

App 与后端服务器通信方法简介

在开始之前

本小册的目标是希望大家在学习完成之后，能少掉笔者当初入门服务器端开发的烦恼。通过本小册，不仅能快速搭建起自己或公司的后端服务器，也能进一步优化，扩展，从而满足业务增长对服务器端的需要。

你将学到什么？

本小册将从基本的 App 与服务器端的通信讲起，涵盖数据的交互、图片的上传加载、H5 页面的请求加载。在基本通信场景功能的基础上，笔者将带领大家一起学习反向代理服务器 Nginx，并为保障业务通信的安全性及反 HTTP 劫持，学习使用 HTTPS。

最后，笔者也将以多年的后端服务器开发经验，就大型服务器端架构设计的演化路径和读者一起探讨。

总的来说，读者将从本小册学习到如下课题：

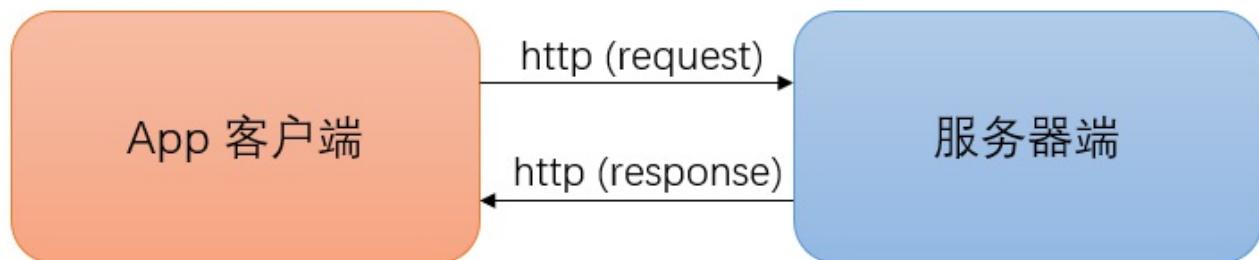
1. 腾讯云上配置开发环境
2. 基于 Tornado 的 HTTP 服务器框架
3. App 客户端/服务器端之间的数据通信
4. 服务器端对数据库 MySQL 的操作
5. 基于 Nginx 的反向代理及基于 HTTPS 的数据加密
6. 完成一款大型服务器的进阶方案和演进路线

本小册所使用的开发语言为 Python 3.6.2，各模块的版本信息将在后续章节中详细介绍。下面进入我们的正式学习阶段。

通信方法简介

很多朋友会问，App 客户端如何与服务器端进行通信呢？格式是什么？场景有那些？本小节将着重解答这些问题。

当前手机 App 客户端与服务器端通信，通常有两种模式：一种是短连接，一般通过 HTTP 进行通信；第二种是长连接，一般为 socket，长连接需要手机客户端与服务器端一直保持连接状态，服务器端压力较大，一般在游戏，服务器端主动向客户端推送服务信息时应用较为广泛。作为入门指南，本小册只讨论短连接 HTTP 的通信场景。而在短连接 HTTP 通信中，数据交互可以自定义，也可使用业界通用方法，即客户端和服务器端的数据交互采用 JSON 格式。本小册将使用业界通用方法，使用 JSON 的原因是 JSON 作为一种通用数据交换格式，被众多计算机语言支持，且开销小，省流量。



下面重点介绍一下 HTTP 和 JSON。

HTTP 与 JSON 简介

HTTP 协议

超文本传输协议（HTTP）是一个客户端和服务器端请求和应答的标准（TCP）。通过使用 App 客户端、Web 浏览器或者其他工具，客户端发起一个到服务器上指定端口（默认端口为 80）的 HTTP 请

求。HTTP 有多种请求方式，而 App 客户端与服务器端的请求应答中，最常用的就是 GET 和 POST，本小册只会介绍这两种方式。如对其他方法感兴趣，可自行谷歌。

GET 方法

当发送一个 HTTP 请求时，查询字符串（名称/值对）是在 GET 请求的 URL 中发送的，即参数暴露在请求 URL 中，如：

`http://www.demo.com/test?name1=value1&name2=value2`

GET 请求的特点

- GET 请求可被缓存
- GET 请求保留在浏览器历史记录中
- GET 请求可被收藏为书签
- GET 请求不应在处理敏感数据时使用
- GET 请求有长度限制
- GET 请求只应当用于取回数据

POST 方法

当发送一个 HTTP 请求时，查询字符串（键/值对）是在 POST 请求的 HTTP 消息主体（body）中发送的，如：

`http://www.demo.com/test`

Request body 中： `name1=value1&name2=value2`

POST 请求的特点

- POST 请求不会被缓存
- POST 请求不会保留在浏览器历史记录中
- POST 不能被收藏为书签
- POST 请求对数据长度没有要求

对于第一次接触 HTTP 的同学，可能还是难以理解如何去使用它们，对吧？不要紧，上面的解释只是作为一种通用理解，刚入门的同学可以这样简单记忆：对安全性要求较高，或键/值对较多的，用 POST，其他的用 GET。后面从第 6 小节起，将介绍如何去使用 GET/POST 方法。

JSON

JSON 是一种轻量级的数据交换格式，易于阅读并能提升网络传输效率。JSON 的语法中，键/值对是用来保存对象的一种方式，如 `{"name1": "value1"}`。

对于熟悉 Python 的同学来讲，JSON 的格式和 Python 的字典很像。简单地理解，Python 的字典数据格式就是 JSON 的数据格式，但它们的不同也显而易见。JSON 的键/值对只允许使用双引号（`" "`）作为边界符号，而 Python 的字典则可以使用多种方式，包括单引号。

小结

本小节重点介绍了 App 客户端与服务器端的通信场景，包括数据交互方式，及 HTTP 和 JSON 的介绍，简单小结如下：

1. App 客户端与服务器端的消息请求应答使用 HTTP 协议；
2. App 客户端与服务器端的消息格式使用 JSON 格式；
3. HTTP 的重点请求方式有两种方式，分别为 GET 和 POST；
4. GET/POST 采用键/值对的方式，信息保密性要求高些，或键/值对多些时，使用 POST 方式。