



链滴

# windows 生产服务器下部署 Flask

作者: [barryzpc](#)

原文链接: <https://ld246.com/article/1595578907045>

来源网站: 链滴

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



## 场景描述

很多python工程师，都会用**flask** 来开发、部署Web 应用。但也有很多工程师就直接使用 `app.run()` 生产环境上启动服务。

那可能会发现 Flask 用 `app.run()`这种方式启动，有异常特别容易退出，而且性能很弱。

生产环境下，python常见的web部署搭配是 `nginx+gunicorn`。

但是这种搭配只适合在Linux环境下。关于这种搭配的配置方法网上有很多。反而在windows环境下使用什么搭配来部署，网上说的比不多。而我们的项目由于某种特殊的原因就必须搭建在windows环境下，所以使用的部署方案是：`Flask + Tornado+nginx`。

## 解决方案示例

### flask服务文件 flask\_app.py

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
```

### 安装Tornado

```
pip install tornado
```

## flask\_app.py目录下编写Tornado的 server.py文件

```
from tornado.httpserver import HTTPServer
from tornado.wsgi import WSGIContainer
from flask_app import app
from tornado.ioloop import IOLoop

s = HTTPServer(WSGIContainer(app))
s.listen(8080) # 监听 8080 端口
IOLoop.current().start()
```

### 启动服务

当前目录下执行 `python server.py` 浏览器中访问 `http://ip:8080` 即可

```
python server.py
```

### 配置nginx反向代理

配置nginx反向代理，这样在浏览器中直接访问[www.test.com](http://www.test.com) 即可

windows下nginx的安装和使用还是很简单的，网上教程很多，这里就不多说了

```
http {
    server {
        listen    80;
        server_name www.test.com;
        charset   utf-8;

        location / {
            root   html;
            index index.html index.htm;
            proxy_pass http://ip:8080;
        }
    }
}
```