URL 简介

目录

- 概述
- 网址的组成部分
  - 协议
  - 主机
  - 端口
  - 路径
  - 查询参数
  - 锚点
- URL 字符
- 绝对 URL 和相对 URL
- <base>

概述

URL 是“统一资源定位符”（Uniform Resource Locator）的首字母缩写，中文译为“网址”，表示各种资源的互联网地址。下面就是一个典型的 URL。

```
https://www.example.com/path/index.html
```

所谓资源，可以简单理解成各种可以通过互联网访问的文件，比如网页、图像、音频、视频、JavaScript 脚本等等。只有知道了它们的 URL，才能在互联网上获取它们。

只要资源可以通过互联网访问，它就必然有对应的 URL。一个 URL 对应一个资源，但是同一个资源可能对应多个 URL。

URL 是互联网的基础。互联网之所以“互联”，就是因为网页可以通过“链接”（link），包含其他 URL。用户只要点击，就可以从一个 URL 跳转到另一个 URL，前往不同的网站。

网址的组成部分
URL 由多个部分组成。下面是一个比较复杂的 URL，实际的 URL 通常不会有这么多部分。

https://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#anchor

我们看看，这个 URL 的各个部分。

协议

协议（ scheme）是浏览器请求服务器资源的方法，上例是 https://的部分，表示使用 HTTPS 协议。

互联网支持多种协议，必须指明网址使用哪一种协议，默认是 HTTP
协议。也就是说，如果省略协议，直接在浏览器地址栏输入 www.example.com，那么浏览器默认会访问 http://www.example.com。 HTTPS 是 HTTP 的加密版本，出于安全考虑，越来越多的网站使用这个协议。

HTTP 和 HTTPS 的协议名称后面，紧跟着一个冒号和两个斜杠（ ://）。其他协议不一定如此，邮件地址协议 mailto: 的协议名后面只有一个冒号，比如 mailto:foo@example.com。

主机

主机（ host）是资源所在的网站名或服务器的名字，又称为域名。上例的主机是 www.example.com。

有些主机没有域名，只有 IP 地址，比如 192.168.2.15。这种情况常常出现在局域网。

端口

同一个域名下面可能同时包含多个网站，它们之间通过端口（ port）区分。 “端口 ”
就是一个整数，可以简单理解成，访问者告诉服务器，想要访问哪一个网站。 HTTP 协议的默认端口是 80 ，如果省略了这个参数，服务器就会返回 80端口的网站。

端口紧跟在域名后面，两者之间使用冒号分隔，比如 www.example.com:80。

路径

路径（ path）是资源在网站的位置。比如， /path/index.html这个路径，指向网站的 /path
子目录下面的网页文件 index.html。

互联网的早期，路径是真实存在的物理位置。现在由于服务器可以模拟这些位置，所以路径只是虚拟位置。

路径可能只包含目录，不包含文件名，比如 /foo/
，甚至结尾的斜杠都可以省略。这时，服务器通常会默认跳转到该目录里面的 index.html文件（即等同于请求 /foo/index.html
），但也可能有其他的处理（比如列出目录里面的所有文件），这取决于服务器的设置。一般来说，访问
www.example.com这个网址，很可能返回的是网页文件 www.example.com/index.html。

查询参数

查询参数（ parameter）是提供给服务器的额外信息。参数的位置是在路径后面，两者之间使用 ?
分隔，上例是 ?key1=value1&key2=value2。

查询参数可以有一组或多组。每组参数都是键值对（ key-value pair）的形式，同时具有键名 (key)和键值 (value)，它们之间使用等号（ =）连接。比如， key1=value就是一个键值对， key1是键名， value1 是键值。

多组参数之间使用 &连接，比如 key1=value1&key2=value2。

锚点

锚点（ anchor）是网页内部的定位点，使用 #加上锚点名称，放在网址的最后，比如 #anchor
。浏览器加载页面以后，会自动滚动到锚点所在的位置。

锚点名称通过网页元素的 id属性命名，详见《元素的属性》一章。

URL 字符
URL 的各个组成部分，只能使用以下这些字符。

26个英语字母（包括大写和小写）
10个阿拉伯数字
连词号（ -）
句点（.）
下划线（ _）

此外，还有 18个字符属于 URL 的保留字符，只能在给定的位置出现。比如，查询参数的开头是问号（ ? ），也就是说，问号只能出现查询参数的开头，出现在其他位置就是非法的，会导致网址解析错误。网址的其
他部分如果要使用这些保留字符，必须使用它们的转义形式。

