlib zlib-devel openssl openssl-devel
pip install pysqlite
```

二.第一个django项目

2.1 django安装

2.1.1 pip方式安装 pip install django==1.9.13

2.1.2 源码方式安装

1） 下载源码包

2） 解压源码包

```
[root@k8s-master1 Django-1.9.13]# tar -zxvf Django-1.9.13.tar.gz
```

3） 安装

```
(proj2) [root@k8s-master1 src]# cd Django-1.9.13
(proj2) [root@k8s-master1 Django-1.9.13]# python setup.py  install
```

2.2.1 创建web项目

```
(proj2) [root@k8s-master1 Django-1.9.13]# django-admin startproject web
```

2.2.2 创建app

```
(proj2) [root@k8s-master1 Django-1.9.13]#  django-admin startapp blog
```

2.2.3 项目目录结构及其说明

```
(proj2) [root@k8s-master1 djclass]# tree
.
├── blog              ##APP名
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── migrations
│   │   └── __init__.py
│   ├── models.py     #定义数据库模型，与数据库进行交互
│   ├── tests.py
│   └── views.py      #后台服务端代码，定义视图函数与url形成映射关系
├── djclass           #项目名
│   ├── __init__.py
│   ├── settings.py #项目的配置文件
│   ├── urls.py       #定义URL与Python代码的映射关系，映射关系称为网站路由
│   └── wsgi.py       #一个web服务器兼容的入口以便运行这个项目
└── manage.py         #与项目进行交互的命令工具集
```

2.3 启动项目

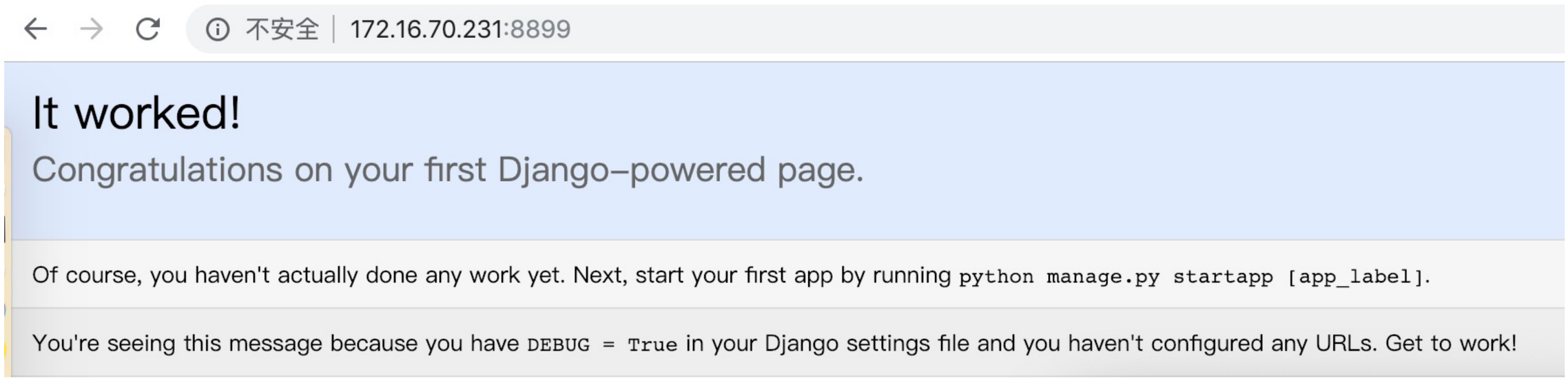
2.3.1 修改配置文件，django配置文件在项目目录中的settings.py文件中定义.

修改 settings.py中的配置 ALLOWED_HOSTS = [] --> ALLOWED_HOSTS = ['*']

2.3.2 启动项目

```
(proj2) [root@k8s-master1 djclass]# python manage.py  runserver 0.0.0.0:8899
```

2.3.3 验证：



三. django配置文件详解

Django配置文件位于settings.py文件中，配置文件中的参数及其含义如下：

1. 指定项目的所在目录

```
import os
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)  ##项目的根目录
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

说明：

__file__ 代表当前文件所在的目录

os.path.dirname() 获取该指定路径的上一层目录

2.

```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '&9cfk46#0p@uj!zn7+v+p$6qk1a4iwk)ud^*h@@8dj6h3!d30a'  ##保障请求的安全性
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True   ##开启网站的Debug
```

3. 设置允许哪些域名可以访问该网站

```
ALLOWED_HOSTS = ['192.168.12.21', 'localhost', '172.16.70.130', '127.0.0.1']
```

4.App注册， 用来记录加载的应用

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'home',
    'blog',
    'person',
]
```

5. 中间件， 自带工具集

```
MIDDLEWARE_CLASSES = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
#    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.auth.middleware.SessionAuthenticationMiddleware',
```

```
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'web.m1.middle1',
]

6.  上下文渲染器
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, "templates")],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
                'web.context_processor.settings',
                'web.context_processor.ipAddr',
            ],
        },
    },
]


7.  数据库引擎,  可以用来连接sqlite 与 mysql

# Database
# https://docs.djangoproject.com/en/1.9/ref/settings/#databases
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}


8.其他
STATIC_URL = '/static/'  ##静态文件的url地址
STATIC_ROOT = os.path.join(BASE_DIR, 'static')  ##静态文件在本地存放的位置

LANGUAGE_CODE = 'en-us'  ##语言
TIME_ZONE = 'UTC' ##时区
ROOT_URLCONF = 'web.urls' ##url的根配置文件 指向 urls.py

WSGI_APPLICATION = 'web.wsgi.application'  ##默认配置不需要管
##密码认证相关 不需要管
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

USE_I18N = True
USE_L10N = True
USE_TZ = True  ##国际化相关配置

Django 支持国际化，多语言。Django的国际化是默认开启的，如果您不需要国际化支持，那么您可以在您的设置文件中设置 USE_I18N = False，那么Django会进行一些优化，不加载国际化支持机制。
NOTE: 18表示Internationalization这个单词首字母I和结尾字母N之间的字母有18个。I18N就是Internationalization（国际化）的意思。
```

In [ ]: