

# 网络数据爬取与处理

---

## 第二章 网络爬虫原理



- 1. 网络爬虫概念**
- 2. 网络爬虫原理**
- 3. HTTP协议**
- 4. 数据在网页中的存在形式**
- 5. 网络爬虫的法律与道德约束**
- 6. 网络爬虫实例**

## **2-1 网络爬虫概念**

# 网络爬虫概念

- 网络爬虫(Web Crawler)又名网络蜘蛛(Web spider), 是一种**模拟浏览器行为**, 自动化浏览网络资源并采集互联网上**公开数据**的程序。
- 网络爬虫的本质是一个递归的过程。首先获取一个URL对应的网页内容, 然后从该网页中提取另一个URL, 再获取该URL对应的网页内容, 并不断的循环这一过程。

# 网络爬虫的类型

通用爬虫：爬取全网数据 搜索引擎

聚焦爬虫：聚焦于某个主题、某个网站、某个网页

## 爬取网页

数据量小，但目标数据包含在大量网页中，  
对爬取速度不敏感，替代繁琐的人工操作。

Requests库

## 爬取网站

数据量较大，目标数据为整个网站的数据，  
对爬取速度敏感。

Scrapy库

## 爬取全网

数据量非常大，目标数据为全网数据，  
对爬取速度非常敏感。搜索引擎

定制开发

聚焦爬虫

通用爬虫

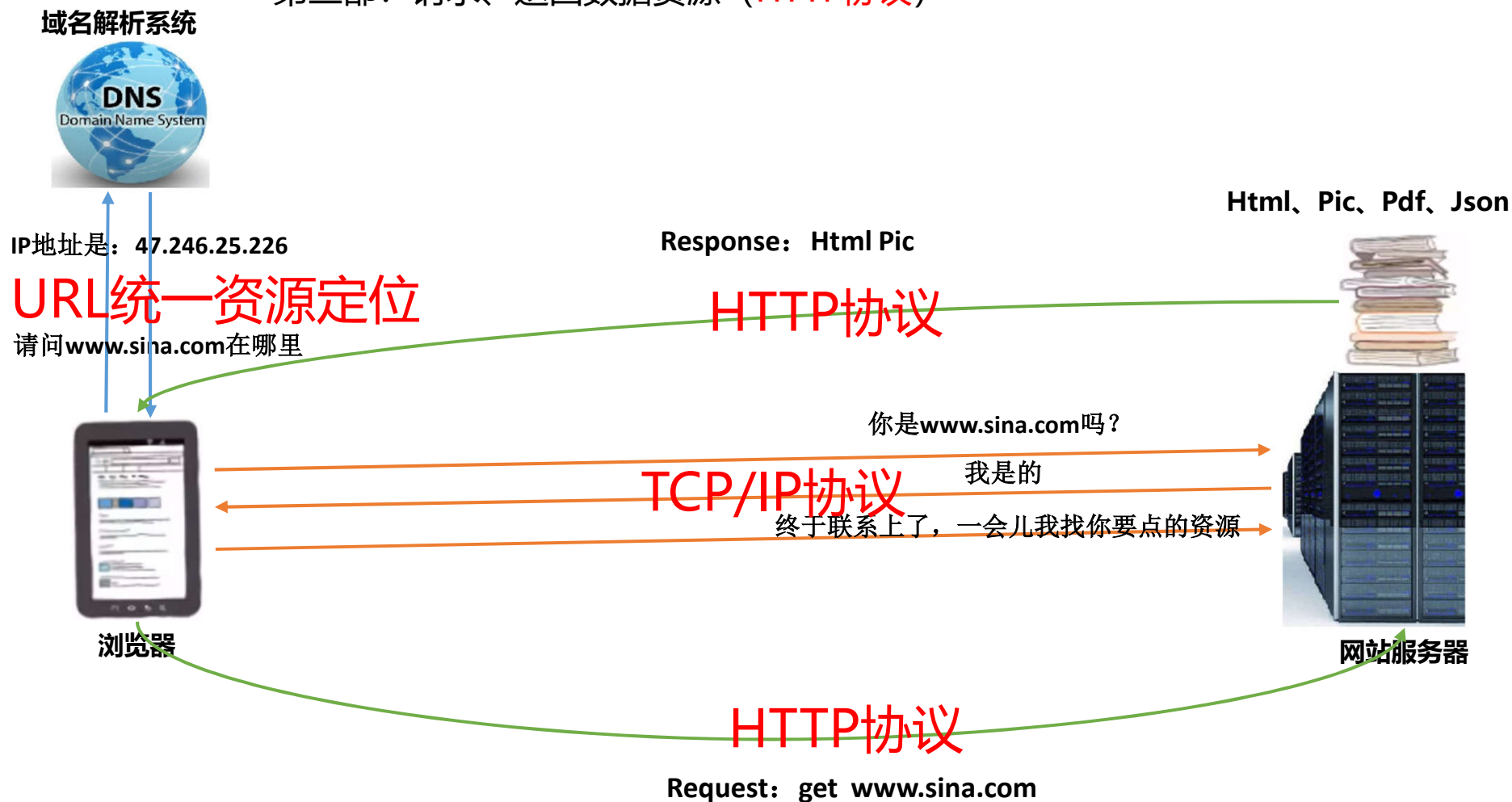
## **2-2 网络爬虫原理**

# 浏览器访问互联网资源的过程

第一步：寻找数据资源所在的服务器（URL统一资源定位）

第二步：与服务器建立联系（TCP/IP协议）

第三步：请求、返回数据资源（HTTP协议）



# 网络爬虫原理

- 寻找服务器、并和服务器建立连接，由URL和TCP/IP完成。
