Tecnologie di Sviluppo per il Web

Il Protocollo HTTP

versione 2.3

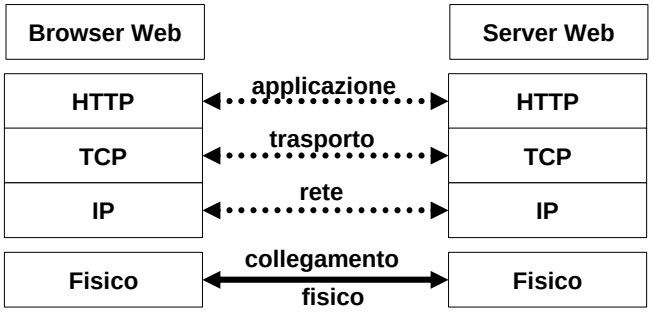
G. Mecca – mecca@unibas.it – Università della Basilicata

Sommario

- HTTP 1.0
- Transazioni
- Autenticazione e Autorizzazione
- Messaggi
- HTTP 1.1
  - ⇨Connessioni Persist.
  - ⇨Host Virtuali
  - ⇨Autenticazione e Autorizzazione
  - ⇨HTTPS, SSL – cenni
- Configurazione di un Server HTTP

HTTP 1.0

- "Standard" IETF (RFC 1945) >> HTTP 1.1
- Protocollo di applicazione



HTTP 1.0: Transazioni

- Scambio di messaggi HTTP
  - ⇨Transazione
- Transazione HTTP
  - ⇨scambio di messaggi tra server e client
  - ⇨il client apre una connessione con il server
  - ⇨il client invia la richiesta sulla connessione
  - ⇨il server invia la risposta sulla connessione
  - ⇨la connessione viene chiusa

HTTP 1.0: Transazioni

- Caratteristiche del protocollo
  - ⇨non orientato alle connessioni – una nuova connessione per ogni transazione
  - ⇨privo di stato – nella transazione successiva non resta traccia di quanto avvenuto nelle transazioni precedenti
- Attenzione
  - ⇨la mancanza di stato influenza la scrittura delle applicazioni

HTTP 1.0: Transazioni

- Come nasce normalmente la richiesta
  - ⇨l'utente seleziona un URI
    es: http://www.unibas.it/index.html
  - ⇨l'URI può essere specificato esplicitamente
    es: nella barra degli indirizzi del browser
  - ⇨oppure può provenire da un collegamento ipertestuale selezionato dall'utente
  - ⇨o da una richiesta implicita

HTTP 1.0: Transazioni in Dettaglio

- I Operazione
  - ⇨risoluzione del nome: il client utilizza il servizio DNS per risolvere il nome in num. IP
    es: www.unibas.it >> 193.204.16.105
- II Operazione
  - ⇨viene richiesta una connessione al numero IP e alla porta specificata
    es: 193.204.16.105:80

