Author：Black

HTTP基本原理

1.1 URL

URI：Uniform Resource Identifier 统一资源标识符

URL：Uniform Resource Locator 统一资源定位符

URN: Uniform Resource Name 统一资源名称

URI包括URL和URN，URL/URI指定了资源的唯一访问方式。包括了协议https，地址，和文件名称

1.2.超文本 hypertext

组成网页的源代码HTML称为超文本

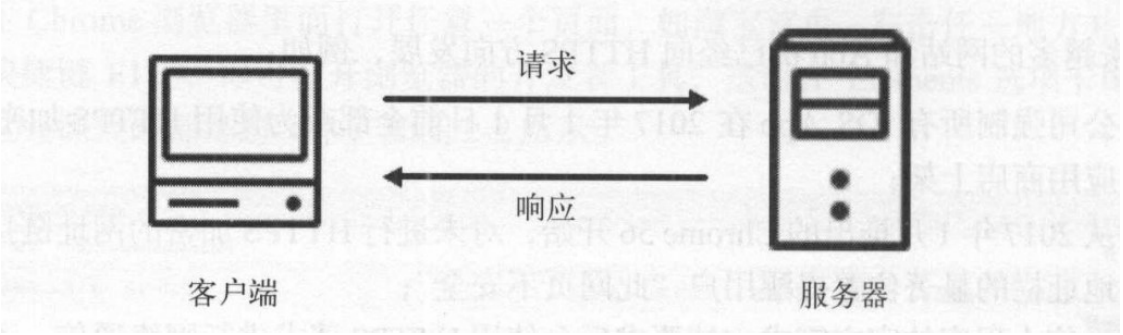
1.3.HTTP 和 HTTPs

HTTP：Hyper Text Transfer Protocol 超文本传输协议

HTTPs：Hyper Text Transfer Protocol over Secure Socket Layer 以安全为目的的HTTP协议，就是安全版的HTTP协议，在HTTP下加入SSL层

1.4，Http的请求过程：

我们在浏览器输入URL之后，浏览器会向网站的服务器发送了一个请求，服务器收到这个请求后会返回对应的响应，响应里包括了页面的代码等内容，浏览器会对这些代码进行解析，将网页呈现出来。



1.5 请求

组成： 请求方法（Request Method）、请求的网址(Request URL）、请求头（Request Headers）、请求体（Request Body）。

1.5.1 请求方法:

GET 和 POST

GET 和 POST 请求方法有如下区别。

GET 请求中的参数包含在 URL 里面，数据可以在 URL 中看到，而 POST 请求的 URL 不会包含这些数据，数据都是通过表单形式传输的，会包含在请求体中 。

GET 请求提交的数据最多只有 1024 字节，而 POST 方式没有限制 。

账户密码保密要求很高，一般使用POST，上传文件时，由于文件内容比较大，也会选用 POST 方式。

其他请求方式：

表 2-1　其他请求方法

方　法	描　述
GET	请求页面，并返回页面内容
HEAD	类似于 GET 请求，只不过返回的响应中没有具体的内容，用于获取报头
POST	大多用于提交表单或上传文件，数据包含在请求体中
PUT	从客户端向服务器传送的数据取代指定文档中的内容
DELETE	请求服务器删除指定的页面
CONNECT	把服务器当作跳板，让服务器代替客户端访问其他网页
OPTIONS	允许客户端查看服务器的性能
TRACE	回显服务器收到的请求，主要用于测试或诊断

