Web Technologies

INFO310

URLs


http://www.google.com/search?q=facebook#result
protocol
domain
path
parameters
fragment

SEO Cheat Sheet: Anatomy of A URL

seomoz.org
Read SEOmoz. Rank Better

1 SEO-FRIENDLY URL

1. Protocol
2. Subdomain
3. Domain
4. Top-Level Domain
5. Folders / Paths
6. Page
7. Named Anchor

Keyword Priority[1]

Observed Google priority of keyword placement:

(1) Domain
(2) Subdomain
(3) Folder
(4) Path/Page

SEO Tips for URLs

- Use subdomains carefully. They may be treated as separate entities, splitting domain authority.
- Separate path & page keywords with hyphens ("-").
- Anchors may help engines understand page structure.
- Keyword effectiveness in URLs decreases as URL length and keyword position increases.[1]

[1] SEOmoz correlational data (2009)

2 OLD DYNAMIC URL

1. Protocol
2. Subdomain
3. Domain
4. Top-Level Domain
5. Page / File Name
6. File Extension
7. CGI Parameters

Popular TLDs[2]

.com - commercial
.net - infrastructure
.org - non-profit
.edu - schools
.info - informational
.biz - small business
.name - personal sites

Popular ccTLDs*

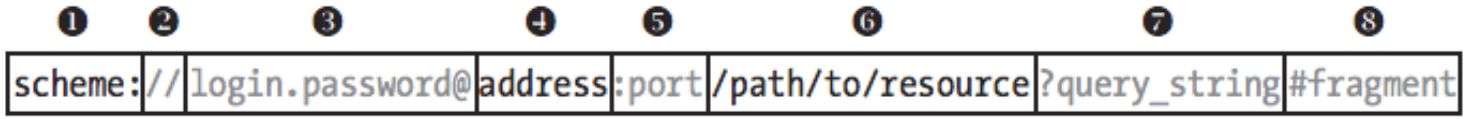
.cn - China
.de - Germany
.uk - United Kingdom
.nl - Netherlands
.eu - European Union
.ru - Russian Federation
.ar - Argentina

Popular Extensions

.htm - Static HTML
.html - Static HTML
.php - PHP code
.asp - ASP code
.aspx - ASP.NET
.cfm - ColdFusion
.jsp - Java Code

[2] Verisign domain report (2009)

* ccTLD = Country Code TLD



❶ Scheme/protocol name

❷ Indicator of a hierarchical URL (constant)

❸ Credentials to access the resource (optional)

❹ Server to retrieve the data from

❺ Port number to connect to (optional)

"Authority"

❻ Hierarchical Unix path to a resource

❼ "Query string" parameters (optional)

❽ "Fragment identifier" (optional)

URL Characters

- Unreserved
  - The alphanumerical upper and lower case character may optionally be encoded:

    A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
    a b c d e f g h i j k l m n o p q r s t u v w x y z
    0 1 2 3 4 5 6 7 8 9 - _ . ~

- Reserved
  - Special symbols must sometimes be percent-encoded:

    ! * ' ( ) ; : @ & = + $, / ? % # [ ]

- Further details can for example be found in
  - RFC 3986
  - http://www.w3.org/Addressing/URL/uri-spec.html

- Source: https://en.wikipedia.org/wiki/Uniform_resource_locator

URL Schemes

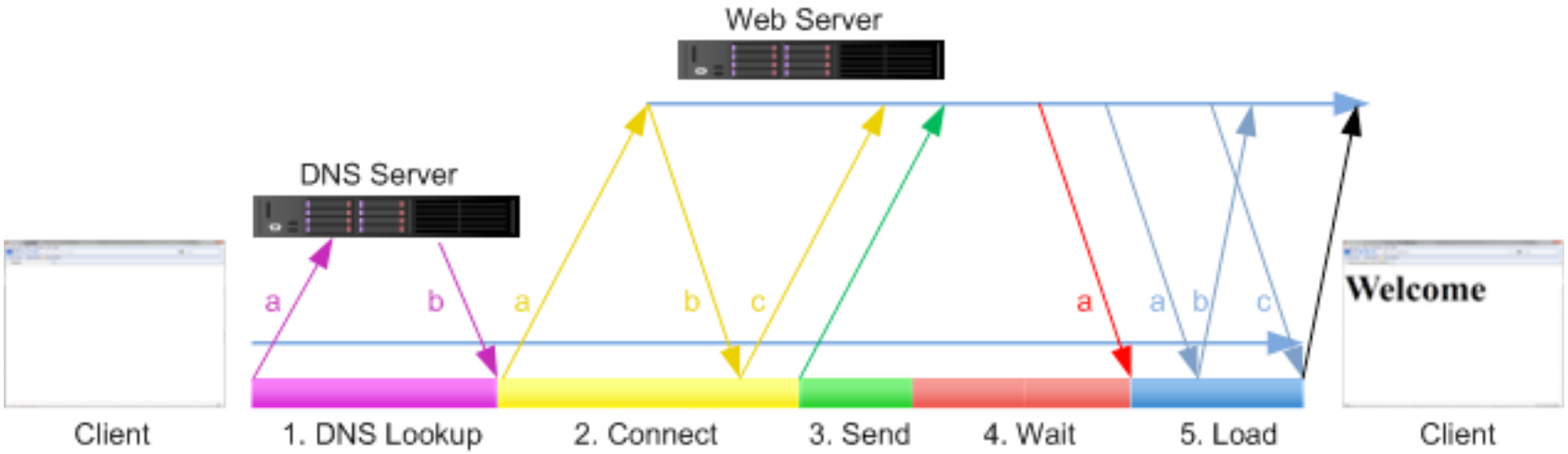
- Tons of supported schemes
  - https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml
- Supporting these can lead so some weirdness
- Common ones you may see:
  - file://
  - ftp://
  - http://
  - https://
  - mailto://
  - sms://