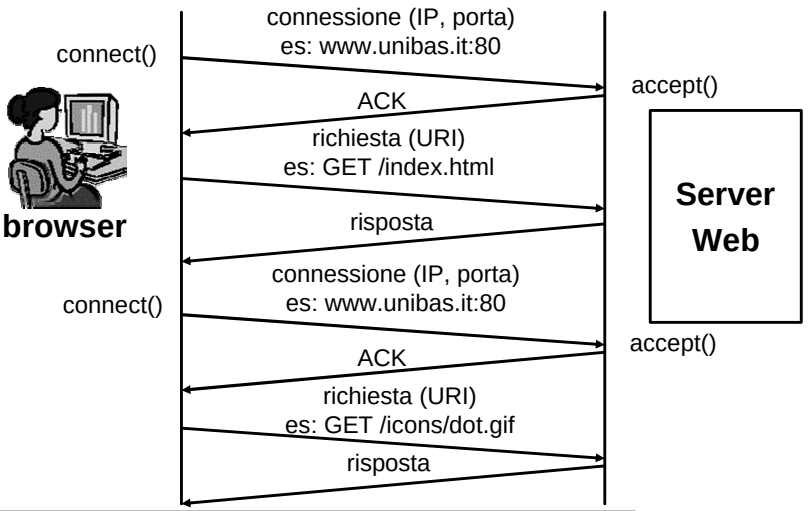
HTTP 1.0: Transazioni

- III Operazione
  - ⇨ottenuta la connessione, il browser effettua una richiesta HTTP al server specificando il percorso e il nome della risorsa
    es: GET /index.html HTTP/1.0
- IV Operazione
  - ⇨il server gestisce la richiesta e fornisce la risposta

HTTP 1.0: Transazioni

- Nota
  - ⇨le richieste HTTP sono difficilmente isolate
- Esempio
  - ⇨pagine HTML che contengono immagini
  - ⇨il codice HTML della pagina e le immagini sono risorse distinte, con URI distinti
  - ⇨viene richiesto il codice HTML
  - ⇨successivamente vengono richieste le immagini necessarie alla visualizz. completa

HTTP 1.0: Transazioni



Autenticazione e Autorizzazione

- Risorse accessibili sul server
  - ⇨tutte quelle contenute nel file system virtuale
- E' possibile limitare l'accesso
  - ⇨definire utenti e password e restringere l'accesso agli utenti autorizzati
- La tecnica standard
  - ⇨autenticazione dell'utente
  - ⇨autorizzazione da parte del server

Autenticazione e Autorizzazione

- Autenticazione
  - ⇨procedura secondo la quale un utente si identifica al server e il server lo riconosce
  - ⇨l'utente fornisce le proprie credenziali (tipicamente nome utente e password)
  - ⇨il server le confronta con un proprio archivio di utenti registrati (un file, una base di dati ecc.)
  - ⇨se non è presente, si verifica un errore

Autenticazione e Autorizzazione

- Autorizzazione
  - ciascuna risorsa protetta è accessibile ad una categoria ristretta di utenti
  - ovvero ha una serie di permessi di accesso
  - il server verifica se l'utente autenticato ha i permessi per eseguire l'operazione richiesta
  - se non ha i permessi, si verifica un errore

Autenticazione e Autorizzazione

- In HTTP
  - la procedura è esattamente quella standard
- Gli strumenti di HTTP
  - archivio degli utenti
  - reami per definire le autorizzazioni
- Reami" ("Realms")
  - risorse che condividono gli stessi permessi; es: cartella; ogni reame ha un nome

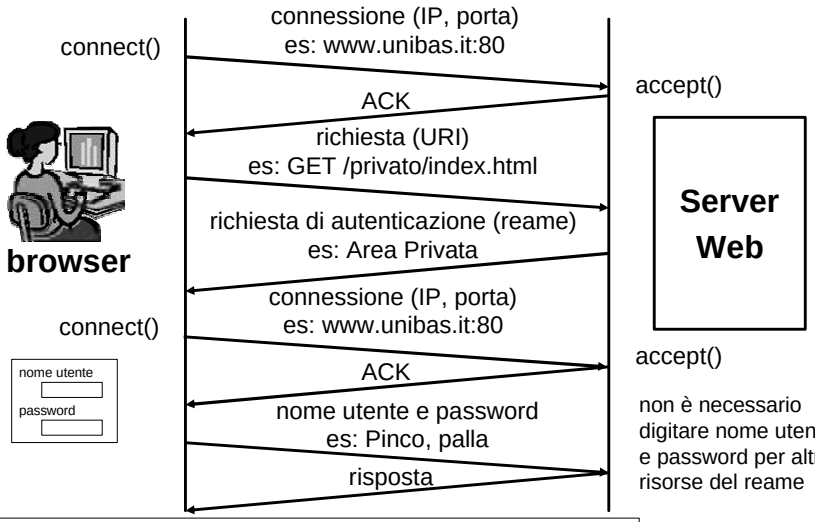
Autenticazione e Autorizzazione

- Autenticazione di tipo elementare (“Basic”)
  - il server chiede al client di autenticarsi per il reame
  - il client chiede nome utente e password all’utente
  - il client invia nome utente e password in chiaro al server (codificati come base64)
- Autorizzazione
  - il server verifica se l’utente è autorizzato ad accedere al reame
  - il browser “ricorda” le autorizzazioni ricevute per la durata della sessione