- 模拟浏览器通过HTTP协议向WEB服务器发送请求，并获得WEB服务器返回的资源，从返回的资源中提取目标数据。

## 网络爬虫的主要任务：

第一步：向目标URL地址发送Request请求

第二步：获得服务器Response返回的资源

第三步：从获得的资源中过滤出目标数据

# 使用开发者工具侦测浏览器发送的请求

打开任意网页后，单击鼠标右键单击“检查元素”或“inspect”菜单，进入浏览器的**开发者工具**界面。以网易新闻主页为例：<https://news.163.com/>

The screenshot shows the 163.com website in a browser. The Chrome DevTools Network tab is open, displaying a list of requests. The first request, 'news.163.com', is selected. The 'Headers' sub-tab is active, showing the 'General' section with the following details:

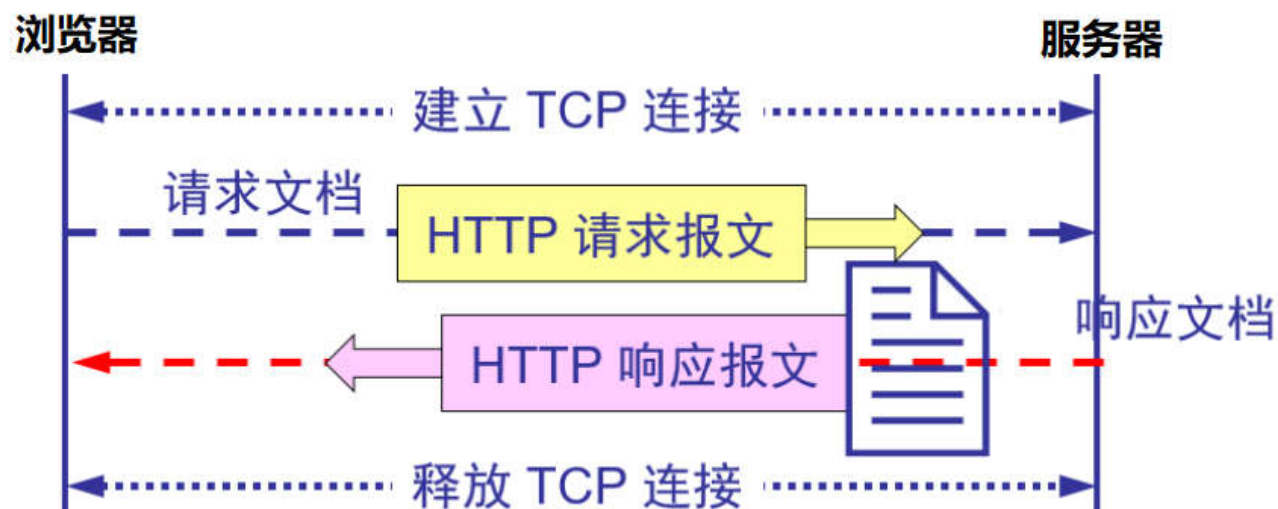
- Request URL: <https://news.163.com/>
- Request Method: GET
- Status Code: 200
- Remote Address: 60.163.162.47:443
- Referrer Policy: strict-origin-when-cross-origin

