

一、简介

uWSGI是一个高性能的、支持多种语言和框架的应用服务器，用于托管和运行 Web 应用程序。uWSGI服务器旨在提供高效的 Web 服务，支持大规模的并发连接和高吞吐量，是构建现代 Web 应用的重要组成部分之一

uwsgi协议

是uWSGI服务器自有的一个协议，用于定义传输消息的类型，可以与Nginx等代理服务器进行通信

二、安装

```
sudo pip install uwsgi
```

三、测试

```
#-*- encoding:utf-8 -*-
def application(environ, start_response):
    start_response('200 OK', [('Content-Type', 'text/html')])
    return [b"Hello World!"]
```

命令行输入 `uwsgi --http :8000 --wsgi-file test.py`

四、配置文件

在生产环境中，一般不回使用命令来启动 Python Web 项目，而是通过编辑好的 uWSGI配置文件 `uwsgi.ini` 来启动

```
[uwsgi] # 声明是uwsgi的配置文件

# 配置服务器监听的ip和端口 0.0.0.0代表监听所有的ip
# 如果让uWSGI作为Nginx的支持服务器的话，设置socket
#socket = 0.0.0.0:8000
# 如果要让uWSGI 作为单独的web-server,用http
http = 0.0.0.0:8080

# 配置项目的目录（项目的根目录）
chdir = /home/weihong/backendFrame/Day_818/uWSGI

# 配置模块的入口（运行哪个文件）
wsgi-file = app.py

# 开启master，会多开一个管理进程，管理其他的服务进程
master = true

# 延时加载，某些功能不立即加载，而是等到接收请求，再加载某些模块
# 可以防止加载时间太长
lazy-apps = true

# 服务器开启的进程数量、处理器个数（一般最大写CPU的核心数量）
processes = 2
```

```
# 服务器进程开启的线程数量
threads = 4

# 以守护进程发布提供服务, 输出信息将打印到log中
daemonize = %(chdir)/log/uwsgi.log/uwsgi.log

# 退出的时候清空环境变量
vacuum = true

# 进程id, 指定进程号存到那个文件中
pidfile = uwsgi.pid

# 设置用于uwsgi包解析时的缓存区大小, 默认为4k
buffer-size = 32768

# 修改python代码时, 自动重启uWSGI服务器
py-autoreload = 1

# 监控器, 由master管理, 会杀死卡住超过指定秒数的进程 (这里设置为30秒)
harakiri = 30

#
stats = 127.0.0.1:8081
```

在命令行使用 改命令即可启动uWSGI服务器

```
uwsgi --ini uwsgi.ini
```

查看日志文件内容, 看看是否启动成功

```
sudo tail -f log/uwsgi.log/uwsgi.log
```

五、常用命令

启动uWSGI服务器

```
uwsgi --ini uwsgi.ini
```

重启uWSGI服务器, 通过配置文件可以将进程id写入uwsgi.pid文件中

```
uwsgi --reload uwsgi.pid
```

查看日志文件内容

```
sudo tail -f log/uwsgi.log/uwsgi.log
```

查看所有 uWSGI进程

```
ps aux | grep uwsgi
```

停止所有的uWSGI 进程

```
sudo pkill -f uwsgi -9
```