Autenticazione e Autorizzazione

- Sessione di lavoro
  - un concetto un po’ labile
  - sequenza di richieste effettuate da un utente da un client (IP) con un browser ad un server
- Tipicamente
  - la sessione di lavoro dura finchè non viene chiusa la finestra del browser
  - ma è possibile definire durate diverse (>>)

HTTP 1.0: Autenticazione



Autenticazione e Autorizzazione

- I limiti di questo approccio
  - ⇨limite di sicurezza: la password viaggia “in chiaro” e può essere intercettata
  - ⇨limite applicativo: gli utenti devono essere registrati nell’archivio del server HTTP dall’amministr. del sito
  - ⇨non è possibile “registrarsi” interattivamente

Configurazione del Server HTTP

- Fino a questo punto
  - ⇨radice del file system virtuale ed alias
  - ⇨porta
  - ⇨documento standard
  - ⇨tipi MIME
- Altri parametri da configurare
  - ⇨utenti e password
  - ⇨definizione dei reami e criteri di protezione

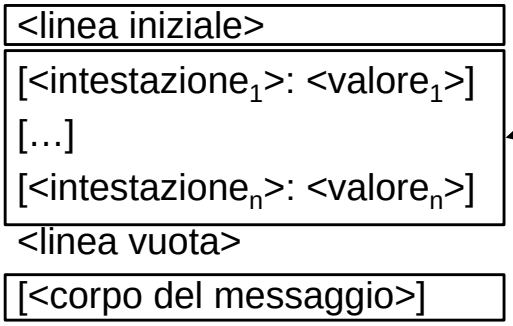
Configurazione di un Server HTTP

- Internet Information Server
  - ⇨adotta il meccanismo di protezione di Windows (utenti e password)
- Apache
  - ⇨creazione degli utenti: comando htpasswd
  - ⇨elemento <Directory> di httpd.conf per configurare reami e tipi di autenticazione per le cartelle

>> httpd.conf
>> areaPrivata

HTTP 1.0: Formato dei Messaggi

- Struttura generale dei messaggi
  - ⇨vale per richiesta e per risposta



HTTP 1.0: Formato dei Messaggi

- Linea iniziale
  - ⇨nella richiesta contiene l'URI
  - ⇨nella risposta contiene l'esito della richiesta
- Corpo
  - ⇨nella richiesta è vuota o contiene la query
  - ⇨nella risposta contiene la risorsa
- Intestazioni
  - ⇨ce ne sono numerose (vedi appendice)

HTTP 1.0: Richiesta

- Linea iniziale della richiesta
  - ⇨<metodo> <URI> HTTP/1.0
- Metodi
  - ⇨GET: metodo ordinario per effettuare richieste specificando l'URI della risorsa
  - ⇨POST: metodo per effettuare richieste specificando l'URI ed una serie di parametri nel corpo della richiesta
  - ⇨HEAD: variante di GET a scopo di controllo

HTTP 1.0: Richiesta

- Metodo GET
  - ⇨metodo standard
  - ⇨viene specificato l'URI della risorsa
  - ⇨il corpo della richiesta è vuoto
  - ⇨eventuali parametri sono nella query (e quindi sono visibili pubblicamente)
  - ⇨GET /index.html HTTP/1.0
  - ⇨GET /users/gmecca/index.html HTTP/1.0
  - ⇨GET /bollo.cgi?targa=AB123DE HTTP/1.0

