

基于 Tornado 的 HTTP 服务器简介及代码组织框架

Tornado 是一个 Python Web 框架和异步网络库，最初是在 FriendFeed 开发的。通过使用非阻塞网络I/O，Tornado 可以扩展到数以万计的开放连接，但却在创建和编写时足够的轻量级。

Tornado 的特点

Tornado 和现在的很多主流 Web 服务器框架（包括大多数 Python 框架）有着明显的区别：它是非阻塞式异步服务器。大多数社交网络应用都会展示实时更新来提醒新消息、状态变化以及用户通知，客户端需要保持一个打开的连接来等待服务器端的任何响应。这些长连接或推送请求使得非异步服务器线程池迅速饱和。一旦线程池的资源耗尽，服务器将不能再响应新的请求。异步服务器在这一场景中的应用则不同，当负载增加时，诸如 Tornado 这样的服务器，会把当前请求正在等待来自其他资源的数据，加以控制并挂起请求，以满足新的请求。这也是 Tornado 在高并发、高效率的 Web 服务器应用很广的原因之一。

Tornado 入门

编写 Hello, world

上面我们已介绍了 Tornado 的强大，现在我们从一个简单的 Hello World 开始。在服务器上任意目录下（如 /data），创建 hello.py 文件，输入如下代码：

```
import tornado.ioloop
import tornado.web

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello, world")

application = tornado.web.Application([
    (r"/", MainHandler),
])

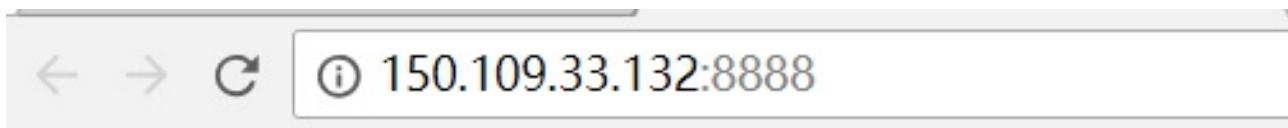
if __name__ == "__main__":
    application.listen(8888)
    tornado.ioloop.IOLoop.current().start()
```

编写一个 Tornado 应用中最多的工作是定义类继承 Tornado 的 RequestHandler 类。在这个例子中，我们创建了一个简单的应用，Tornado 监听给定的端口 8888，并在根目录（"/"）响应请求，响应的处理方法为继承了 RequestHandler 的 MainHandler 类。在 MainHandler 中返回 Hello, world。

测试代码

```
[root@VM_0_8_centos data]#
[root@VM_0_8_centos data]#
[root@VM_0_8_centos data]# python3 hello.py
```

在浏览器上打开 <http://150.109.33.132:8888>，测试结果如下：



Hello, world

注：服务器上需要放开 8888 端口，如果是公有云云主机，注意安全组配置是否已放开。

至此，我们已完成基于 Tornado 服务器的 Hello, world。下面来简单介绍一下 Tornado 的整体框架。

Tornado 框架

Tornado 大体上可以被分为 4 个主要的部分：

1. Web 框架 (包括创建 Web 应用的 RequestHandler 类，还有很多其他支持的类)；
2. HTTP 的客户端和服务端实现 (HTTPServer and AsyncHTTPClient)；
3. 异步网络库 (IOLoop and IOStream)，为 HTTP 组件提供构建模块，也可以用来实现其他协议；
4. 协程库 (tornado.gen) 允许异步代码写得更直接而不用链式回调的方式。

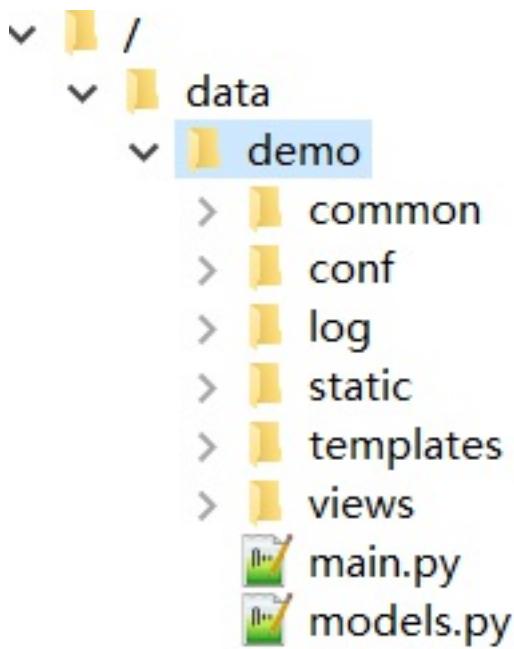
这里只做简单的了解，如需深入了解 Tornado，建议读者通读学习 [Tornado 官方文档 \(http://tornado-zh.readthedocs.io/zh/latest/guide.html\)](http://tornado-zh.readthedocs.io/zh/latest/guide.html)。

代码组织框架

在认识了 Tornado 之后，我们将正式进入本小册核心的学习。首先 Tornado 的学习必定是从 Hello, world 开始，并逐步按照个人编程习惯和组织习惯完善整个框架。

代码的组织框架因人而异，作为入门小册，这里提供一种简化的组织框架思路并贯穿整个小册，读者在熟练应用后，可采用自身的风格。

在某个目录下，创建本次的工程文件，如 demo，并依次创建如下文件：



目录及文件说明

common：存放公共类和方法

conf：存放配置文件

log：存放相关日志

static：存放静态文件，如样式（CSS）、脚本（js）、图片等

templates：公用模板目录，主要存放 HTML 文件

views：视图函数，业务逻辑代码目录

main.py：Tornado 主程序入口

models.py：数据库表结构定义

小结

本小节简单介绍了 Tornado HTTP 服务器及本小册中使用的代码组织框架，从下一小节开始，正式进入代码编写讲解。