



UNIVERSITÀ
DEGLI STUDI
DI PALERMO



Il World Wide Web e il protocollo HTTP

CORSO DI PROGRAMMAZIONE WEB E MOBILE

A.A. 2021/2022

PROF. ROBERTO PIRRONE

Sommario

- Generalità sul web
- Ipertesti e architettura dell'informazione
- Concetto di URL e funzionamento del servizio web
- Il protocollo HTTP

Il World Wide Web

- Il World Wide Web (WWW) è un servizio di consultazione di documenti organizzati secondo una struttura ipertestuale distribuita sulla rete Internet.
- Il WWW è sicuramente il componente più visibile di Internet, tanto che si tende ad identificarlo con quest'ultima.

Il World Wide Web

- Si possono distinguere due grandi usi della tecnologia web:
 - Come strumento ipertestuale di comunicazione (in forma nativa)
 - Come strumento di comodo interfacciamento di altri servizi di rete (accesso a banche dati, e-commerce, sistemi informativi aziendali, etc.)
- Nella seconda forma la tecnologia del web viene usata anche in ambienti “chiusi”, come le reti aziendali, cioè non direttamente collegati ad internet. In questa accezione nasce il concetto di [Intranet](#).

Cos'è un sito web

- Un sito web può essere definito come un particolare tipo di *ipertesto*.
 - **Ipertesto**: organizzazione di materiale informativo (in genere multimediale) in forma tale da consentirne una fruizione non lineare in cui sono presenti dei percorsi di fruizione predefiniti dal progettista secondo un preciso progetto comunicativo.

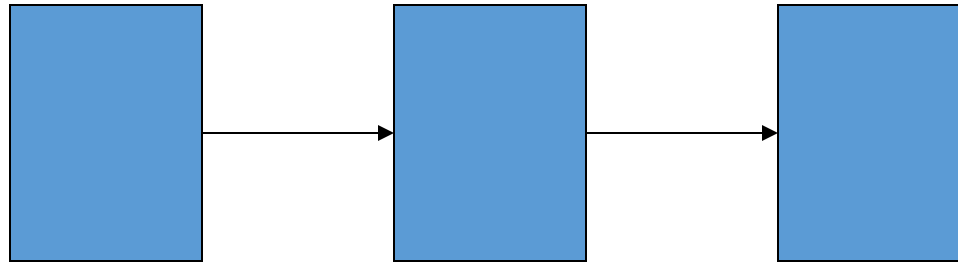
Elementi di un ipertesto

- In un qualunque ipertesto (e quindi anche nel web) si possono individuare i seguenti elementi fondamentali:
- **Nodo**: unità informativa autosufficiente, testuale e non; in genere in un nodo sono presenti più elementi eterogenei (testo, grafica, filmati etc.)
- **Ancora**: punto di origine o di termine di un collegamento ad un'altra unità.
- **Link**: collegamento ad un'altra unità; può avere diverse funzioni (salto, riferimento, comando, nota esplicativa etc.) ed è agganciato a due ancore.

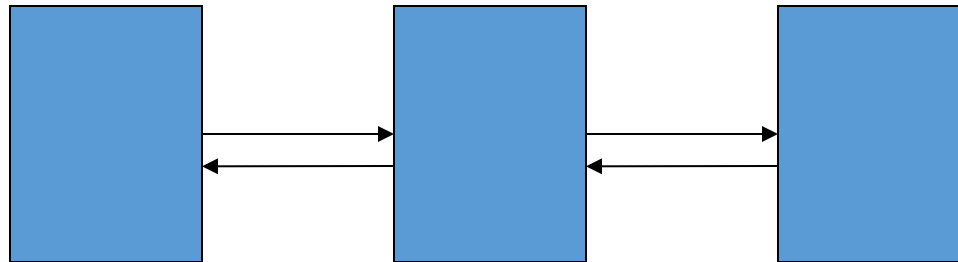
Architettura dell'Informazione

- L'Architettura dell'informazione di un sito web consiste nel suo **design strutturale** atto a facilitare l'esecuzione dei compiti e l'accesso intuitivo ai contenuti.
- Si può dire che l'Architettura dell'informazione sia la combinazione di **organizzazione, categorizzazione e schemi di navigazione** dei contenuti all'interno di un sito.