The 'Response Headers' section is also visible, showing details like 'age: 6', 'cache-control: no-cache, no-store, private', 'cdn-ip: 60.163.162.47', 'cdn-source: baishan', 'cdn-user-ip: 101.228.32.125', 'content-encoding: gzip', 'content-type: text/html; charset=GBK', 'date: Sat, 26 Sep 2020 07:07:25 GMT', 'expires: Sat, 26 Sep 2020 07:07:50 GMT', 'last-modified: Sat, 26 Sep 2020 07:07:01 GMT', 'p3p: CP=CAO PSA OUR', 'server: nginx', 'status: 200', 'vary: Accept-Encoding', 'x-cache-remote: HIT', and 'x-ser: BC44\_dx-1t-yd-shandong-jinan-5-cache-5, BC45\_1'.

## **2-3 HTTP协议**

# HTTP协议

HTTP协议（HyperText Transfer Protocol，超文本传输协议）是用于从网站服务器传输超资源到本地浏览器的传输协议。HTTP协议必须是浏览器首先发起请求，然后服务器回送响应。**一次完整的浏览器请求与服务器响应的过程，包含四个步骤：**



1. 首先客户端与服务器需要建立连接，只要单击某个超级链接，HTTP协议便开始工作；
2. 建立连接后，客户端发送一个请求给服务器；
3. 服务器接到请求后，返回相应的响应信息；
4. 接收服务器所返回的信息通过浏览器显示在用户的显示屏上，然后与服务器断开连接。

# HTTP请求报文格式：请求行 + 请求头 + 数据体

|       |     |     |     |      |        |     |      |
|-------|-----|-----|-----|------|--------|-----|------|
| 请求方法  | 空格  | URL | 空格  | 协议版本 | 回车符    | 换行符 | 请求行  |
| 头部字段名 | :   | 值   | 回车符 | 换行符  | } 请求头部 |     |      |
| ...   |     |     |     |      |        |     |      |
| 头部字段名 | :   | 值   | 回车符 | 换行符  |        |     |      |
| 回车符   | 换行符 |     |     |      |        |     |      |
|       |     |     |     |      |        |     | 请求数据 |

**请求行：**规定了发送请求的方法，目标URL地址以及HTTP协议的版本。**请求方法**是请求行包含的最为重要的信息，它指明了客户端向服务器交互信息的方法，这也是爬虫程序模拟浏览器向服务器发送信息得关键。

| 方法名  | 功能                                                           |
|------|--------------------------------------------------------------|
| GET  | 请求由 Request-URI 所标识的资源，一般情况下 GET 请求会通过 URL 地址向服务器传送信息        |
| POST | 在 Request-URI 所标识的资源后附加需要传送的信息，目前主流网站的登录账号和密码信息都通过 POST 方法传送 |

**请求头：**请求头的格式类似于python的字典类型，由一系列键值对构成。

| 构成元素            | 功能                       |
|-----------------|--------------------------|
| Accept          | 客户端可以接受的响应格式             |
| User-Agent      | 标识客户端浏览器                 |
| Accept-Encoding | 表明浏览器可以接受的编码方式           |
| Accept-Language | 表明浏览器可以接受的语言种类           |
| connection      | 用来告诉服务器是否可以维持固定的 HTTP 连接 |
| Cookie          | 向服务器发送 Cookie            |
| Host            | 表明 URL 中的 Web 名称和端口号     |
| Referer         | 先前网页的地址                  |
| Content-Type    | 表明请求的内容类型                |
| Accept-Charset  | 表明客户端可以接受的字符编码           |
| Accept-Encoding | 表明客户端可以接受的编码方式           |

**数据体：**根据请求行和请求头规定的方法与格式，请求正文主要包含客户端提交给服务器的数据信息。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/808025103120006054>