1.5.2 请求网址：即URL

1.5.3 请求头：

请求头，用来说明服务器要使用的附加信息，比较重要的信息有 Cookie 、 Referer 、 User-Agent 等。

图 15-4　HTTP 请求报文

Accept： 请求报头域，用于指定客户端可接受哪些类型的信息。

Accept-Language： 指定客户端可接受的语言类型。

Accept-Encoding： 指定客户端可接受的内容编码。

Host： 用于指定请求资源的主机 IP 和端口号，其内容为请求 URL 的原始服务器或网关的位置。 从 HTTP 1. l 版本开始，请求必须包含此内容。

Cookie： 也常用复数形式 Cookies ，这是网站为了辨别用户进行会话跟踪而存储在用户本地的数据 。 它的主要功能是维持当前访问会话。 例如，我们输入用户名和密码成功登录某个网站后，服务器会用会话保存登录状态信息，后面我们每次刷新或请求该站点的其他页面时，会发现都是登录状态，这就是 Cookies 的功劳。 Cookies 里有信息标识了我们所对应的服务器

的会话，每次浏览器在请求该站点的页面时，都会在请求头中加上 Cookies 并将其发送给服务器，服务器通过 Cookies 识别出是我们自己，并且查出当前状态是登录状态，所以返回结果就是登录之后才能看到的网页内容。

Referer： 此内容用来标识这个请求是从哪个页面发过来的，服务器可以拿到这一信息并做相

应的处理，如做来源统计、防盗链处理等。

User-Agent： 简称 UA ，它是一个特殊的字符串头，可以使服务器识别客户使用的操作系统及版本 、 浏览器及版本等信息。 在做爬虫时加上此信息，可以伪装为浏览器；如果不加，很可能会被识别为爬虫 。

Content-Type： 也叫互联网媒体类型 （ Internet Media Type ）或者 MIME 类型，在 HTTP 协议消息头中，它用来表示具体请求中的媒体类型信息 。 例如， text/html 代表 HTML 格式，image/gif 代表 GIF 图片， app lication/jso n 代表 J SON 类型，更多对应关系可以查看此对照表：

1.5.4：请求体

请求体 - 般承载的内容是 POST 请求中的表单数据，而对于 GET 请求，请求体则为空 。

Content-Type 和 POST 提交数据方式的关系：

表 2-2　Content-Type 和 POST 提交数据方式的关系

Content-Type	提交数据的方式
application/x-www-form-urlencoded	表单数据
multipart/form-data	表单文件上传
application/json	序列化 JSON 数据
text/xml	XML 数据

1.1.6 响应

组成：响应状态码（ Response Status Code ）、响应头(Response Headers ）和响应体 （ Response Body ）

响应状态码：响应状态码表示服务器的响应状态

HTTP状态码列表

状态码	状态码英文名称	中文描述
100	Continue	继续。客户端应继续其请求
101	Switching Protocols	切换协议。服务器根据客户端的请求切换协议。只能切换到更高级的协议，例如，切换到HTTP的新版本协议
200	OK	请求成功。一般用于GET与POST请求
201	Created	已创建。成功请求并创建了新的资源
202	Accepted	已接受。已经接受请求，但未处理完成
203	Non-Authoritative Information	非授权信息。请求成功。但返回的meta信息不在原始的服务器，而是一个副本
204	No Content	无内容。服务器成功处理，但未返回内容。在未更新网页的情况下，可确保浏览器继续显示当前文档
205	Reset Content	重置内容。服务器处理成功，用户终端（例如：浏览器）应重置文档视图。可通过此返回码清除浏览器的表单域
206	Partial Content	部分内容。服务器成功处理了部分GET请求
300	Multiple Choices	多种选择。请求的资源可包括多个位置，相应可返回一个资源特征与地址的列表用于用户终端（例如：浏览器）选择
301	Moved Permanently	永久移动。请求的资源已被永久的移动到新URI，返回信息会包括新的URI，浏览器会自动定向到新URI。今后任何新的请求都应使用新的URI代替
302	Found	临时移动。与301类似。但资源只是临时被移动。客户端应继续使用原有URI
303	See Other	查看其它地址。与301类似。使用GET和POST请求查看
304	Not Modified	未修改。所请求的资源未修改，服务器返回此状态码时，不会返回任何资源。客户端通常会缓存访问过的资源，通过提供一个头信息指出客户端希望只返回在指定日期之后修改的资源
305	Use Proxy	使用代理。所请求的资源必须通过代理访问
306	Unused	已经被废弃的HTTP状态码
307	Temporary Redirect	临时重定向。与302类似。使用GET请求重定向
400	Bad Request	客户端请求的语法错误，服务器无法理解
401	Unauthorized	请求要求用户的身份认证
402	Payment Required	保留，将来使用
403	Forbidden	服务器理解请求客户端的请求，但是拒绝执行此请求
404	Not Found	服务器无法根据客户端的请求找到资源（网页）。通过此代码，网站设计人员可设置"您所请求的资源无法找到"的个性页面
405	Method Not Allowed	客户端请求中的方法被禁止
406	Not Acceptable	服务器无法根据客户端请求的内容特性完成请求