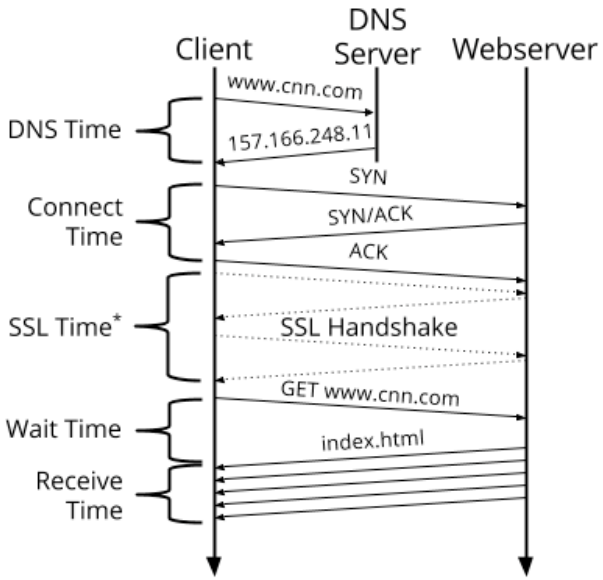
Things can get weird

- http://127.0.0.1/
  - This is a canonical representation of an IPv4 address.
- http://0x7f.1/
  - This is a representation of the same address that uses a hexadecimal number to represent the first octet and concatenates all the remaining octets into a single decimal value.
- http://017700000001/
  - The same address is denoted using a 0-prefixed octal value, with all octets concatenated into a single 32-bit integer.
- http://example.com&gibberish=1234@167772161/
  - Where do you think this goes?
- http://example.com\@coredump.cx/
  - How about this one?
- http://example.com;.coredump.cx/
  - And this?

- Source: Tangled Web by Michal Zalewski (pages 26 and 30)