Architettura dell'informazione

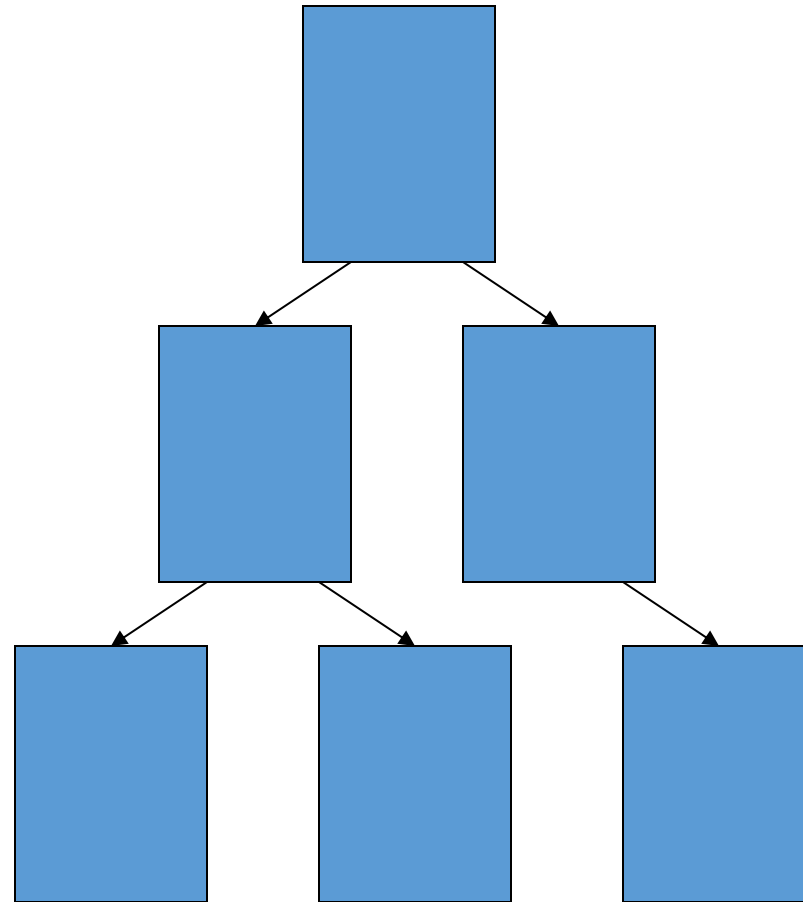


Lista



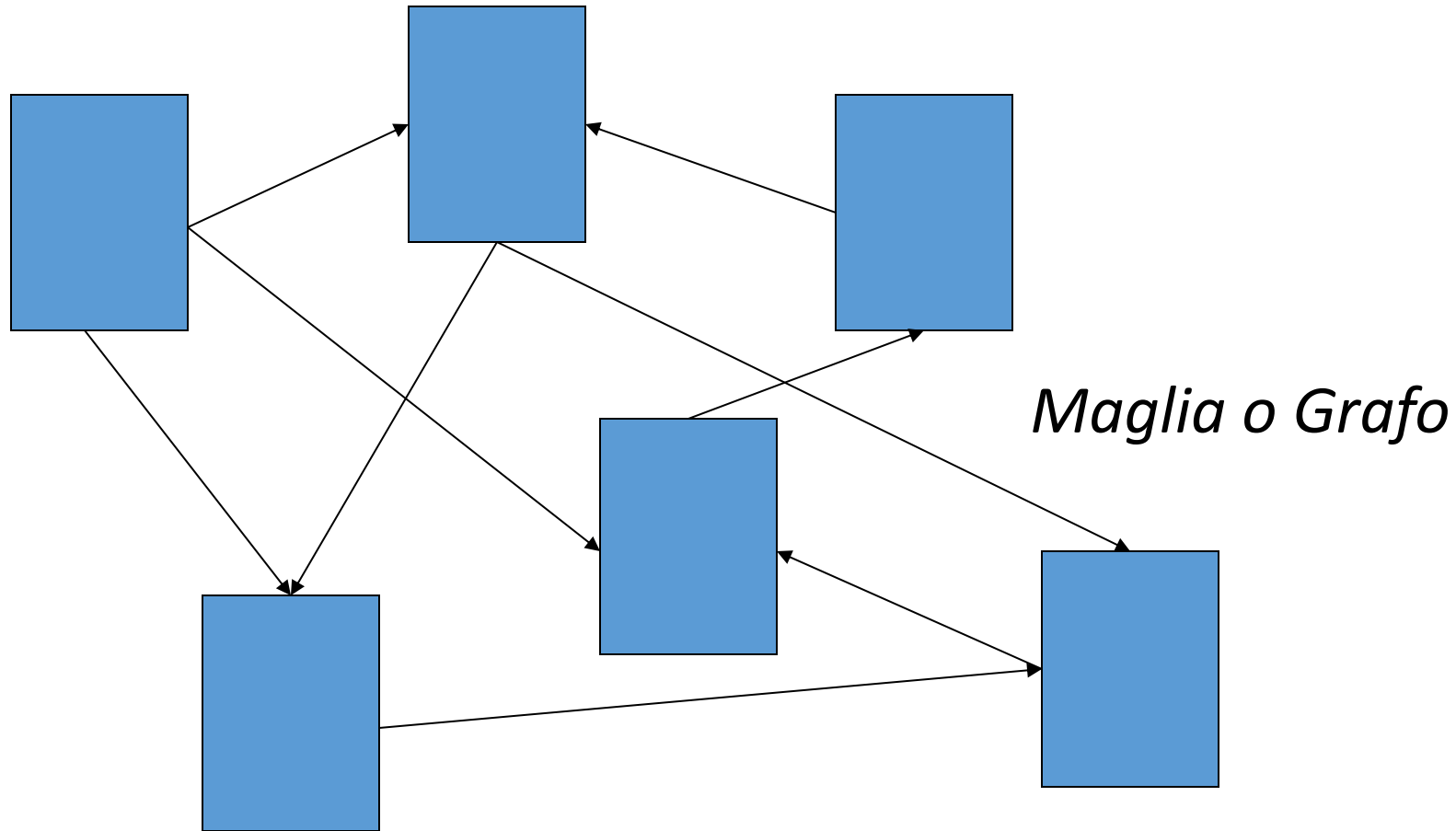
Lista doppia

Architettura dell'informazione



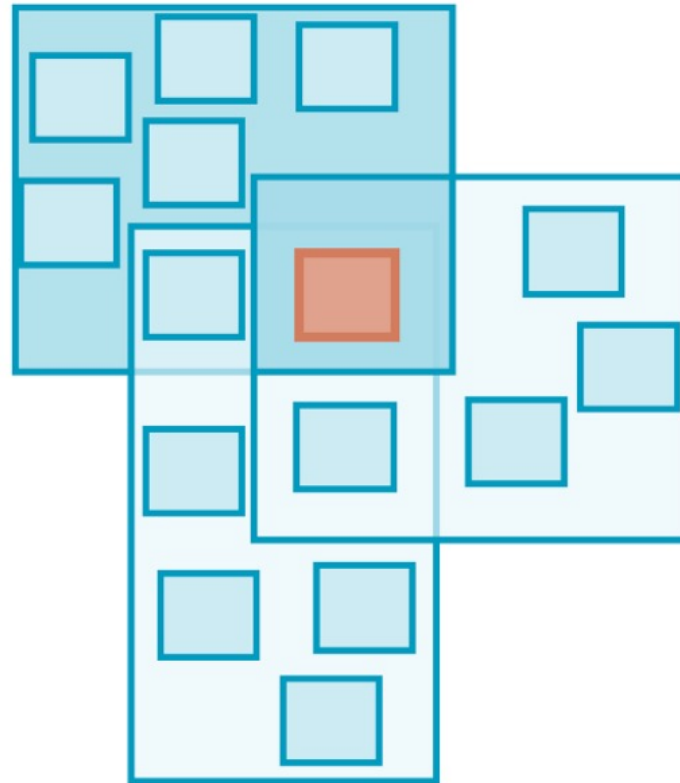
Albero

Architettura dell'informazione



Architettura dell'informazione

A faccette



Ogni item informativo è caratterizzato da un insieme di attributi (sfaccettature o faccette) che hanno ciascuno il proprio valore.

In fase di ricerca, per una o più faccette, si specifica il valore/intervallo richiesto creando così un *sottoinsieme di item*.

Quegli item che si trovano *all'intersezione di tutti i sottoinsiemi* sono il risultato della ricerca e vengono mostrati, in genere, con una struttura di tipo lineare.

