

25网络通信

网络通信是客户端/服务器设计模式，本质上是并发。

网络通信相关的重要概念

IP 地址

网络接口由 IP 地址标识。IPv4 地址是 32 位的数字，写成四个 8 位的部分。例如（截至撰写本文时）：

- 18.9.22.69 是麻省理工学院（MIT）网页服务器的 IP 地址。
- 173.194.193.99 是谷歌网页服务器的地址。
- 104.47.42.36 是微软 Outlook 电子邮件处理程序的地址。
- 127.0.0.1 是回环地址或本地主机地址，它始终指向本地机器。从技术上讲，任何第一个八位组为 127 的地址都是回环地址，但 127.0.0.1 是标准的回环地址。

你可以在谷歌上查询你当前的 IP 地址。一般来说，当你携带笔记本电脑时，每次将机器连接到网络，它都可能被分配一个新的 IP 地址。

主机名

主机名是可以转换为 IP 地址的名称。单个主机名在不同时间可以映射到不同的 IP 地址，多个主机名也可以映射到同一个 IP 地址。例如：

- web.mit.edu 是麻省理工学院网页服务器的名称。

端口号

单个机器可能有多个服务器应用程序供客户端连接，因此我们需要一种方法将同一网络接口上的流量导向不同的进程。

网络接口有多个端口，由一个 16 位的数字标识。端口 0 被保留，因此端口号实际上从 1 到 65535（即 $2^{16} - 1$ ，最大的 16 位数字）。

服务器进程会绑定到特定的端口，即它会在该端口上监听。一个端口在同一时间只能有一个监听器，所以如果其他服务器进程尝试在同一个端口上监听，就会失败。

客户端需要知道服务器正在监听的端口号。有一些众所周知的端口被保留用于系统级进程，并为特定服务提供标准端口。例如：

- 端口 22 是标准的 SSH 端口。当你使用 SSH 连接到 athena.dialup.mit.edu 时，软件会自动使用端口 22。
- 端口 25 是标准的电子邮件服务器端口。
- 端口 80 是标准的网页服务器端口。当你在网页浏览器中连接到 URL <http://web.mit.edu> 时，它会连接到 18.9.22.69 的 80 端口。

当端口不是标准端口时，它会作为地址的一部分被指定。例如，URL <http://128.2.39.10:9000> 指的是 128.2.39.10 这台机器上的 9000 端口。

网页API

超文本传输协议（HTTP），【HTTP是什么？】https://www.bilibili.com/video/BV1zb4y127JU/?share_source=copy_web&vd_source=1eb3ec14476ecc498848d58cf48a175d

路由

网页服务器通常会将其提供服务的网站划分为多个部分，称为路由。例如，web.mit.edu 的服务可能有以下路由：

- /education（可通过 URL <http://web.mit.edu/education> 访问）
 - /research
 - /campus-life
- 等等。

在网页 API 中，路由通常定义请求的函数名称。例如，美国国家气象局 API 定义了以下路由：

- <https://api.weather.gov/points/>... 获取与经纬度点相关的信息
- <https://api.weather.gov/gridpoints/>... 获取特定区域的天气预报
- <https://api.weather.gov/alerts/active>... 获取特定区域的天气预警信息

参数

要将调用网页 API 视为调用函数，我们需要了解如何向其传递参数以及如何获取结果。

有三种提供参数的方式：

路径组件

URL 的路径由以 “/” 分隔的组件组成。除了指定函数名称外，其余组件可能是额外的参数，例如：

- <https://api.weather.gov/points/42.3541,-71.1104> 将经纬度对作为路径组件。

- <https://api.weather.gov/gridpoints/BOX/69,75/forecast>有三个路径组件参数：预报办公室 BOX、该办公室的网格位置 69,75 以及所需的信息类型 forecast。

查询参数

URL 的路径后面可以有一个可选的查询部分，以 “?” 开头，由以 “&” 分隔的 “名称 = 值” 对组成。你在网页浏览中可能已经见过这种方式，因为它是网页表单常用的方式。例如：

- <https://www.google.com/search?q=MIT>有一个查询参数 q，其值为 MIT。
- <https://api.weather.gov/alerts/active?area=MA&severity=Minor>有两个查询参数：area 的值为 MA，severity 的值为 Minor。

主体参数

这些参数不会出现在 URL 中。一些 HTTP 请求可以有一个主体，它是任意的额外字节序列。主体的格式可以是：

- 纯文本
- 二进制数据块，可能从文件加载
- JSON 数据，这是一组类似于查询参数的 “名称 = 值” 对

主体参数比路径参数或查询参数少见。HTTP 请求有两种常见类型，其中只有一种允许有主体：

- GET 请求是 curl 发出的常规请求，也是你在网页浏览器地址栏中输入 URL 或点击普通超链接时浏览器发出的请求。GET 请求有 URL 但没有主体，所以它们只能有路径和查询参数。GET 请求专为不改变服务器上任何状态的观察操作而设计。网页客户端可以放心，重复发出 GET 请求不会改变或创建任何新内容，所以如果浏览器需要重新发出 GET 请求以重新加载页面，它可以安全地这样做。
- POST 请求既有 URL 又有主体。你不能从网页浏览器的地址栏发出 POST 请求，但 curl 可以做到，网页表单也经常使用 POST 请求。POST 请求用于在服务器上更改或创建数据的操作：修改器、生产者、创建者。网页客户端在重新发出 POST 请求时会更加谨慎。在网页浏览过程中，你可能在某些时候看到过一个确认对话框，询问你是否真的想重新发送表单，这是因为浏览器不确定再次发出 POST 请求是否安全，例如可能会导致购买第二套昂贵的机票。

结果

客户端发出的 HTTP 请求通常会得到服务器的 HTTP 响应，其中包括两种提供结果的方式：

状态码

每个 HTTP 响应都有一个三位数字的状态码，指示请求是否成功。以下是一些常见的状态码：

- 200 OK 表示请求成功完成。
- 404 Not Found 表示网页 API 或网页不存在。
- 400 Bad Request 通常用于请求参数因某种原因无效的情况。
- 500 Internal Server Error 表示请求因其他原因失败。

回复主体

与请求一样，回复也可以附带任意类型的数据主体。典型的主体类型有：

- HTML（通常用于为人类浏览设计的网页服务器，而不是网页 API）
- 纯文本
- JSON

对于易于解析的简单结果（例如用逗号分隔的数字对），纯文本很适用。然而，对于复杂的结果，JSON 已成为首选的结果类型。我们在上面查询美国国家气象局的剑桥天气预报时，已经看到了 JSON 结果的样子。

HTTP有get请求，post请求等。

【HTTP请求方法：GET, POST, PUT, DELETE详解 - Curious Code】

https://www.bilibili.com/video/BV1nS411A71U/?share_source=copy_web&vd_source=1eb3ec14476ecc498848d58cf48a175d