Browser Requests





HTTP Requests

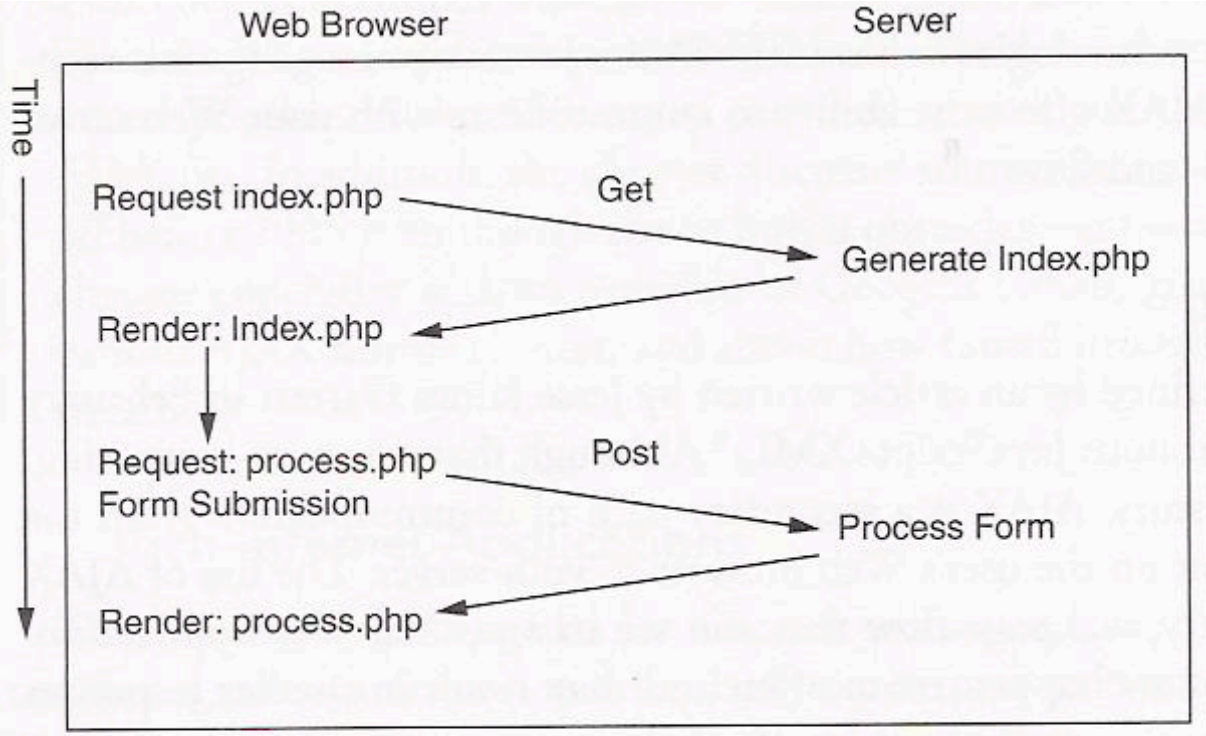


FIGURE 1-1
Web application request flow

HTTP Request/Response

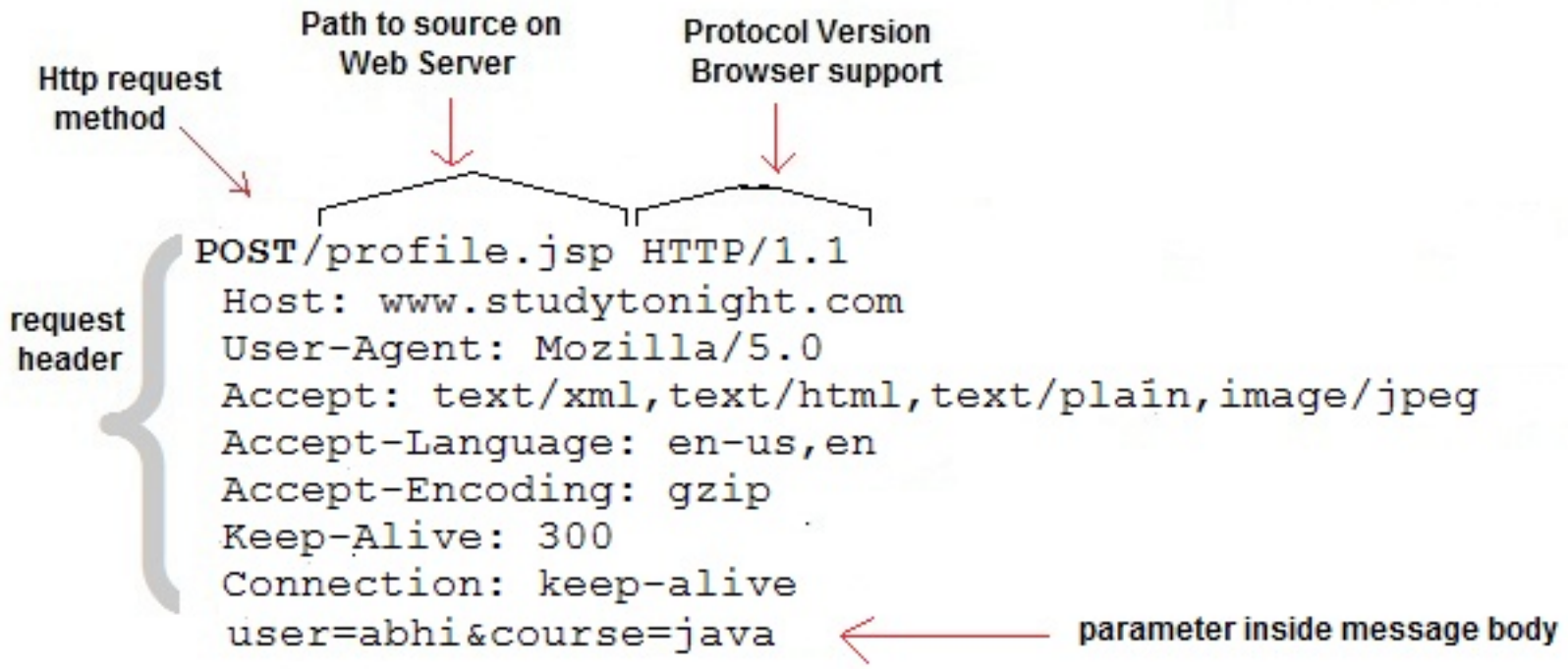
```
POST /fuzzy_bunnies/bunny_dispenser.php
HTTP/1.1
Host: www.fuzzybunnies.com
User-Agent: Bunny-Browser/1.7
Content-Type: text/plain
Content-Length: 17
Referer:
http://www.fuzzybunnies.com/main.html
I REQUEST A BUNNY
```

```
HTTP/1.1 200 OK
Server: Bunny-Server/0.9.2
Content-Type: text/plain
Connection: close
BUNNY WISH HAS BEEN GRANTED
```

