

Chapter 3: HTTP Basics

By Tim Condon

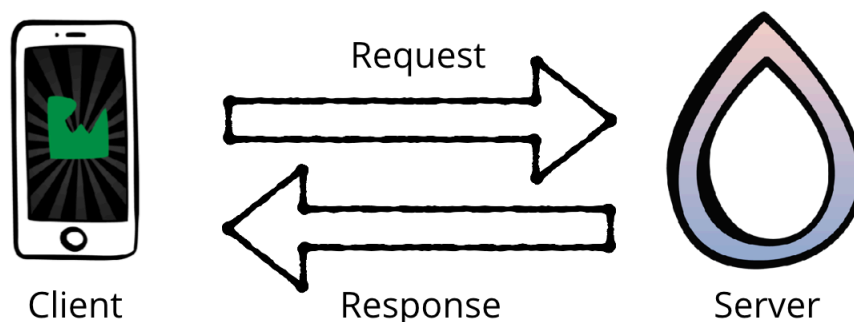
在您开始使用Vapor之前，您将首先回顾一下Web和HTTP如何运作的基础知识。

本章介绍了您需要了解的有关HTTP及其方法，以及最常见响应代码的信息。您还将了解Vapor如何增强您的Web开发体验，它的优势以及它与其他Swift构架的区别。

Powering the web

超文本传输协议（HTTP）是Web的基础。每次访问网站时，浏览器都会向服务器发送HTTP请求并从服务器接收响应。许多专用应用程序 - 从智能手机订购咖啡，将视频流传输到电视或在线玩游戏 - 在幕后都是使用HTTP。

HTTP的核心很简单。有一个客户端（如：一个iOS应用程序；一个Web浏览器；甚至一个简单的cURL会话）和一个服务器。客户端向服务器发送HTTP请求，返回HTTP响应。



HTTP请求

HTTP请求由几个部分组成：

- **The request line:** 它指定要使用的HTTP方法，请求的资源 and HTTP版本。
GET /about.html HTTP/1.1就是一个例子。您将在本章后面了解HTTP版本。
- **The host:** 用于处理请求的服务器的名称。当多个服务器托管在同一地址时，需要这样做。
- **Other request headers:** 例如Authorization, Accept, Cache-Control, Content-Length, Content-Type等。
- **Optional request data:** 如果HTTP方法需要。

HTTP方法指定客户端请求的操作类型。 HTTP规范定义了以下方法：

- GET
- HEAD
- POST
- PUT
- DELETE
- CONNECT
- OPTIONS
- TRACE
- PATCH

最常见的HTTP方法是GET。它允许客户端从服务器检索资源。单击浏览器中的链接或点击新闻应用程序中的故事都会触发对服务器的GET请求。

另一种常见的HTTP方法是POST。它允许客户端将数据发送到服务器。输入用户名和密码后单击登录按钮可以触发对服务器的POST请求。当您阅读本书时，您还将了解到其他HTTP方法。

通常，服务器需要的不仅仅是资源的名称才能正确地为请求提供服务。此附加信息在 **request headers** 中发送。Request headers 只不过是键值对。

一些常见的请求标头是：Authorization，Cookie，Content-Type 和 Accept。您将在后面的章节中学习 Vapor 如何使用其中的一些来使您的服务器端应用程序更加强大。

HTTP 响应

服务器在处理请求时返回 HTTP 响应。HTTP 响应包括：

- **The status line:** 包含版本，状态码和消息
- **Response headers**
- **An optional response body**

状态码及其相关消息指示请求的结果。有许多状态码，但您不会使用或遇到大多数状态码。根据第一个数字，它们分为 5 组：

- **1:** 信息响应。这些不经常发生。
- **2:** 成功响应。最常见的 200 OK 表示请求已成功完成。
- **3:** 重定向响应。这些经常使用。
- **4:** 客户端错误。其中最常见的是 404 Not Found。您可能已经看到了一些不同且有趣的 404 页面！
- **5:** 服务器错误。这经常表示服务器配置不正确，资源耗尽或服务器端应用程序中存在错误。

甚至有一个愚人节的笑话状态码：418 I'm a teapot!