HTTP 1.0: Richiesta

- Metodo POST
  - ⇨utilizzato per colloquiare con i servizi
  - ⇨viene specificato l'URI della risorsa senza parametri
  - ⇨parametri contenuti nel corpo del messaggio
  - ⇨utile per dati privati o di una certa lunghezza
  - ⇨POST /bollo.cgi HTTP/1.0
    (in questo caso i parametri sono nel corpo)

HTTP 1.0: Richiesta

- Metodo HEAD
  - ⇨variante di GET utilizzata principalmente a scopo di controllo (es: validità) e debugging
  - ⇨la richiesta è del tutto simile ad una GET
  - ⇨in risposta il server fornisce solo le intestazioni (e non il corpo)
  - ⇨HEAD /index.html HTTP/1.0
  - ⇨HEAD /bollo.cgi?targa=AB123DE HTTP/1.0

HTTP 1.0: Richiesta

- Chi decide il metodo di richiesta ?
  - ⇨non lo decide l'utente (trasparente)
  - ⇨il client ordinariamente utilizza il metodo GET
  - ⇨es: l'utente specifica un URI nella barra
  - ⇨es: l'utente seleziona un collegamento
- Metodo POST
  - ⇨quando l'utente sottomette una maschera (form) il metodo può essere POST o GET

>> /sviluppoWeb/matLS/provaMetodi.html + file di log

HTTP 1.0: Richiesta

- Intestazioni, alcuni esempi
  - ⇨User-Agent – es: User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; Q312461)
  - ⇨If-Modified-Since – es: If-Modified-Since: Thu, 01 Apr 2002 16:00:00 GMT
  - ⇨Authorization – es: Authorization: Basic ZGRpbjpvcGVuIHNl==
  - ⇨Referer – es. Referer: http://www.unibas.it/index.html

HTTP 1.0: Risposta

- Linea iniziale della risposta
  - ⇨HTTP/1.0 <codice numerico> <descrizione>
- Codice numerico
  - ⇨1xx: messaggio informativo
  - ⇨2xx: richiesta esaudita con successo
  - ⇨3xx: c'è stata una redirezione
  - ⇨4xx: errore sul lato del client
  - ⇨5xx: errore sul lato del server

HTTP 1.0: Risposta

- Esempi:
  - ⇨HTTP/1.0 200 OK
    risorsa nel corpo del messaggio
  - ⇨HTTP/1.0 301 Moved Permanently
    HTTP/1.0 302 Moved Temporarily
    nuovo URI nel corpo del messaggio
  - ⇨HTTP/1.0 404 Not Found
    HTTP/1.0 401 Unauthorized
  - ⇨HTTP/1.0 500 Server Error

HTTP 1.0: Risposta

- Intestazioni, alcuni esempi
  - Content-Type – es: Content-Type: text/html
  - Content-Length – es: Content-Length: 650
  - Last-Modified –
    es: Last-Modified Thu, 01 Apr 2002 16:00:00 GMT
  - Pragma – es: Pragma: no-cache
  - Server – es: Server: Apache 1.3.20
  - Location –
    es: Location: http://www.unibas.it/newindex.html
  - WWW-Authenticate –
    es: WWW-Authenticate: Basic realm="Area Privata"

HTTP 1.0: Un Esempio di GET

- Richiesta

GET /news/index.html HTTP/1.0
User-Agent: Mozilla/4.0
(compatible; MSIE 5.0;
Windows XP) Opera 6.0 [en]
Referer:
http://www.unibas.it/index.html
<linea vuota>