Architettura dell'Informazione e navigazione

- La struttura del sito è strettamente connessa alla navigazione: l'una non può prescindere dall'altra
- Esistono diversi tipi di strumenti di navigazione in una pagina con posizioni tipiche ben precise all'interno del layout
 - Navigazione principale
 - Navigazione secondaria
 - Navigazione contestuale
 - Meta-navigazione

Tipi di navigazione

Navigazione Principale

Navigazione Secondaria

Navigazione contestuale

Meta-navigazione

The screenshot shows the profile page of Roberto Pirrone on the Unipa website. The page is divided into several sections: a header with navigation links, a profile section with a photo and basic information, a biographical note, and a bibliography. Colored lines are drawn over the page to illustrate different navigation types:

- Blue line:** Points to the main navigation bar at the top, labeled "Navigazione Principale".
- Yellow line:** Points to the secondary navigation bar below the header, labeled "Navigazione Secondaria".
- Red lines:** Point to the profile section, the biographical note, and the bibliography, collectively labeled "Navigazione contestuale".
- Green lines:** Point to the "Orari di ricevimento" and "Contatti" sections, labeled "Meta-navigazione".

Profile Section:

ROBERTO PIRRONE
PROFESSORE ASSOCIATO (ING-INF/05)
Ingegneria

Orari di ricevimento
Martedì dalle 10:30 alle 12:30 presso Edificio 6, III piano stanza 8
Accedi al portale studenti per prenotare il ricevimento

Contatti
+3909123862625
roberto.pirrone@unipa.it

Nota biografica

Ascogli
Roberto Pirrone è Professore Associato di Sistemi di Elaborazione delle Informazioni (ING-INF/05) presso il Dipartimento dell'Innovazione Industriale e Digitale (DIID) dell'Università degli Studi di Palermo.

Insegna *Comunicazione Digitale* presso il Corso di Laurea in Scienze della Comunicazione per i Media e le Istituzioni, *Programmazione Web e Mobile* presso il Corso di Laurea in Ingegneria Informatica, *Informatica Grafica* presso il Corso di Laurea Magistrale in Ingegneria Informatica e *Web Design e Programming* presso il Corso di Laurea Magistrale in Teorie della Comunicazione. E' componente del Collegio dei Docenti del Dottorato in Ingegneria dell'Innovazione Tecnologica, attivo presso il DIID ed è Coordinatore Vicario del Coordinamento Interclasse dei Corsi di Studio in Ingegneria Informatica e delle Telecomunicazioni. E' stato Delegato del Rettore per le Attività Informatiche e per il Portale di Ateneo.

Dirige il Laboratorio di Interazione Uomo-Macchina (CHI-Lab) presso il DIID. I principali interessi di ricerca ricadono nel campo dell'interazione uomo-macchina in linguaggio naturale e le interfacce utente adattive, con applicazioni nel campo dell'e-learning e dei beni culturali, la grafica tridimensionale nonch edacut; gli algoritmi e i sistemi di supporto alla diagnostica per immagini. In relazione alla propria attivit  di ricerca, ha coordinato le unit  di ricerca dei progetti finanziati I-TUTOR, nell'ambito del programma europeo Lifelong Learning Programme (LLP) 2007-2013 e DIGITEMA, di cui   anche responsabile scientifico, nell'ambito del Piano Azione e Coesione del PON Ricerca e Competitivit .

Ha organizzato diversi eventi nazionali internazionali nel campo dell'Intelligenza Artificiale,   revisore di numerose riviste scientifiche internazionali ed   coautore di pi  di 150 articoli scientifici di cui pi  di 100 su riviste ed atti di convegni internazionali indicizzati nelle principali banche dati.

Bacheca

Archivio Bacheca

Rinvio della lezione di "Web Design e Programming" di marted  5 dicembre 2017
5-dic-2017
Si informano gli studenti di "Web Design e Programming" che, a causa di una indisposizione del docente, la lezione odierna   rinviata a gioved  7 dicembre p.v. alle ore 12:00 presso l'aula 12. Poich  l'aula 12 non   fornita di PC, si prega quanti possono farlo di presentarsi alla lezione muniti del proprio portatile.
Parole chiave:
web design e programming, orario lezioni, bacheca, teorie della comunicazione

Materiale didattico del corso di Informatica Grafica a.a. 2017-2018
27-set-2017
Al seguente [link Dropbox](#)   possibile scaricare il materiale per gli studenti della laurea magistrale in Ingegneria Informatica.
Parole chiave:
materiale didattico, ingegneria informatica, informatica grafica

Risorse informative: URI, URL, URN

- **URI: *Uniform Resource Identifier***, è una stringa che identifica univocamente una risorsa generica sulla rete.
- Gli URI rendono disponibili le risorse secondo una varietà di protocolli quali HTTP, FTP, ecc.

Risorse informative: URI, URL, URN

- **URN: Uniform Resource Name**, è un URI che identifica una risorsa mediante un "nome" in un particolare dominio di nomi ("namespace") senza nessuna indicazione sulla sua esistenza e/o posizione nella rete.
 - [RFC 8141](#)
 - Sintassi **urn**:<Namespace Identifier>:<Namespace Specific String>
 - Es. **urn:isbn:0-395-36341-1**

Risorse informative: URI, URL, URN

- **URL: Uniform Resource Locator** è un URI che, oltre a identificare una risorsa, fornisce mezzi per agire su o per ottenere una rappresentazione della risorsa descrivendo il suo meccanismo di accesso primario o la sua "ubicazione" ("location") in una rete.
- [RFC1738](#), [RFC1808](#)
- *Ogni singolo elemento di una pagina web è da considerarsi risorsa*

URL

- Una URL ha una sintassi generica del tipo:
<*protocollo*><*indirizzo*><*percorso*>
- <http://www.unipa.it/persone/docenti/p/roberto.pirrone>
- <mailto:roberto.pirrone@unipa.it>
- <ftp://ftp.unipa.it/pub/linux/redhat/>

Tipi di URL

- Richiesta di un documento ipertestuale:

`http://<nome_server>[:<numero_porta>]/<percorso>[/<nome_file_html>][#<ancora>]`

- Numero porta: la porta TCP di erogazione del servizio, se diversa da 80 (default).
- Se il nome del file html manca, si assume index.html, default.html o index.php, altrimenti bisogna specificarlo a priori.
- Ancora: riferimento ad una zona precisa all'interno della pagina (usata per i riferimenti incrociati interni).

Tipi di URL

- Richiesta di trasferimento file sicuro:

`sftp://[<username>[:<password>]@]<nome server>/ <percorso>/<nome file>`

- *Username e password*: identificazione di un utente specifico, altrimenti si assume l'utente *anonymous* senza password.