响应可以包括 **response body**，诸如页面的 HTML 内容，图像文件或资源的 JSON 描述。然而，response body 是可选的，并且一些响应代码 - 例如 204 No Content - 将不具有响应内容。

最后，响应可能包括一些 **response headers**。这些类似于前面描述的请求标头。一些常见的响应标头是：Set-cookie，WWW-Authenticate，Cache-Control 和 Content-Length。

HTTP in web browsers

当您要求浏览器加载页面时，它会发送该页面的HTTP GET请求。服务器返回响应body中的HTML。当浏览器解析HTML时，它会为任何页面引用的资源（图像，JavaScript，CSS）生成额外的HTTP GET请求。

正确格式的HTML页面包含<head>和<body>部分。处理页面时，浏览器会等待，直到它收到<head>部分中引用的所有外部资源来渲染页面。客户端在接收它们时渲染<body>部分中引用的资源。

Web浏览器仅使用GET和POST HTTP方法。大多数浏览器请求都是GET请求。浏览器可以使用POST来提交表单数据或上传文件。这将在后面的章节中变得重要；然后你会学习解决这个问题方法。也无法自定义浏览器发送的请求标头。

HTTP in iOS apps

您的iOS应用程序 - 这也适用于其他HTTP客户端，例如Rested，JavaScript，Postman - 受到的限制要少得多。这些应用程序能够使用所有HTTP方法，添加自定义请求标头并实现自定义响应处理。这是更多的工作，但灵活性允许您自由地开发您需要的西。

HTTP 2.0

今天大多数Web服务使用HTTP版本1.1 - 在1997年1月发布为[RFC 2068](#)。到目前为止，您学到的所有内容都是HTTP/1.1的一部分，除非另有说明，否则1.1是本书中使用的版本。

HTTP/2扩展了客户端和服务端之间的通信，以提高效率并减少延迟。单个请求与HTTP/1.1中的请求相同，但它们可以并行进行。服务器可以预测客户端的请求，并在客户端请求数据之前将数据（例如样式表和图像）推送到客户端。Vapor在其客户端和服务端功能中都支持HTTP/1.1和HTTP/2。

REST

REST或代表性状态转移是与HTTP密切相关的架构标准。应用程序使用的许多API都是REST API，您经常会听到这个术语。您将在第7章“CRUD Database Operations”中了解有关REST以及它与HTTP和CRUD的关系的更多信息。REST提供了一种定义从API访问资源的通用标准的方法。例如，对于缩略词API，您可以定义以下端点：

- **GET** `/api/acronyms/`: get all acronyms.
- **POST** `/api/acronyms`: create a new acronym.
- **GET** `/api/acronyms/1`: get the acronym with ID 1.
- **PUT** `/api/acronyms/1`: update the acronym with ID 1.
- **DELETE** `/api/acronyms/1`: delete the acronym with ID 1.

拥有从REST API访问资源的通用模式简化了构建客户端的过程。

Why use Vapor?

使用Swift和Vapor进行服务器端应用程序开发是一种独特的体验。与许多传统的服务器端语言相比 - 例如PHP，JavaScript，Ruby - Swift是强类型和静态类型的。此特性大大减少了iOS应用程序中的运行时崩溃次数，您的服务器端应用程序也将享受此优势。

服务器端Swift的另一个潜在好处是提高了性能。因为Swift是一种编译语言，所以使用Swift编写的应用程序可能比使用解释语言编写的应用程序执行得更好。

不过，编写服务器端Swift应用程序的最大原因是你可以使用Swift！Swift是增长最快，最受喜爱的语言之一，其现代语法和功能结合了多种语言的精华。如果您目前正在为iOS开发，那么您可能已经很熟悉该语言。这意味着您可以开始在服务器端应用程序和iOS应用程序之间共享核心业务逻辑代码。

选择Swift也意味着您可以使用Xcode开发服务器应用程序！虽然Linux里的Foundation是iOS和macOS上的一部分，但您可以在Xcode中进行大部分开发。这使您可以在IDE中访问强大的调试功能，这是大多数服务器端语言所没有的功能。

How Vapor compares

Vapor不是唯一可用的服务器端Swift框架，为什么选择Vapor呢？选择Vapor有很多充分的理由；这里有几个。

在撰写本文时，Vapor是唯一一个在整个API中使用Codable的服务器端框架。这允许您编写非常少的代码来保存数据库中的模型并将它们作为JSON返回。它还允许您接受数据并在单行代码中将其转换为类型安全的struct。与之前版本的框架相比，Vapor 3使用Codable大大减少了您需要编写的代码量。

Vapor还采用完全无阻塞架构。如您所知，当服务器应用程序收到HTTP请求时，它必须返回HTTP响应。在阻塞体系结构中，如果您的应用程序需要很长时间来处理请求 - 例如，等待数据库查询返回 - 那么应用程序无法响应该线程上的其他请求。正如您将在下一章中学到的，产生多个线程并在它们之间切换会带来额外的开销。在非阻塞体系结构中，切换请求并处理其他请求，直到原始请求的结果准备就绪。因此，非阻塞框架的性能明显优于阻塞框架。Vapor在每个级别都是非阻塞的，包括数据库驱动程序。

最后，关于Vapor的最佳功能是它的社区。Vapor拥有最活跃和最具活力的社区，这是其日益普及的一个主要因素！