

服务器端组件框架的选择与介绍

本小册的目标在于搭建一款简洁、高效，并能快速部署及上手的 App 后端服务器。

在编程语言的选择上，我们选择了 Python 3.6.2，这也是本小册对读者的基本要求，从 HTTP 服务器，到数据库操作及逻辑分析处理，将全部使用 Python 语言。

前两节的介绍中讲到，我们使用 HTTP 协议来定义 App 客户端与服务器端之间的通信。我们不可能自己写一个 HTTP 服务器来响应客户端的 HTTP 请求，所以在 App 服务器端开发中，我们工作的重中之重，就是要寻找到一款合适的、支持高并发、易扩展并真正能阐释代码简洁美的框架。在调研了众多的 HTTP 服务器框架之后（如 Django、Pyramid），笔者最终选择了 Tornado。从多年前第一次遇到 Tornado 开始，笔者就爱上了它：简洁高效，易扩展，高并发。著名的知乎也是建立在 Tornado 之上。好好利用这把利剑，必定能在平时的生活工作中，助你更上一层楼。

除了编程语言和 HTTP 服务器框架外，我们还要选择操作系统及数据库。本小册中将使用 CentOS 7.2 x64 作为操作系统，MySQL 作为数据库，操作数据库使用 ORM（Object Relational Mapping）的方式。服务器端的整体框架如下图所示。

CentOS

CentOS 是大名鼎鼎的 Red Hat 的开源版本，由 Red Hat 公司维护测试，并在 Linux 内核稳定分支上进行开发，系统相对稳定。Red Hat 一早就在中国布局，市面上书籍众多，网上资料丰富，很多公司，第一版优先支持的版本也是 CentOS，这也是我们选择 CentOS

的原因，当出现疑难杂症时，能第一时间找到解决方案。在本次服务器端开发中，我们并不需要精通 CentOS，只需要会简单地使用 Linux 的命令即可，如 `yum install <module name>`。

MySQL

MySQL 是最流行的关系型数据库管理系统，在 Web 应用方面是最好的关系型数据库管理系统软件之一，也是最早一批被国内用户熟知的数据库软件之一。同样，当出现疑难杂症时，丰富的图书及网络资源能帮助我们尽快找到解决方案。

SQLAlchemy

在操作数据（如 MySQL）的过程中，我们可以使用原生的 MySQL 语句（如 `insert`、`update`、`delete`），也可以使用 ORM（Object Relational Mapping）的方式。简单来说，可以使用第三方软件来操作数据库，使用第三方的好处是很多底层 MySQL 的命令被封装成简单的 API 暴露给用户，并提供强大的整合功能。当然坏处也有，如相对于原生命令效率低些，学习成本高些。而选择 SQLAlchemy 的原因是其使用 Pythonic 的代码风格，在本小册中不会给读者增加太多学习成本，另外，SQLAlchemy 全面的 API 参考文档也是我们选择它的原因之一。

Tornado

Tornado 作为我们选用的 HTTP 服务器框架，在后续的章节中，我们将作详尽的诠释。

小结

本小节简单介绍了整个服务器端的组件框架，从下一节开始，我们将进入环境搭建并开始真正的代码编写。