

WEB 请求和服务端响应对象

主讲人：和凌志

主要内容

- URL 的组成
- WEB 请求对象
- 服务器响应对象

URL 的组成

http:// https://	localhost www.baidu.com	:3000 :8080	/about /search	?test=1 ?name=a&age=18	#head #footer
协议	主机名	端口	路径	查询字符串	信息片段

URL 的组成

http:// https://	localhost www.baidu.com	:3000 :8080	/about /search	?test=1 ?name=a&age=18	#head #footer
协议	主机名	端口	路径	查询字符串	信息片段

- 协议
- 协议确定如何传输WEB（客户端）请求
- 主要是处理 http 和 https。其他常见的协议还有 file 和 ftp。

URL 的组成

http://	localhost	:3000	/about	?test=1	#head
https://	www.baidu.com	:8080	/search	?name=a&age=18	#footer
协议	主机名	端口	路径	查询字符串	信息片段

- **主机名**：主机名标识服务器
- 运行在本地计算机(localhost)，或用IP 地址表示
- 运行在 Internet 环境下，主机名通常以一个顶级域名结尾，比如 .com

URL 的组成

http://	localhost	:3000	/about	?test=1	#head
https://	www.baidu.com	:8080	/search	?name=a&age=18	#footer
协议	主机名	端口	路径	查询字符串	信息片段

- **端口**：每一台服务器都有一系列端口号。特殊端口： 80 和 443 端口。
- 如果省略端口值，那么默认 80 端口负责 HTTP 传输，443 端口负责 HTTPS 传输。
如果不使用 80 和 443 端口，就需要一个大于 1023 的端口号。通常使用容易记忆的端口号，如 3000、8080 或 8088。

URL 的组成

http://	localhost	:3000	/about	?test=1	#head
https://	www.baidu.com	:8080	/search	?name=a&age=18	#footer
协议	主机名	端口	路径	查询字符串	信息片段

- **路径**：URL 中影响应用程序的第一个组成部分通常是路径
- 路径是应用中的页面或其他资源的唯一标识。

URL 的组成

http://	localhost	:3000	/about	?test=1	#head
https://	www.baidu.com	:8080	/search	?name=a&age=18	#footer
协议	主机名	端口	路径	查询字符串	信息片段

- **查询字符串**，是一种键值对集合，是可选的。它以问号(?)开头，键值对以(&) 分隔开
- 所有的名称和值都必须是 URL 编码的

URL 的组成

http://	localhost	:3000	/about	?test=1	#head
https://	www.baidu.com	:8080	/search	?name=a&age=18	#footer
协议	主机名	端口	路径	查询字符串	信息片段

- **信息片段**：被严格限制在浏览器中使用，不会传递到服务器
- 控制单页面应用（SPA）或 AJAX 富应用，会用到它
- 最初，信息片段只是用来让浏览器展现网页中锚点标记的部分



HTTP 请求对象

HTTP 请求机制

- HTTP协议确定了客户端与服务器通信的请求方法集合(通常称为 HTTP verbs)。
- 在浏览器中键入一个URL(或点击一个链接)，服务器会接收到一个 HTTP 请求，其中的重要信息是 URL 路径和查询字符串。至于如何响应，则需要应用程序结合方法、路 径和查询字符串来决定。

HTTP 请求体

- 除请求报头外，请求还有一个主体
- 一般 GET 请求没有主体内容，但 POST 请求是有的
- POST 请求体最常见的媒体类型是 `application/x-www-form-urlencoded`，是键值对集合的简单编码，用 & 分隔

HTTP 请求对象

- HTTP 请求对象，通常传递到回调方法的参数中，可以随意命名，通常命名为 req 或 request)

HTTP 请求对象的属性和方法

- req.params

一个数组，包括路由参数

- req.param(name)

返回命名的路由参数，或者 GET 请求或 POST 请求参数。

HTTP请求对象的属性和方法

- req.query

一个对象，包含以键值对存放的查询字符串参数(通常称为 GET 请求参数)

- req.body

一个对象，包含 POST 请求参数。这样命名是因为 POST 请求参数在 REQUEST 正文中传递，而不像查询字符串在 URL 中传递。

要使 req.body 可用，需要中间件能够解析请求的正文内容类型，

HTTP 请求对象的属性和方法

- `req.route`

关于当前匹配路由的信息，主要用于路由调试。

- `req.cookies`

一个对象，包含从客户端传递过来的 cookies 值

- `req.headers` 从客户端接收到的请求报头

- `req.ip` 客户端的 IP 地址

- `req.path` 请求路径



HTTP 响应对象

HTTP 响应对象

- HTTP 响应对象，通常传递到回调方法的参数中，可以随意命名，常命名为 response 、 res 、 resp)

HTTP 响应对象常用属性和方法

- `res.status(code)`

设置 HTTP 状态代码。默认为 200(成功)

- 返回状态 404(页面不存在)或 500(服务器内部错误)，或任何一个其他的状态码。
- 对于重定向(状态码 301、302、303 和 307)，可用方法 `:redirect`。

HTTP 响应对象常用属性和方法

- `res.set(name,value)`

设置响应头。这通常不需要手动设置

- `res.cookie(name,value,[options])`

- `res.clearCookie(name,[options])`

设置或清除客户端 cookies 值。需要中间件支持

- `res.redirect([status],url)` 重定向浏览器

HTTP 响应对象常用属性和方法

- `res.send(body)`, `res.send(status,body)`
向客户端发送响应及可选的状态码
- 默认内容类型是 `text/html`
- 如果想改为 `text/plain`, 需要在 `res.send` 之前调用
`res.set('Content-Type','text/plain')`

HTTP 响应对象常用属性和方法

- `res.json(json), res.json(status,json)`

向客户端发送 JSON 以及可选的状态码。

- `res.jsonp(json), req.jsonp(status,json)`

向客户端发送 JSONP 及可选的状态码。

HTTP 响应对象常用属性和方法

- `res.sendFile(path,[option],[callback])`
- 这个方法可根据路径，读取指定文件，并将内容发送到客户端
- 具体操作方法： 使用静态中间件，并将发送到客户端的文件放在公共目录下

小结

- ✓ 对 HTTP 请求和响应对象作了概述。
- ✓ 大部分时候，我们只需用到其中一小部分