Python Spider

张志鹏

Published
with GitBook

目錄

这是一本python 爬虫的教学书籍

目录

第一部分 入门指南

第1章 HTTP概述

第2章 网络爬虫概述

第3章 小试牛刀，写一个小爬虫

第5章 Python库介绍

第6章 高级爬虫

part 1 —— 第一部分 入门指南

第一部分部分宽泛的讲解学习爬虫要了解到的哪些知识，包括HTTP介绍、爬虫介绍，并且做一个稍微复杂爬虫示例。
第1章主要讲解HTTP，什么是HTTP，它是如何运作的。
第2章讲解爬虫的基本知识，什么是爬虫，爬虫搜索策略和类型。
第3章介绍如何抓取数据，并以一个示例进行介绍如何分析一个爬虫需求。

第1章 —— HTTP概述

> 超文本传输协议（HyperText Transfer Protocol，HTTP）是互联网上应用最为广泛的一种网络协议。设计HTTP最初的目的是为了提供一种发布和接收HTML页面的方法。通过HTTP或者HTTPS协议请求的资源由统一资源标识符（Uniform Resource Identifiers，URI）来标识。
>
> HTTP的发展是万维网协会（World Wide Web Consortium，W3C）和互联网工程任务组（Internet Engineering Task Force，IETF）合作的结果，（他们）最终发布了一系列的RFC，其中最著名的是1999年6月公布的RFC 2616，定义了HTTP协议中现今广泛使用的一个版本—HTTP 1.1。

本章将讲解HTTP是什么，web客户端、web服务器是什么，它们之间是如何通过HTTP进行通信的，以及通信结构、格式是什么样子。

> HTTP是一个客户端终端（用户）和服务器端（网站）请求和应答的标准（TCP）。通过使用Web浏览器、网络爬虫或者其它的工具，客户端发起一个HTTP请求到服务器上指定端口（默认端口为80）。我们称这个客户端为用户代理程序（user agent）。应答的服务器上存储着一些资源，比如HTML文件和图像。我们称这个应答服务器为源服务器（origin server）。在用户代理和源服务器中间可能存在多个“中间层”，比如代理、网关或者隧道（tunnel）。
>
> 尽管TCP/IP协议是互联网上最流行的应用，HTTP协议中，并没有规定必须使用它或它支持的层。事实上，HTTP可以在任何互联网协议上，或其他网络上实现。HTTP假定其下层协议提供可靠的传输。因此，任何能够提供这种保证的协议都可以被其使用。因此也就是其在TCP/IP协议族使用TCP作为其传输层。
>
> 通常，由HTTP客户端发起一个请求，创建一个到服务器指定端口（默认是80端口）的TCP连接。HTTP服务器则在那个端口监听客户端的请求。一旦收到请求，服务器会向客户端返回一个状态，比如"HTTP/1.1 200 OK"，以及返回的内容，如请求的文件、错误消息、或者其它信息。

那么，HTTP究竟是什么呢？

通俗地讲，HTTP（又指HTTP协议，全名为“超文本传输协议”）指一种约束，协议并不是听起来的特别的深奥，它只是一个专业词。用于服务端（网站）和客户端（浏览器/用户）之间的数据传输。

要理解HTTP，用我们日常生活中能够形象表示的HTTP协议的东西，就是收发信件。

服务端，就是指一个资源站点。服务端，并不是我们想象中的，云里来雾里去。每一台电脑，都是服务端，都是客户端。 通俗地讲，服务器可以理解为邮局、收信人家门口信箱。

客户端，就是指某个用户(电脑)、用户浏览器、访问服务端的程序。 通俗地讲，客户端可以理解为发信人家里门口的信箱。

HTTP协议，就是指发送、接受信件过程（称之为邮递）的一种约束，可以指过程，也可以指对过程的规范。具体，可以认为是信封上面要写的内容，必须要写的内容有发信人、发信地址、发信邮编、收信人、收信地址、收信邮编、邮票等。这些内容，就可以认为是HTTP协议，是必须要填的的信息，用于确认信件可以准确传达到指定邮局、指定收信人。