Tipi di URL

- Richiesta di terminale remoto sicuro:

`ssh://<utente>[:<password>] @<nome server>[:<numero porta>]`

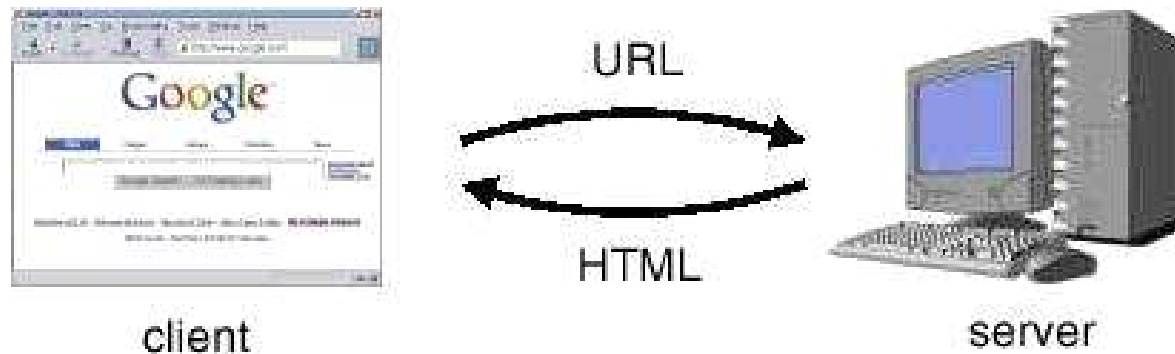
- Richiesta di invio posta elettronica:

`mailto:<nome utente>@<nome server>`

- Spesso il nome del server coincide con quello del dominio: è il DNS del dominio a conoscere l' esatto indirizzo del server)

HTTP

- HTTP: **H**yper-**T**ext **T**ransfer **P**rotocol
 - Protocollo di applicazione – su TCP
 - Modello di comunicazione client-server
 - La comunicazione avviene attraverso coppie richiesta – risposta su stream TCP
 - HTTPRequest, HTTPResponse
 - [RFC7230](#), [RFC7231](#), [RFC7232](#), [RFC7233](#), [RFC7234](#), [RFC7235](#) (HTTP/1.1)
 - [RFC7540](#), [RFC7541](#) (HTTP/2)



HTTP Request

```
GET /search?q=Introduction+to+XML+and+web+Technologies HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.2)
↳ Gecko/20040803
Accept: text/xml,application/xml,application/xhtml+xml,
↳ text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: da,en-us;q=0.8,en;q=0.5,sw;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.google.com/
```

- Linea di richiesta
 - Metodi GET e POST

HTTP Request

```
GET /search?q=Introduction+to+XML+and+Web+Technologies HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.2)
↳ Gecko/20040803
Accept: text/xml,application/xml,application/xhtml+xml,
↳ text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: da,en-us;q=0.8,en;q=0.5,sw;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.google.com/
```

- Linee di intestazione
 - Host
 - Che URL stiamo richiedendo?

HTTP Request

```
GET /search?q=Introduction+to+XML+and+Web+Technologies HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.2)
↳ Gecko/20040803
Accept: text/xml,application/xml,application/xhtml+xml,
↳ text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: da,en-us;q=0.8,en;q=0.5,sw;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.google.com/
```

- Linee di intestazione
 - User-agent
 - Identificazione del browser/S.O./piattaforma software

HTTP Request

```
GET /search?q=Introduction+to+XML+and+Web+Technologies HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.2)
↳ Gecko/20040803
Accept: text/xml,application/xml,application/xhtml+xml,
↳ text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: da,en-us;q=0.8,en;q=0.5,sw;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.google.com/
```

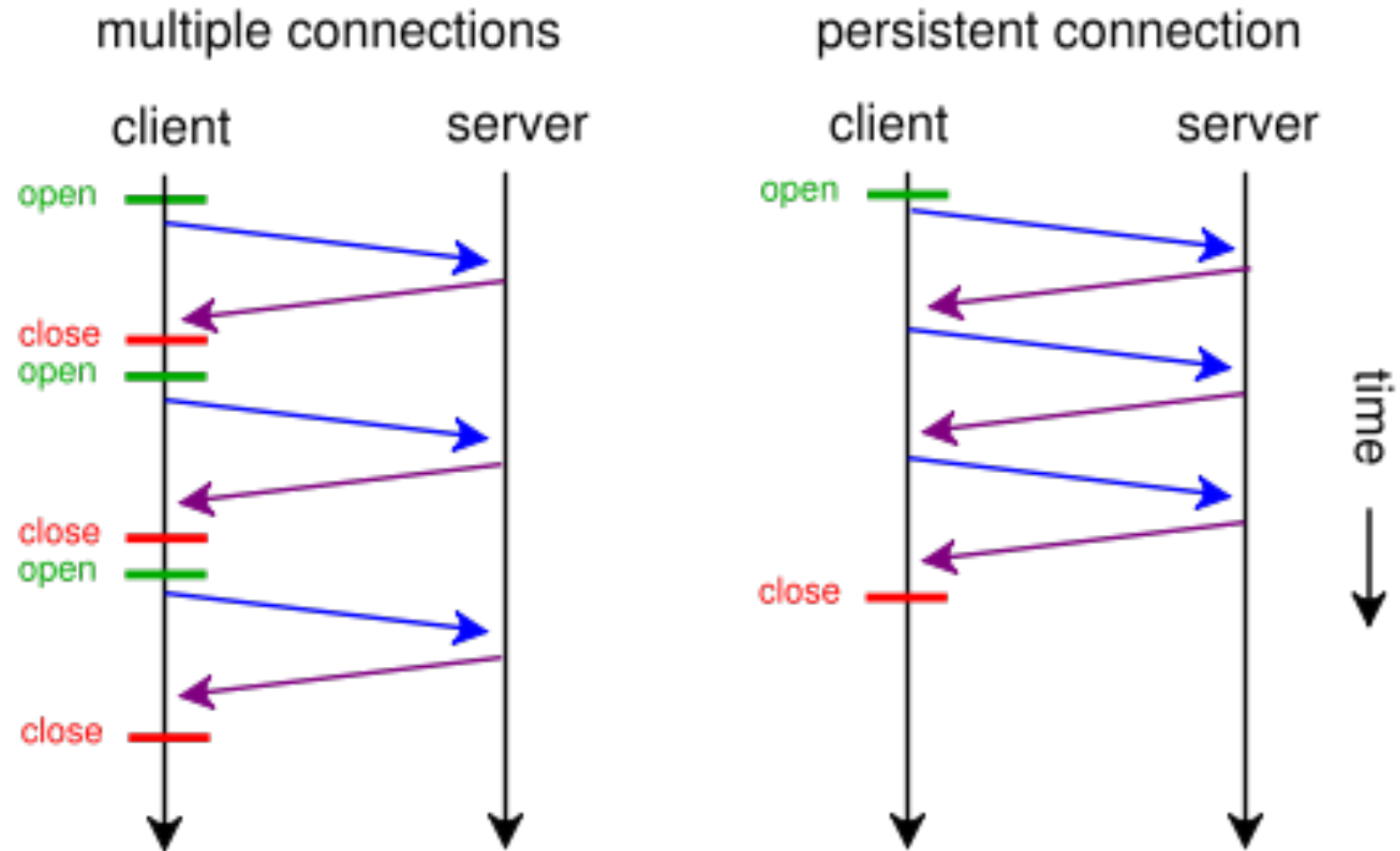
- Linee di intestazione
 - **Accept-...**
 - Formati dei dati, compressione, linguaggio e set dei caratteri accettati dal browser