- Risposta

HTTP/1.0 200 OK
Date: Thu, 01 Apr 2002 16:00:00
GMT
Content-Type: text/html
Content-Length: 1534

```
<html>
  <head>
  …
  ...
  </body>
</html>
```

corpo della risposta: contenuto del file index.html

HTTP 1.0: Un Esempio di POST

- Richiesta

```
POST /bollo.asp HTTP/1.0
User-Agent: Mozilla/4.0
    (compatible; MSIE 5.0;
    Windows XP) Opera 6.0 [en]

targa=AB123DE&utente=Mario%
    20Rossi
```

si suppone che l'utente abbia riempito e sottomesso una maschera basata sul metodo POST

- Risposta

```
HTTP/1.0 200 OK
Date: Thu, 01 Apr 2002 16:00:00
    GMT
Content-Type: text/html
Content-Length: 2384
Pragma: no-cache
```

```
<html>
 …
 targa: AB123DE
 ...
 </html>
```

corpo della risposta: codice HTML generato dinamicam.

HTTP 1.0: Un Esempio di POST

- Attenzione alle differenze
  - ⇨nel primo caso stiamo richiedendo il contenuto di un file (index.html)
  - ⇨nel secondo caso stiamo chiedendo l'esecuzione di un'applicazione, passando dei parametri
  - ⇨l'applicazione genera il codice HTML corrispondente al messaggio di risposta

HTTP 1.1

- Standard IETF (RFC 2616)
- Principali obiettivi
  - migliorare le prestazioni di HTTP 1.0
  - rendere il protocollo più flessibile
- Attualmente
  - è implementato dalla maggior parte dei server e dei browser
  - ma viene mantenuta compatibilità con il passato per via dei vecchi browser

HTTP 1.1

- Problemi di HTTP 1.0
  - lentezza e congestione nelle connessioni >> connessioni multiple (“hack”)
  - limitatezza nel numero di IP (un IP per ciascun server Web)
  - limiti del meccanismo di autorizzazione (password in chiaro)
  - limiti nel controllo dei meccanismi di caching

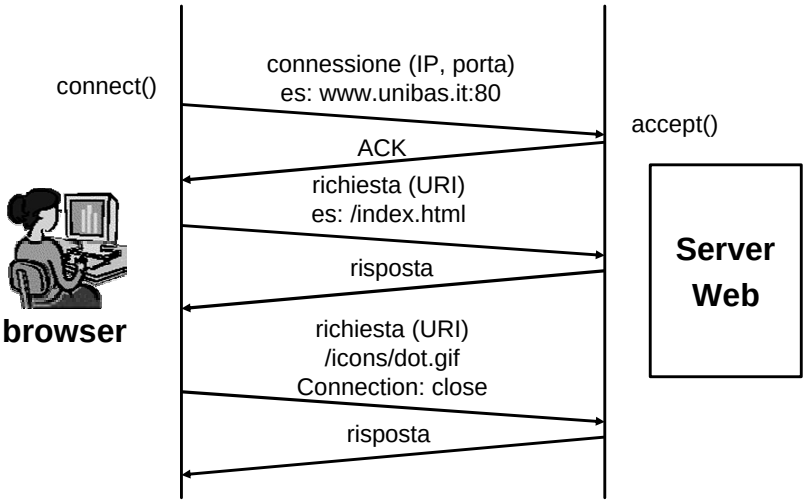
HTTP 1.1

- Novità principali
  - ⇨connessioni persistenti
  - ⇨host virtuali
  - ⇨autenticazione crittografata (“digest”)
- Altre novità
  - ⇨nuovi metodi di accesso, miglioramento dei meccanismi di caching, “chunked encoding”)

HTTP 1.1: Connessioni Persistenti

- Modalità standard di HTTP/1.1
  - ⇨più di una transazione si può svolgere lungo la stessa connessione TCP
  - ⇨nuova intestazione del client Connection: close
  - ⇨nuovo messaggio del server HTTP/1.1 100 Continue
  - ⇨il server può chiudere la connessione unilateralmente dopo un certo “timeout”