407	Proxy Authentication Required	请求要求代理的身份认证，与401类似，但请求者应当使用代理进行授权
408	Request Time-out	服务器等待客户端发送的请求时间过长，超时
409	Conflict	服务器完成客户端的 PUT 请求时可能返回此代码，服务器处理请求时发生了冲突
410	Gone	客户端请求的资源已经不存在。410不同于404，如果资源以前有现在被永久删除了可使用410代码，网站设计人员可通过301代码指定资源的新位置
411	Length Required	服务器无法处理客户端发送的不带Content-Length的请求信息
412	Precondition Failed	客户端请求信息的先决条件错误
413	Request Entity Too Large	由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个Retry-After的响应信息
414	Request-URI Too Large	请求的URI过长（URI通常为网址），服务器无法处理
415	Unsupported Media Type	服务器无法处理请求附带的媒体格式
416	Requested range not satisfiable	客户端请求的范围无效
417	Expectation Failed	服务器无法满足Expect的请求头信息
500	Internal Server Error	服务器内部错误，无法完成请求
501	Not Implemented	服务器不支持请求的功能，无法完成请求
502	Bad Gateway	作为网关或者代理工作的服务器尝试执行请求时，从远程服务器接收到了一个无效的响应
503	Service Unavailable	由于超载或系统维护，服务器暂时的无法处理客户端的请求。延时的长度可包含在服务器的Retry-After头信息中
504	Gateway Time-out	充当网关或代理的服务器，未及时从远端服务器获取请求
505	HTTP Version not supported	服务器不支持请求的HTTP协议的版本，无法完成处理

响应头：响应头包含了服务器对请求的应答信息

Date： 标识响应产生的时间。

Last-Modified： 指定资源的最后修改时间 。

Content-Encoding： 指定响应 内容的编码。

Server： 包含服务器的信息 ， 比如名 称 、版本号等。

Content-Type： 文档类型 ， 指定返回的数据类型是什么 ， 如 tex t/ htm l 代表返回 HTML 文档，application/x-javascript !J! U 代表返回 JavaScript 文件， image/jpeg 则代表返回图片 。

Set- Cookie： 设置 Cookies 。 响应头 中的 Set- Cooki e 告诉浏览器需要将此内容放在 Cookies中， 下次请求携带 Cookies 请求。

Expires： 指定响应的过期时间 ， 可以使代理服务器或浏览器将加载的 内容更新到缓存巾 。 如果再次访问 时，就可 以直接从缓存中加载， 降低服务器负载，缩短加载时间 。