HTTP Request

```
GET /search?q=Introduction+to+XML+and+Web+Technologies HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.2)
↳ Gecko/20040803
Accept: text/xml,application/xml,application/xhtml+xml,
↳ text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: da,en-us;q=0.8,en;q=0.5,sw;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.google.com/
```

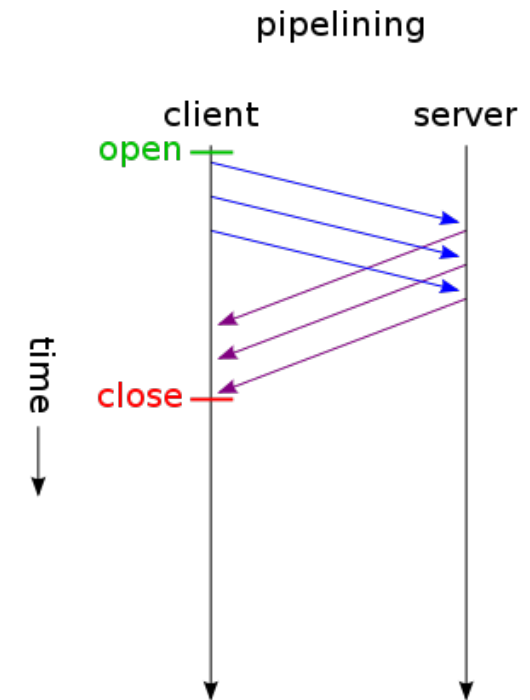
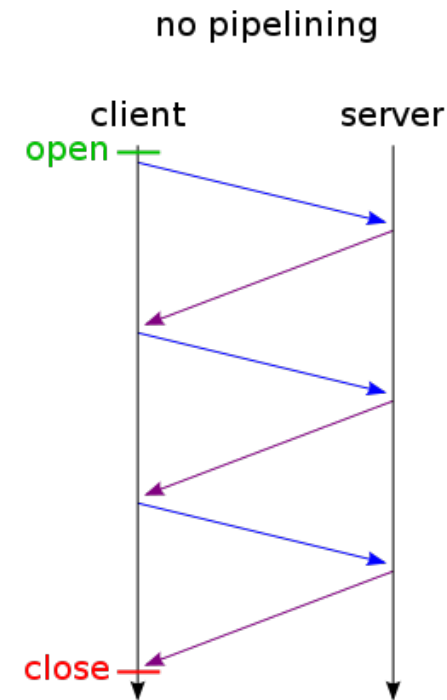
- Linee di intestazione
 - Keep-alive, Connection
 - Tempo di mantenimento della connessione che viene impostata come “persistente” per quel lasso di tempo

Modello di connessione in HTTP



Connessioni persistenti

- Coppie richiesta – risposta multipla sulla stessa connessione TCP
- Pipelining
 - Più richieste partono di seguito prima che vengano ricevute le risposte
 - Tipicamente per le richieste GET



HTTP Request

```
GET /search?q=Introduction+to+XML+and+Web+Technologies HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.2)
↳ Gecko/20040803
Accept: text/xml,application/xml,application/xhtml+xml,
↳ text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: da,en-us;q=0.8,en;q=0.5,sw;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.google.com/
```

- Linee di intestazione
 - **Referer**
 - L'URL da cui il browser sta eseguendo la richiesta cioè l'origine del link ipertestuale

HTTP Response

HTTP/1.1 200 OK

Date: Fri, 17 Sep 2009 07:59:01 GMT

Server: Apache/2.0.50 (Unix) mod_perl/1.99_10 Perl/v5.8.4

↳ mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8 mod_bigwig/2.1-3

Last-Modified: Tue, 24 Feb 2009 08:32:26 GMT

ETag: "ec002-afa-fd67ba80"

Accept-Ranges: bytes

Content-Length: 2810

Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>...</html>

- **Linea di stato**
 - Il codice di stato che informa il client sul soddisfacimento o meno della richiesta e sul motivo

Codici di stato HTTP

- 1xx Informational
 - Richiesta ricevuta, continua l'elaborazione
 - **100** Continue
 - Il server ha ricevuto l'header della richiesta e il client deve procedere ad inviare il corpo della richiesta (richieste di tipo POST)
- 2xx Success
 - **200** OK
 - Risposta standard per le richieste HTTP andate a buon fine

Codici di stato HTTP

- 3xx Redirection
 - Il client deve eseguire ulteriori azioni per soddisfare la richiesta
- 301 Moved Permanently
 - Questa e tutte le future richieste andranno dirette ad un altro URI
- 304 Not Modified
 - Non c'è bisogno di ritrasmettere la risorsa: è una redirezione implicita alla cache

Codici di stato HTTP

- 4xx Client Error
 - La richiesta è sintatticamente scorretta o non può essere soddisfatta
- 400 Bad Request
 - La richiesta non può essere soddisfatta a causa di errori di sintassi
- 401 Unauthorized
- 403 Forbidden
 - La richiesta è legittima ma il server si rifiuta di soddisfarla
- 404 Not Found
 - La risorsa richiesta non è stata trovata

Codici di stato HTTP

- 4xx Client Error
 - La richiesta è sintatticamente scorretta o non può essere soddisfatta
- 405 Method Not Allowed
 - La richiesta è stata eseguita usando un metodo non permesso

Codici di stato HTTP

- 5xx Server Error
 - Il server ha fallito nel soddisfare una richiesta apparentemente valida
- 500 Internal Server Error
 - Messaggio di errore generico senza alcun dettaglio
- 503 Service Unavailable
 - Il server non è al momento disponibile. Generalmente è una condizione temporanea

HTTP Response

```
HTTP/1.1 200 OK
Date: Fri, 17 Sep 2009 07:59:01 GMT
Server: Apache/2.0.50 (Unix) mod_perl/1.99_10 Perl/v5.8.4
      mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8 mod_bigwig/2.1-3
Last-Modified: Tue, 24 Feb 2009 08:32:26 GMT
ETag: "ec002-afa-fd67ba80"
Accept-Ranges: bytes
Content-Length: 2810
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>...</html>
```