URL 字符转义的方法是，在这些字符的十六进制 ASCII 码前面加上百分号（ %）。下面是这 18
个字符及其转义形式。

! ： %21

- `#`：%23
- `$`：%24
- `&`：%26
- `'`：%27
- `(`：%28
- `)`：%29
- `*`：%2A
- `+`：%2B
- `,`：%2C
- `/`：%2F
- `:`：%3A
- `;`：%3B
- `=`：%3D
- `?`：%3F
- `@`：%40
- `[`：%5B
- `]`：%5D

举例来说，有一个网页的 URL 是 `foo?bar.html`，即文件里面包含一个问号，那么需要写成 `foo%3Fbar.html`。

URL 的合法字符，其实也可以采用这种转义方法，但是不建议使用。比如，字母 `a` 的十六进制 ASCII 码是 `61`，转义形式后就是 `%61`。因此，`www.apple.com` 又可以写成 `www.%61pple.com`，浏览器一样识别。

值得注意的是，空格的转义形式是 `%20`。对于那些包含空格的文件名，这个转义是必须的。

既不属于合法字符、也不属于保留字符的其他字符（比如汉字），理论上不需要手动转义，可以直接写在 URL 里面，比如 `www.example.com/中国.html`，浏览器会自动将它们转义，发给服务器。转义方法是使用这些字符的十六进制 UTF-8 编码，每两位算作一组，然后每组头部添加百分号（`%`）。

举例来说，汉字 `中` 的 UTF-8 十六进制编码是 `e4b8ad`，每两个字符一组，URL 转义后就为 `%e4%b8%ad`。也就是说，URL 里面凡是有汉字 `中` 的地方，都要写成 `%e4%b8%ad`。因此，访问 `www.example.com/中国.html` 这个网址，需要写成下面的样子。

```
www.example.com/%e4%b8%ad%e5%9b%bd.html
```

上面代码中，`中` 的转义形式是 `%e4%b8%ad`，`国` 是 `%e5%9b%bd`。

绝对 URL 和相对 URL

URL 分成两种：绝对 URL 和相对 URL。

绝对 URL 指的是，只靠 URL 本身就能确定资源的位置。这意味着，URL 必须带有资源的完整信息，包含协议、主机、路径等部分。前面的例子都是绝对 URL。

相对 URL 指的是，不包含资源位置的全部信息，必须结合当前网页的位置，才能定位资源。比如，当前网页的 URL 是 `https://www.example.com/path/index.html`，该网页上面有一个资源，URL 指向 `a.html`，这个就是相对 URL。因为只知道 `a.html`，并不能定位资源。浏览器假定，`a.html` 与当前网址在同一个子目录下面，从而得到绝对 URL `https://www.example.com/path/a.html`。

相对 URL 如果以斜杠（`/`）开头，就表示网站的根目录。否则，必须以当前目录为起点，推算资源的位置。比如，相对 URL `/foo/bar.html` 表示网站根目录的子目录 `foo`，`foo/bar.html` 表示在当前目录的 `foo` 子目录。

URL 还可以使用两个特殊简写，表示特定位置。

- `.`：表示当前目录，比如 `./a.html`（当前目录下的 `a.html` 文件）
- `..`：表示上级目录，比如 `../a.html`（上级目录下的 `a.html` 文件）

这两种简写可以多个连用，比如 `../../` 表示上两级目录。

绝对 URL 也可以使用这两个简写，比如 `www.example.com/./index.html` 等同于 `www.example.com/index.html`，这时 `.` 相当于根目录的当前目录，即根目录本身。

<base>

`<base>` 标签指定网页内部的所有相对 URL 的计算基准。整张网页只能有一个 `<base>` 标签，而且只能放在 `<head>` 里面。它是单独使用的标签，没有闭合标签，下面是一个例子。

```
<head>
<base href="https://www.example.com/files/" target="_blank">
</head>
```

`<base>` 标签的 `href` 属性给出计算的基准网址，`target` 属性给出如何打开链接的说明（参见《链接》一章）。已知计算基准是 `https://www.example.com/files/`，那么相对 URL `foo.html`，就可以转成绝对 URL `https://www.example.com/files/foo.html`。

注意，`<base>` 标签必须至少具有 `href` 属性或 `target` 属性之一。

```
<base href="http://foo.com/app/">
<base target="_blank">
```

一旦设置了 `<base>`，就对整个网页都有效。如果要改变某个链接的行为，只能用绝对链接替代相对链接。尤其需要注意锚点，这时锚点也是针对 `<base>` 计算的，而不是针对当前网页的 URL。

版本号 #1
由 壳壳虫 创建于 2019-05-05 23:20:04 UTC
由 壳壳虫 更新于 2021-08-27 16:29:46 UTC