Kentico

HTTP Introduction

Slavomír Moroz
2015

Topics

- HTTP methods
- Query string
- HTTP statuses
- HTTP headers
- Cookies
- Session
- HTTPS
- REST services
- Developer tools

HTTP

- HTTP works as a request-response protocol between a client and server.
- A web browser may be the client, and an application on a computer that hosts a web site may be the server.

Request format:

- A Request-line (request method, URI & protocol version)
- Zero or more header
- An empty line indicating the end of the header fields
- Optionally a message-body

HTTP Request Methods: GET and POST

- Two commonly used methods for a request-response between a client and server are: GET and POST.
- **GET** - Requests data from a specified resource
- **POST** - Submits data to be processed to a specified resource

http://www.w3schools.com/tags/ref_httpmethods.asp - **Compare GET vs. POST**

GET method

The GET method is used to retrieve information from the given server using a given URI.

Requests using GET should only retrieve data and should have no other effect on the data.

```
GET /hello.htm HTTP/1.1
Host: www.tutorialspoint.com
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

Post method

A POST request is used to send data to the server, for example, customer information, file upload, etc. **using HTML forms.**

```
POST /cgi-bin/process.cgi HTTP/1.1
Host: www.tutorialspoint.com
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Content-Type: application/x-www-form-urlencoded
Content-Length: length
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

[BODY]
```

Query string

- Contains data sent to server.
- The query string can be sent to the server using either HTTP GET or POST request method.

```
GET /test/demo_form.asp?name1=value1&name2=value2
```

```
POST /test/demo_form.asp HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2
```

URL encoding (also known as Percent-encoding)

- used to deal with special characters that cannot be part of the URL (GET method).
- encoding of POST data is determined based on "Content-Type" header.

```
?first=this+is+a+field&second=was+it+clear+%28already%29%3F
```

https://en.wikipedia.org/wiki/Query_string#URL_encoding

HTTP response statuses

- Each HTTP response contains completion status information.
- First line of the HTTP response is called the *status line* and includes a numeric *status code* (such as "404") and a textual *reason phrase* (such as "Not Found").

Status categories:

- 1xx Informational
- 2xx Sucess
- 3xx Redirection
- 4xx Client error
- 5xx Server error

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

HTTP headers

- **HTTP header fields** are components of the header section of request and response messages in the Hypertext Transfer Protocol (HTTP). They define the operating parameters of an HTTP transaction.

```
Host: en.wikipedia.org
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Accept: text/html
Accept-Language: en-US
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: 348
Connection: Keep-Alive
```

https://en.wikipedia.org/wiki/List_of_HTTP_header_fields#Request_fields
https://en.wikipedia.org/wiki/List_of_HTTP_header_fields#Response_fields

REST

- REST (REpresentational State Transfer) is an architectural style, and an approach to communications that is often used in the development of Web services.
- Alternative to SOAP.

- HTTP verbs tell the server what to do with the data identified by the URL.
- The most important verbs for building RESTful API are GET, POST, PUT and DELETE.

`GET /clients HTTP/1.1` – receives a list of clients
`GET /clients/anne HTTP/1.1` – receives detailed information about client with identifier anne
`DELETE /clients/anne HTTP/1.1` – deletes client with identifier anne
`PUT /clients/robin HTTP/1.1` – creates new client with identifier robin (client data are included in the request body)

http://code.tutsplus.com/tutorials/a-beginners-guide-to-http-and-rest--net-16340
Api example: https://developers.google.com/drive/v1/reference/

HTTP session state

- HTTP is a stateless protocol. A stateless protocol does not require the HTTP server to retain information or status about each user for the duration of multiple requests.

- Web applications implement states or server side sessions using for instance HTTP cookies or Hidden variables within html forms.

HTTP cookie

- Is a small piece of data sent from a website and stored in a user's web browser while the user is browsing that website.
- Every time the user loads the website, the browser sends the cookie back to the server to notify the website of the user's previous activity.
- A cookie consists of the following components: name, value and zero or more attributes.

Server response setting cookies

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: theme=light
Set-Cookie: sessionToken=abc123; Expires=Wed, 09 Jun 2021 10:18:14 GMT
```

Browser request providing previously stored cookies

```
GET /spec.html HTTP/1.1
Host: www.example.org
Cookie: theme=light; sessionToken=abc123
```

https://en.wikipedia.org/wiki/HTTP_cookie

Web session

- Server side feature.
- In general session is a data container containing data that were used during previous requests of the same user.

ASP .NET session

- ASP.NET session state identifies requests from the same browser during a limited time window as a session, and provides a way to persist variable values for the duration of that session.
- By default, the session identifier is stored in a non-expiring session cookie in the browser.

http://machinesaredigging.com/2013/10/29/how-does-a-web-session-work/
https://msdn.microsoft.com/en-us/library/ms178581.aspx

HTTPS

- HTTP within encrypted connection
- Uses symmetric encryption.
- Secure connection is handled by web server, no special actions are required from web developer.
- Web server must be configured to allow HTTS requests. An encryption certificate must be installed on the server.
- If certificate is not signed by trusted authority, connection is considered as dangerous and client may refuse this connection (common problem in development environments).
- Default port of HTTP protocol is 80, for HTTPS it's 443.

https://blog.hartleybrody.com/https-certificates/
https://technet.microsoft.com/en-us/library/cc753127(v=ws.10).aspx
https://technet.microsoft.com/en-us/library/cc754841.aspx

Resources & tools

HTTP

- https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html
- http://www.tutorialspoint.com/http/index.htm

Tools

- http://getfirebug.com/
- http://www.telerik.com/fiddler