HTTP 1.1: Connessioni Persistenti



HTTP 1.1: Host Virtuali

- Ad uno stesso IP
  - ⇨possono corrispondere nomi diversi sul DNS
  - ⇨che corrispondono a server HTTP diversi
  - ⇨requisito importante per i “provider”
  - ⇨IP e porta non bastano più ad identificare il server
- Nuova intestazione del client
  - ⇨Host: serve a specificare il nome del server es: Host: www.tin.it

HTTP 1.1: Hosts Virtuali

- Indirizzo IP 192.168.3.109 con due host:
  - ⇨www.tin.it, www.virgilio.it >> /news/index.html
- Richiesta al sito 1:

```
GET /news/index.html HTTP/1.1
Host: www.tin.it
```

- Richiesta al sito 2:

```
GET /news/index.html HTTP/1.1
Host: www.virgilio.it
```


HTTP 1.1: Autenticazione "Digest"

- Le password non vengono trasmesse
- Il server invia al browser una stringa
  - ⇨"nonce"
- Il browser risponde con
  - ⇨nome utente
  - ⇨un valore crittografato basato su: nome utente, password, URI e nonce (algoritmo MD5, sunto di 128 bit in formato ASCII)
  - ⇨il browser ricorda l'autorizzazione

HTTP 1.1: Autenticazione “Digest”

- Esempio: Richiesta

GET /privato/index.htm HTTP/1.1

- Risposta

HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
realm=“Area Privata",
nonce="dcd98b7102dd2f0“

- Il browser richiede nome utente e password all’utente

- Nuova Richiesta

GET /privato/index.htm HTTP/1.1
Authorization: Digest
username=“Pinco",
realm=“Area Privata",
nonce=" dcd98b7102dd2f0",
uri="/privato/index.htm",
response="6629fae49393a05
397450978507c4ef1“

- Nuova Risposta (2xx oppure 4xx)

HTTP 1.1: Autenticazione “Digest”

- Vantaggio
  - ⇨le password non vengono trasmesse direttamente sulla rete in chiaro
  - ⇨il meccanismo è decisamente più sicuro
- Ma
  - ⇨non è sicuro al 100%
  - ⇨è possibile intercettare la richiesta con URI, nonce e sunto e riprodurla per accedere alle risorse protette

HTTP 1.1: Autenticazione "Digest"

- Inoltre
  - ⇨soffre dello stesso limite "applicativo" dell'approccio basic
- Di conseguenza
  - ⇨utilizzeremo un approccio diverso
  - ⇨in cui autenticazione e autorizzazione sono gestite a livello di applicazione con una base di dati di utenti
  - ⇨e gli utenti possono registrarsi interattivam.

HTTPS: Cenni

- La soluzione: HTTPS
- HTTPS: HTTP over SSL (RFC 2818)
  - ⇨soluzione considerata più sicura
- SSL: Secure Socket Layer
  - ⇨protocollo di trasporto
  - ⇨tutti i messaggi sono crittografati
  - ⇨crittografia a chiave pubblica (certificato)
  - ⇨trasparente per lo sviluppatore

HTTP 1.1: Altre Novità

- Nuovi Metodi di Accesso
  - Aggiornamenti delle risorse sul server
    - PUT: salvare risorse sul file system del server
    - DELETE: eliminare risorse dal file system del server
    - poco utilizzati per motivi di sicurezza
  - Diagnostica della rete
    - OPTIONS
    - TRACE
    - UPGRADE

HTTP 1.1: Altre Novità

- Miglioramento dei meccanismi di caching
  - gestione molto più sofisticata delle cache
  - più accuratezza nella specifica di validità
  - intestazione Cache-Control
- "Chunked-Encoding"
  - la risposta può essere inviata al client a pezzi, anche prima di conoscerne la lunghezza totale

Configurazione del Server HTTP

- A questo punto
  - ⇨siamo in grado di completare gli aspetti relativi alla configurazione del server HTTP
- Ricapitoliamo
- Prima operazione
  - ⇨installazione dell'applicazione
  - ⇨es: Internet Information Services
  - ⇨es: Apache