- Linee di intestazione
 - Date
 - La data della risposta

HTTP Response

```
HTTP/1.1 200 OK
Date: Fri, 17 Sep 2009 07:59:01 GMT
Server: Apache/2.0.50 (Unix) mod_perl/1.99_10 Perl/v5.8.4
└─ mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8 mod_bigwig/2.1-3
Last-Modified: Tue, 24 Feb 2009 08:32:26 GMT
ETag: "ec002-afa-fd67ba80"
Accept-Ranges: bytes
Content-Length: 2810
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>...</html>
```

- Linee di intestazione
 - Server
 - Identificazione server/S.O./moduli e librerie installati

HTTP Response

```
HTTP/1.1 200 OK
Date: Fri, 17 Sep 2009 07:59:01 GMT
Server: Apache/2.0.50 (Unix) mod_perl/1.99_10 Perl/v5.8.4
      ↳ mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8 mod_bigwig/2.1-3
Last-Modified: Tue, 24 Feb 2009 08:32:26 GMT
ETag: "ec002-afa-fd67ba80"
Accept-Ranges: bytes
Content-Length: 2810
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>...</html>
```

- Linee di intestazione
 - Last-modified
 - La data di ultima modifica della risorsa

HTTP Response

```
HTTP/1.1 200 OK
Date: Fri, 17 Sep 2009 07:59:01 GMT
Server: Apache/2.0.50 (Unix) mod_perl/1.99_10 Perl/v5.8.4
↳ mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8 mod_bigwig/2.1-3
Last-Modified: Tue, 24 Feb 2009 08:32:26 GMT
ETag: "ec002-afa-fd67ba80"
Accept-Ranges: bytes
Content-Length: 2810
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>...</html>
```

- Linee di intestazione
 - Etag
 - Entity tag (opzionale) usato come chiave della risorsa per la gestione efficiente della cache del browser

Cache

- La cache è usata dai browser, dai server e anche sulla rete attraverso i *proxy server*
 - Cache: uno spazio di archiviazione e di memoria centrale, gestito dal software che la usa, per l'accesso efficiente alle risorse già precaricate e indicizzate
- Direttiva Cache-Control:
 - **no-store** non conserva in cache
 - **no-cache** può conservare in cache (richiede validazione)
 - **public** può conservare in cache
 - **private** per utenti singoli
 - **max-age** scadenza
 - **must-revalidate** obbligo di validazione

HTTP Response

```
HTTP/1.1 200 OK
Date: Fri, 17 Sep 2009 07:59:01 GMT
Server: Apache/2.0.50 (Unix) mod_perl/1.99_10 Perl/v5.8.4
↳ mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8 mod_bigwig/2.1-3
Last-Modified: Tue, 24 Feb 2009 08:32:26 GMT
ETag: "ec002-afa-fd67ba80"
Accept-Ranges: bytes
Content-Length: 2810
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>...</html>
```

- Linee di intestazione
 - **Accept-Ranges**
 - Il server esaudirà anche richiesta parziali di una risorsa inviando gruppi di byte

Ranges

- Il client richiede porzioni di risorsa
 - bytes: 2048-
- Il server risponde con un messaggio di stato apposito
 - **206** Partial Content

HTTP Response

```
HTTP/1.1 200 OK
Date: Fri, 17 Sep 2009 07:59:01 GMT
Server: Apache/2.0.50 (Unix) mod_perl/1.99_10 Perl/v5.8.4
↳ mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8 mod_bigwig/2.1-3
Last-Modified: Tue, 24 Feb 2009 08:32:26 GMT
ETag: "ec002-afa-fd67ba80"
Accept-Ranges: bytes
Content-Length: 2810
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>...</html>
```

- Linee di intestazione
 - Content-Length, Content-Type
 - Lunghezza in byte e tipo MIME della risposta

MIME

- Multipurpose Internet Mail Exchange
 - [RFC2045](#), [RFC2046](#), [RFC2047](#), [RFC4288](#), [RFC4289](#), [RFC2049](#)
 - Usato come standardizzazione dei messaggi di posta elettronica multi-formato, cioè con allegati
 - I messaggi mail sarebbero di puro testo – un carattere è codificato in un byte
 - Le etichette MIME battezzano gli allegati che sono codificati in testo equivalente usando la codifica *Base64*
 - Esteso alla gestione di qualunque contenuto trasferito con HTTP

MIME – Base64

- Il flusso dati da codificare viene suddiviso in sequenze da 6 bit ciascuna
- Ogni intero da 6 bit viene convertito in un carattere secondo la tabella di codifica
- 3 byte → 4 caratteri codificati
 - Es. “Man” → “TWFu”
 - In caso di mancato allineamento si usa “=” per il padding a destra

Base64 Encoding Table

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

MIME – Base64

“Bis”: | 01000010 | 01101001 | 01110011

| 01000010 | 01101001 | 01110011

“ Q m l z ”

“La”: | 01001010 | 01100001

| 01001010 | 01100001 | 00000000

“ S m E = ”

Base64 Encoding Table

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

MIME – Content Type

- Etichette *tipo/sottotipo*
 - text/plain - il formato testo (estensione .txt)
 - text/html – il testo HTML
 - application/octet-stream - il flusso dei byte di un programma eseguibile
 - image/jpeg, audio/mp3, video/mp4 – audio e video
 - application/msword - documento word
 - ...

HTTP Response

```
HTTP/1.1 200 OK
Date: Fri, 17 Sep 2009 07:59:01 GMT
Server: Apache/2.0.50 (Unix) mod_perl/1.99_10 Perl/v5.8.4
↳ mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8 mod_bigwig/2.1-3
Last-Modified: Tue, 24 Feb 2009 08:32:26 GMT
ETag: "ec002-afa-fd67ba80"
Accept-Ranges: bytes
Content-Length: 2810
Content-Type: text/html
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>...</html>
```

- **Corpo della risposta**
 - Il flusso di caratteri che compongono la risorsa se HTML
 - Il contenuto binario di un'immagine o altro dato multimediale
 - Un file Javascript o CSS esterno
 - ...

GET e POST

- I due *metodi* base di richiesta di risorse
 - GET è la richiesta standard di risorse e informazioni
 - Può essere usata anche per inviare informazioni
 - POST è di norma usata per inviare informazioni al server e innescare ulteriori elaborazioni
 - L'invio di informazioni ad un server HTTP si effettua tramite i *Form* HTML

Form HTML

- Tag HTML dedicati per l'inserimento di dati testuali, selezione da menu, radio button, checkbox ...

```
<h3>The Poll Service</h3>
<form action="http://freewig.brics.dk/users/laudrup/soccer.jsp"
      method="post">
who wins the world cup 2006?
<select name="bet">
<option value="br">Brazil!</option>
<option selected value="dk">Denmark!</option>
<option value="other country">someone else?</option>
</select><br>
Please enter your email address:
<input type="text" name="email"><br>
<input type="submit" name="send" value="Go!">
</form>
```

The Poll Service

Who wins the World Cup 2006?