GET Request

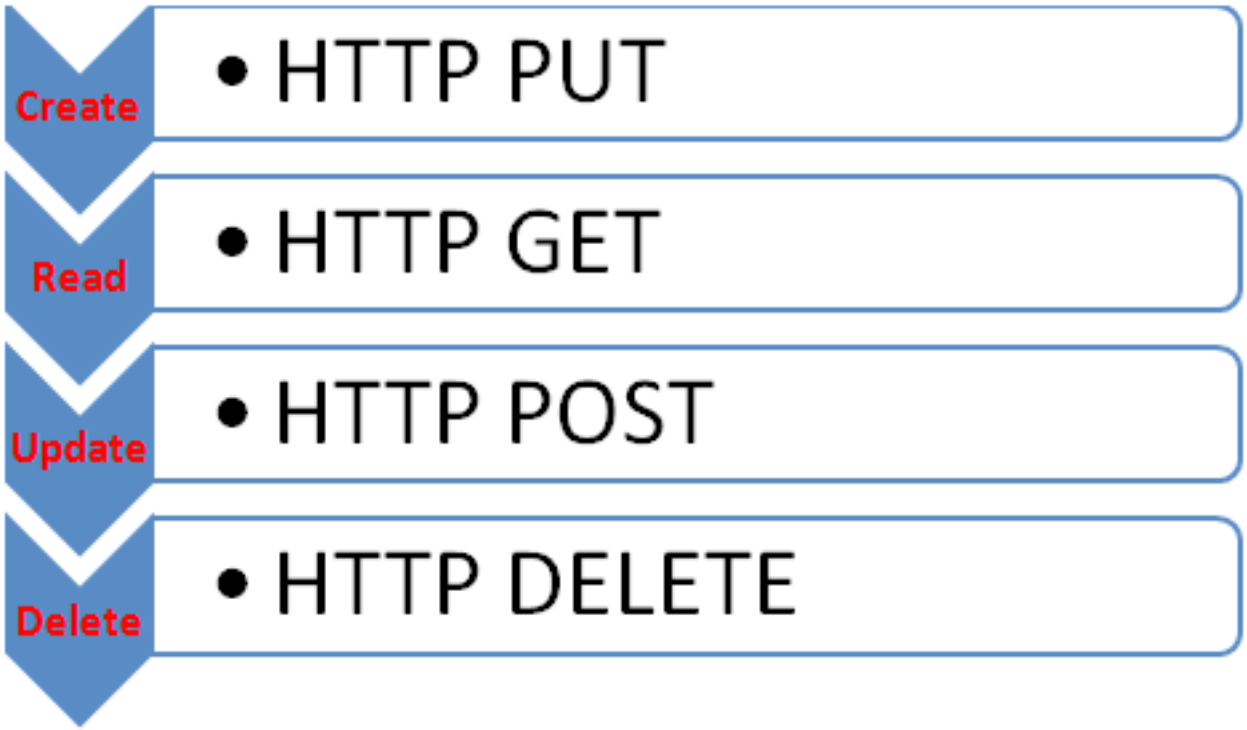

Http request method
Path to source on Web Server
Parameters
Protocol Version Browser support
request header
GET/profile.jsp?user=abhi&course=java HTTP/1.1
Host: www.studytonight.com
User-Agent: Mozilla/5.0
Accept: text/xml,text/html,text/plain,image/jpeg
Accept-Language: en-us,en
Accept-Encoding: gzip
Keep-Alive: 300
Connection: keep-alive

POST Request



HTTP Methods

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options


Create
• HTTP PUT
Read
• HTTP GET
Update
• HTTP POST
Delete
• HTTP DELETE

HTTP Headers

- Define the operating parameters of the HTTP transaction
- There are tons “official” ones:
  - https://en.wikipedia.org/wiki/List_of_HTTP_header_fields
- Colon separated
- Ultimately they can be whatever you want
- No limit on size of name or value

Cookies

- A small bit of data sent by a web server to a browser that is stored by the browser and sent back with subsequent requests
- Designed to provide a storage mechanism for stateful information and record a user's browsing activity
- Structure
  - Name
  - Value
  - 0+ attributes

Cookie Attributes

- Domain and Path
  - Defines scope of cookie
- Expires and Max-age
  - Defines when the browser should delete the cookie
- Secure
  - Directs the browser on whether or not to send the cookie over encrypted connection only or not
- HttpOnly
  - Directs the browser on JavaScripts access to the cookie

Cookies

GET /index.html HTTP/1.1
Host: www.example.org
...

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
...

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