Configurazione di un Server HTTP

- Configurazione
  - ⇨radice del file system virtuale ed alias
  - ⇨eventuale porta e documento standard
  - ⇨tipi MIME
  - ⇨utenti e password
  - ⇨reami e criteri di protezione (basic o digest)
- Aspetti avanzati
  - ⇨logging
  - ⇨caching
  - ⇨host virtuali

Riassumendo

- HTTP 1.0
- Transazioni
- Autenticazione
- Messaggi
- HTTP 1.1
  - ⇨Connessioni Persistenti
  - ⇨Host Virtuali
  - ⇨Autenticazione
  - ⇨HTTPS, SSL – cenni
- Configurazione di un Server HTTP

HTTP 1.0: Intestazioni

- Sono classificate in varie categorie
  - ⇨intestazioni generali
  - ⇨intestazioni dei messaggi (valide sia per richiesta che per risposta)
  - ⇨intestazioni specifiche della richiesta
  - ⇨intestazioni specifiche della risposta
- Nel seguito
  - ⇨un elenco per HTTP 1.0 e HTTP 1.1

HTTP 1.0: Intestazioni

- Intestazioni dei messaggi (“Entity header”)
  - Content-Type – es: Content-Type: text/html
  - Content-Encoding – es: Content-Encoding: x-zip
  - Content-Length – es: Content-Length: 650
  - Last-Modified –
    es: Last-Modified Thu, 01 Apr 2002 16:00:00 GMT
  - Expires –
    es: Expires: Thu, 01 Apr 2002 16:00:00 GMT
  - Allow – es: Allow: GET, HEAD

HTTP 1.0: Intestazioni

- Intestazioni generali (“General header”)
  - Date – es: Date: Thu, 01 Apr 2002 16:00:00 GMT
  - Pragma – es: Pragma: no-cache
- Intestazioni della risposta (“Response h.”)
  - Server – es: Server: Apache 1.3.20
  - Location –
    es: Location: http://www.unibas.it/newindex.html
  - WWW-Authenticate –
    es: WWW-Authenticate: Basic realm=“Area Privata”

HTTP 1.0: Intestazioni

- Intestazioni della richiesta (“Request h.”)
  - User-Agent – es. User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; Q312461)
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:0.9.4) Gecko/20011019 Netscape6/6.2
  - From – es. From: mecca@unibas.it
  - If-Modified-Since –
    es. If-Modified-Since: Thu, 01 Apr 2002 16:00:00 GMT
  - Authorization –
    es. Authorization: Basic QWxhZGRpbjpvcGVuIHNl==
  - Referer – es. Referer: http://www.unibas.it/index.html

HTTP 1.1: Nuove Intestazioni

- Intestazioni Generali
  - Date
  - Pragma
  - Cache-Control
  - Connection
  - Trailer
  - Transfer-Encoding
  - Upgrade
  - Via
  - Warning
- Intestazioni di Entità
  - Allow
  - Content-Encoding
  - Content-Length
  - Content-Type
  - Expires
  - Last-Modified
  - Content-Language
  - Content-Location
  - Content-MD5
  - Content-Range

HTTP 1.1: Nuove Intestazioni

- Int. di Richiesta
  - Authorization
  - From
  - If-Modified-Since
  - Referer
  - User-Agent
  - Accept
  - Accept-Charset
  - Accept-Encoding
  - Accept-Language
  - Expect
  - Host
  - If-Match
  - If-None-Match
  - If-Range
  - If-Unmodified-Since
  - Max-Forwards
  - Proxy-Authorization
  - Range
  - TE

HTTP 1.1: Nuove Intestazioni

- Intestazioni di Risposta
  - Location
  - Server
  - WWW-Authenticate
  - Age
  - ETag
  - Retry-After
  - Vary

Termini della Licenza