响应体：服务器返回的数据

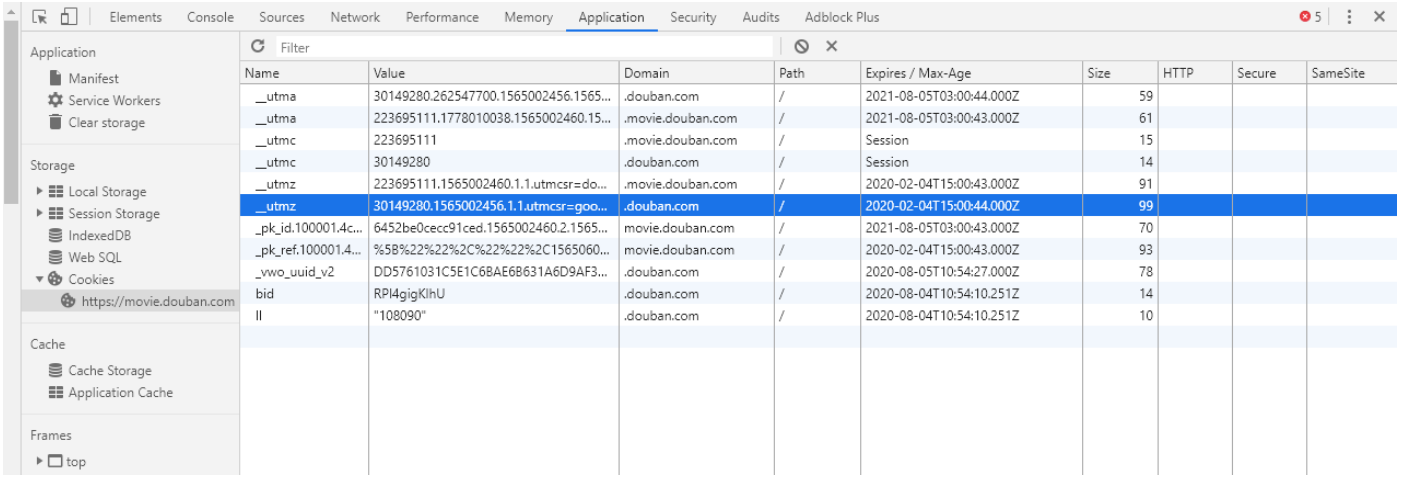
Cookie相关内容：

HTTP 协议对事务处理是没有记忆能力的，也就是说服务器不知道客户端是什么状态。Cookies 指某些网站为了辨别用户身份 、 进行会话跟踪而存储在用户本地终端上的数据。

Cookie实现原理：当客户端第一次请求服务器时，服务器会返回一个请求头中带有 Set-Cookie 字段的响应给客户端，用来标记是哪一个用户，客户端浏览器会把 Cookies 保 存起来 。 当浏览器下一次再请求该网站时，浏览器会把此 Cookies 放到请求头一起提交给服务器，Cookies 携带了会话 ID 信息，服务器检查该 Cookies 即可找到对应的会话是什么，然后再判断会话来以此来辨认用户状态。

如何查看Cookie：，在浏览器开发者工具中打开Application 选项卡（Chrome浏览器按F12）

Cookie内容：



Name： 该 Cookie 的名称。一旦创建，该名称便不可更改。

Value： 该 Cookie 的值。 如果值为 Unicode 字符，需要为字符编码。 如果值为二进制数据，则需要使用 BASE64 编码。98 第 2 幸爬虫基础

Domain： 可以访问该 Cookie 的域名 。 例如，如果设置为. zhihu.com ，则所有以 zh ihu .co m 结尾的域名都可以访问该 Cookie 。

Max Age： 该 Cookie 失效的时间， 单位为秒，也常和 Expires 一起使用，通过它可以计算其有效时间 。 Max Age 如果为正数 ， 贝lj 该 Cookie 在 Max Age 秒之后失效。 如果为负数，则关闭浏览器时 Cookie 即失效，浏览器也不会以任何形式保存该 Cookie 。

Path： 该 Cookie 的使用路径 。 如果设置为 / path/ ，则只有路径为 / path/ 的页面可以访问该Cookie 。 如果设置为人 则本域名下的所有页面都可以访问该 Cookie 。Size 字段： 此Cookie 的大小。

HTTP 字段： Cookie 的 httponly 属性。 若此属性为 true ， 则只有在 HTTP 头中会带有此Cookie 的信息，而不能通过 document.cookie 来访问此 Cookie 。

Secure： 该 Cookie 是否仅被使用安全协议传输。 安全协议有 HTTPS 和 SSL 等，在网络上传输数据之前先将数据加密。 默认为 false 。