

# 技术报告

14331077 关伟杰

## Tornado

Tornado 是使用 Python 编写的一个强大的、可扩展的 Web 服务器。它在处理 严峻的网络流量时表现得足够强健，但却在创建和编写时有着足够的轻量级， 并能够被用在大量的应用和工具中。同事 Tornado 在设计之初就考虑到了性能因素，旨在解决 C10K 问题，这样的设计使得其成为一个拥有非常高性能的框架。

此外，它还拥有处理安全性、用户验证、社交网络以及与外部服务（如数据库 和网站 API）进行异步交互的工具。tornado 常用的模块包括 `httpserver`、`options`、`ioloop` 以及 `web` 等。`options` 用于对服务器启动进行设置，如端口等参数，在启动服务器时一并在 命令行中输入，通过调用 `parse_command_line()` 方法进行解析，也可以在代码中 直接设置。通过 `web.Application()` 可以初始化应用对象，参数为元祖列表，每一个元祖 包含两个元素。第一个元素为 url，支持正则表达式；第二个元素为对应的 handler 的类。handler 类通过继承 `web.RequestHandler` 实现，内部可以重载 `get`、`post` 等方法，对应对不同 http 方法的请求进行处理。`get`、`post` 方法除了 `self` 参数以 外，之后的参数均为 url 中正则表达式里()包含的部分，也可以通过 `get_argument` 获取 url 中的参数。如同 `nodejs` 一样，tornado 框架同样提供了异步请求的功能。假如应用中需 要服务器向其他网站请求数据，使用原有的方式在多请求的情况下显得相当的耗 时。使用 `httpclient.AsyncHTTPClient().fetch()` 方法，可以设置请求 url 与回调函数， 在回调函数中对响应数据进行处理。tornado 还有一大特点就是可以轻松处理长轮询。对于一些数据会频繁变动的对象，大量的短而频繁的 http 请求相当的低效，而长轮询则是维护一个建立 的连接，服务器仅当新的数据被“推送”时对数据进行处理，省去了连接的建立 与关闭过程。实现长轮询，仅需在 handler 里的函数定义前加上装饰器 “`@tornado.web.asynchronous`”。