Please enter your email address:

I due metodi per inviare dati al server

- Andare su:
https://tryphp.w3schools.com/showphp.php?filename=demo_form_post
 - Attivare gli strumenti per sviluppatori da Firefox e osservare il traffico di Rete
- Andare su:
https://tryphp.w3schools.com/showphp.php?filename=demo_form_get
 - Attivare gli strumenti per sviluppatori da Firefox e osservare il traffico di Rete

GET o POST?

- GET

- Va bene per **richiedere** dati
- Non sicuro: l'informazione è nella URL
- Limiti lunghezza della URL

- POST

- Richiede che i dati nel corpo messaggio siano in qualche modo “subordinati” alla URI richiesta, quindi serve per **inviare** dati
- Non sicuro: l'informazione è nel corpo della HTTP Request – poco più sicuro di GET
- Consente diversi tipi di encoding
- Consente di usare una cache delle richieste

URL encoding

- Nella URL di richiesta possono esserci spazi sia nei nomi sia nei valori dei dati e può essere utilizzata anche una tabella dei caratteri non standardizzata
- La stringa viene “ricodificata” usando il set di caratteri Unicode (UTF8)
 - Gli spazi vengono sostituiti da “+” ovvero dalla sequenza “%20”

Altri metodi HTTP

- HEAD
 - Identico a GET, ma solo per la richiesta dell'intestazione della risposta; utile per analizzare i metadati
- PUT
 - Crea o modifica l'URI richiesta con il contenuto del messaggio
- DELETE
 - Cancella la risorsa specificata

Altri metodi HTTP

- TRACE
 - Rimanda indietro la risorsa per consentire al client di verificare le modifiche inserite da server intermedi lungo la connessione (come i proxy)
- OPTIONS
 - Restituisce i metodi supportati dal server relativi alla URL specificata nella richiesta
 - OPTIONS * effettua il controllo di funzionalità di un web server
- CONNECT
 - Richiede a un proxy di stabilire un *HTTP tunneling* con l'host di destinazione
- PATCH
 - Applica modifiche parziali alla risorsa richiesta al contrario di PUT che la sostituisce del tutto

Metodi safe, idempotenti e cacheable

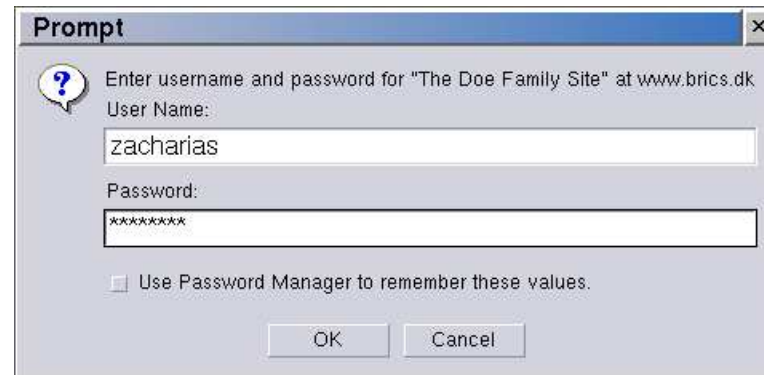
- Metodi safe
 - Sono quelli la cui esecuzione non crea effetti collaterali sul server
 - HEAD, GET, OPTIONS e TRACE
 - Richiedono informazioni
 - POST, PUT, DELETE e PATCH invece modificano anche parzialmente una risorsa
- Metodi idempotenti
 - Multiple richieste identiche lasciano lo stato del server inalterato
 - GET, HEAD, PUT, DELETE, OPTIONS e TRACE
- Metodi cacheable
 - Possono essere gestiti con la web cache
 - GET, POST e HEAD

Autenticazione in HTTP

- Tramite Form che inviano coppie username/password che vengono gestite da applicazioni lato server
 - Soluzione più sicura
- HTTP Basic
- HTTP Digest
 - Sicuro: usa cifratura MD5 di username e password
- [RFC2617](#)

Autenticazione HTTP Basic

- Il server invia al client che ha richiesto la risorsa un *challenge*:
 - HTTP/1.1 401 Authorization Required WWW-Authenticate: Basic realm="The Doe Family Site"
- Il client invia username e password in un'unica stringa codificata in *Base64*
 - Authorization: Basic emFjaGFyaWFzOmFwcGxlcGllCg==
 - La stringa è la concatenazione di username e password col carattere ":"
 - Autenticazione in chiaro: <http://pippo:pluto@www.somewhere.com>



Autenticazione HTTP Basic

- *realm*:
 - È una stringa che definisce il nome di un “protection space” cioè un gruppo di risorse che ricadono nello stesso ambito di autenticazione e sicurezza
 - Possono definirsi diversi realm per l’autenticazione a porzioni diverse di risorse dello stesso server web

Autenticazione HTTP Digest

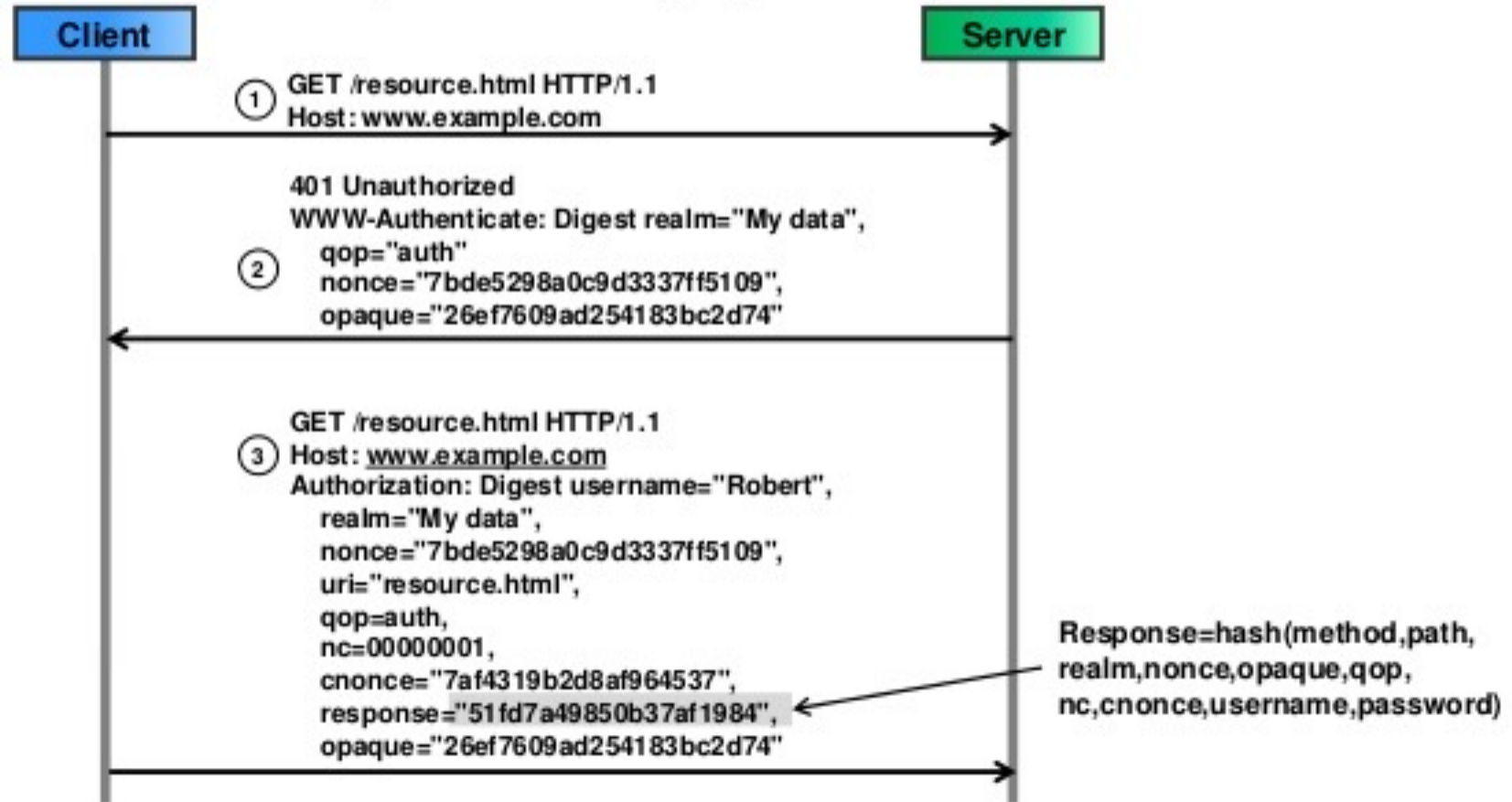
- Il client richiede la risorsa
- Il server risponde negando l'autorizzazione (401) e invia un challenge che contiene un realm e un **nonce** generato casualmente
- Il client calcola l'hash MD5 di una stringa che contiene username, realm, password, nonce del server, nonce generato dal client e altre informazioni

Numero pseudo-casuale generato per un unico utilizzo. Dall'espressione inglese «for the nonce» - per l'occasione

Autenticazione HTTP Digest

- Composizione della risposta
 - HA1 = MD5(username:realm:password)
 - HA2 = MD5(metodo:URI risposta)
 - risposta = MD5(HA1:nonce:nonceCount:clientNonce:qop:HA2)
 - **qop** (*quality of protection*) tipicamente “auth” definisce il livello di protezione e quindi di cifratura, eventualmente multipla
- La password in chiaro *non viaggia mai*

Autenticazione HTTP Digest

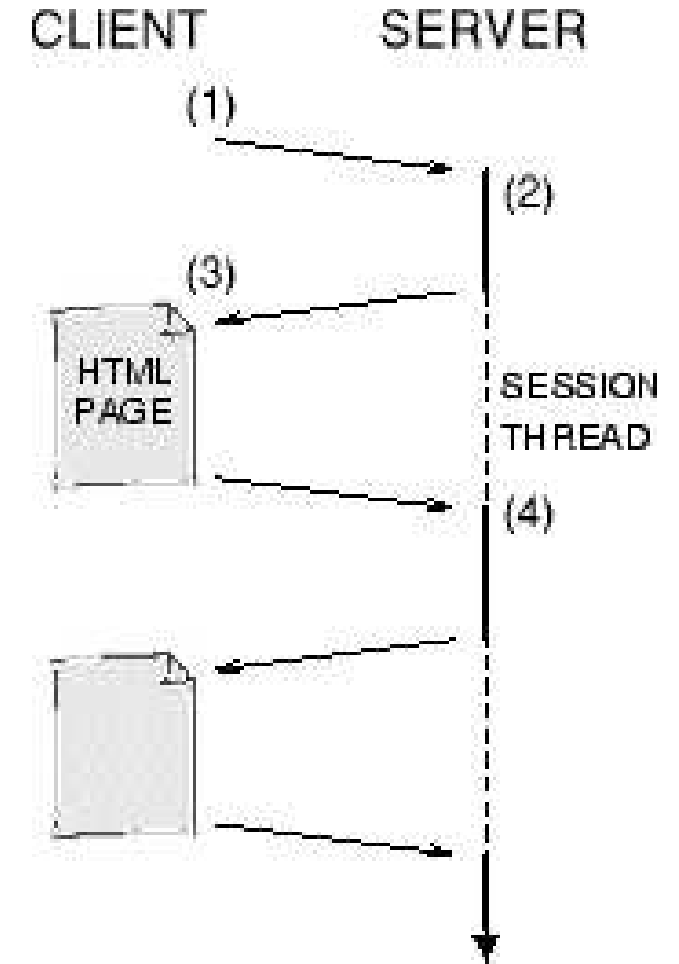


Limitazioni di HTTP

- Protocollo *stateless*
 - Non c'è controllo del tracciamento dell'attività del client, cioè la **sessione**
- Non ci sono meccanismi interni di sicurezza

Gestione della sessione

- URL rewriting
 - Inserimento dei dati utili a tracciare l'attività dell'utente già nel corpo della URL, ad esempio nomi appropriati delle cartelle nel percorso
- Esempio:
 - http://www.pets.com/show_a_product.php?product_id=7
 - <http://www.pets.com/parrots/norwegian-blue/>
- Campi nascosti nelle form
 - Contengono le informazioni di tracciamento
- Cookie
- Sessioni SSL



Cookie

- File di piccole dimensioni conservati dal server *nel client*
 - Possono essere disabilitati dal client
 - Il contenuto del cookie è reinviato al server nell'intestazione della risposta del client per avere traccia della sessione
- Server:
 - Set-Cookie: sessionid=21A9A8089C305319; path=/
- Client
 - Cookie: sessionid=21A9A8089C305319

Sicurezza con SSL/TLS

- SSL: **S**ecure **S**ocket **L**ayer [RFC6101](#)
- TLS: **T**ransport **L**ayer **S**ecurity [RFC5246](#)
- Strato *intermedio* tra TCP e HTTP
 - `https://...` - risponde alla porta 443
 - Usa **crittografia RSA a chiave pubblica**
 - I server si autenticano con i **certificati** – file contenenti la firma digitale del server, emessi da “autorità di certificazione”
 - Il client verifica la firma del server e la “catena di autenticazione” dal server all’authority