

EL PROTOCOL HTTP

ÍNDEX

- 1. INTRODUCCIÓ**
- 2. MISSATGES DE PETICIÓ**
- 3. MISSATGES DE RESPOSTA**
- 4. CAPÇALERES HTTP**
- 5. MÈTODES HTTP**

Introducció

- El protocol HTTP segueix el model general de peticions i respostes entre un client i un servidor.
- En la primera versió (**HTTP 1.0**) s'ha d'establir una connexió per a cada petició. En el **HTTP 1.1** el sistema millora amb les connexions persistents que permeten intercanviar diferents peticions i respostes en una sola connexió.
- Versió HTTP 1.1 (RFC 2616)

<http://www.w3.org/Protocols/HTTP/1.1/rfc2616.pdf>

Introducció

- Des del mes de Juny de l'any 2014 el conjunt de documents RFC 7230 a 7235 han fet obsolet el document RFC 2616 sobre la versió 1.1 del protocol HTTP.

- Podeu trobar-los a:

<http://www.w3.org/Protocols/>

<http://www.rfc-editor.org/rfc-index.html>

Uniform Resource Identifier

L'especificació de la definició d'URI es troba a la RFC 3986. La forma general d'un URI es pot expressar de la forma següent:

esquema: identificador

- En HTTP les URI bàsicament identifiquen un recurs a partir del nom del servidor, el port de comunicacions, la ruta, etc..El seu format està definit a la secció 2.7 del document RFC7230 (<http://tools.ietf.org/html/rfc7230#section-2.7>)
- En HTTP quan es parla de localització de recursos s'utilitza el tipus específic d'URI anomenat **URL** (Uniform Resource Locator).

HTTP URL

- La URL amb l'esquema HTTP és un cas específic d'URI i es representa de la següent manera:

```
http_URL = "http:" "://" host [ ":" port ] [ abs_path [ "?" query ] ]
```

- Per exemple: <http://www.w3.org/protocols>
 - El host = www.w3.org (podria ser una IP)
 - El port si no s'especifica correspon al 80
 - El camí absolut al recurs = /protocols
 - En aquest cas no hi ha query.

Tipus i format del missatges HTTP

- D'acord amb el document RFC 7230 secció 3 (<http://tools.ietf.org/html/rfc7230#section-3>), els missatges HTTP poden ser de dos tipus:
 - **Petició** (request) del client al servidor
 - **Resposta** (response) del servidor al client.
- Els missatges consisteixen en:
 - Una **línia inicial** (diferent en el cas de la petició o de la resposta)
 - Camps de **capçalera** (headers fields)
 - El **cos del missatge** (especificat al RFC 822)
- Entre el camps de capçalera i el cos del missatge hi ha una línia en blanc (CRLF)

Tipus de missatges HTTP

- El cos del missatge, juntament amb els camps de la capçalera que proporcionen informació sobre el seu contingut, formen el que en l'HTTP es denomina una **entitat**. Cada entitat correspon a un recurs.
- Depenent de si el missatge HTTP és una petició o una resposta, la primera línia rep el nom de **línia de petició** o **línia d'estatus**, respectivament.

Tipus de missatges HTTP

■ Format missatge de petició:

línia de petició	→	GET / HTTP/1.0	MÈTODE espai URL espai VERSIÓ CR LF
capçaleres	→	Host: www.upc.es	CAPÇALERA: espai VALOR CR LF
		...	
		User-agent: telnet	CAPÇALERA: espai VALOR CR LF
línia en blanc	→	Cr lf	CR+LF
entity body	→		En aquest cas està buit. S'utilitza amb el mètode POST (mètode usat sovint per a omplir formularis)

■ Format missatge de resposta:

línia d'estat	→	HTTP/1.1 200 OK	VERSIÓ espai CODI_ESTAT espai FRASE cr lf
capçaleres	→	Date:	CAPÇALERA: espai VALOR cr lf
		...	
		Content-Type:	CAPÇALERA: espai VALOR cr lf
línia en blanc	→	cr lf	CR+LF
entity body	→		Conté l'objecte sol·licitat

Missatge petició (request)

- La **línia inicial** dels missatges de petició, anomenada **línia de petició**, té el següent format:

```
mètode <SP> URI <SP> versió <CRLF>
```

- El **mètode** especifica el tipus d'operació que sollicita el client a fet una petició. El document Document [RFC 7231 secció 4](#) defineix els diferents mètodes. Els més important són **GET**, **PUT** i **POST**.
- L'URI** especifica el recurs al qual s'aplica el mètode seleccionat i la **versió** correspon a la versió del protocol.

Missatge petició - Exemple

Per exemple per obtenir el recurs *principal.html* del servidor *www.acme.com* al port *8000*:

<http://www.acme.com:8000/doc/principal.html>

- Primer ens hem de connectar a aquest servidor en aquest port i enviar una petició amb un missatge com aquest:

```
GET /doc/principal.html HTTP/1.0
```

- En aquest cas s'utilitza el mètode GET que vol dir obtenir el recurs especificat.

Missatge resposta (response)

- La **linia inicial** del missatge de resposta rep el nom "***status line***".

versió <SP> codi <SP> frase <CRLF>

- La **versió** especifica el mateix que en el missatge de petició. El **codi** és un nombre de tres dígitos que indica el tipus de resposta rebuda d'acord amb el document **RFC 7231 secció 6** i la **frase** correspon a un text explicatiu del codi.

Codis resposta

- Al missatge de petició anterior (veure exemple) es podria rebre qualsevol dels codis de resposta que especifica el protocol depenent de la versió utilitzada, per exemple:

CODI	SIGNIFICAT
200	Operació efectuada (OK)
301	El recurs sol·licitat ha canviat d'adreça URI (Moved Permanently)
400	Petició incorrecte (Bad Request)
403	Accés prohibit al recurs (Forbidden)
404	No s'ha trobat el recurs (Not Found)
500	Error intern del servidor

HTTP Headers: Camps generals

- **Date:** indica l'hora i la data en què es va originar el missatge.
- **Pragma:** proporciona una llista de directrius als sistemes que intervenen en la transferència (clients o servidors).
 - **no-cache:** al afegir aquesta directiva a una petició significa que la petició al servidor sempre serà realitzada encara que es trobi una còpia d'aquesta mateixa petició durant el camí entre el client i el servidor.

HTTP Headers: Camps generals

- **Cache-Control:** s'utilitza per especificar les directives corresponents als mecanismes de memòries cau al realitzar peticions HTTP.
- **Connection:** permet especificar les opcions de la connexió d'una petició HTTP.
 - **close:** aquesta opció indica que la connexió es tancarà després de rebre una resposta a la petició. Els sistemes que no implementen connexions persistents utilitzaran sempre aquesta opció a totes les peticions.

HTTP Headers: Camps generals

- **Transfer-Encoding:** codificació aplicada al cos del missatge.
- **Upgrade:** el client informa de les versions del protocol suportades, aleshores el servidor escull la millor.
- **Via:** s'utilitza per part dels servidors intermediaris per afegir informació a la petició com el nom del servidor i el protocol que utilitzen.

HTTP Headers: Camps entitat

- **Content-Length:** indica la longitud en bytes del cos de l'entitat.
- **Content-Type:** indica el tipus de contingut del cos de l'entitat (per exemple text/html, si és un document HTML).
- **Content-Encoding:** indica si s'ha aplicat alguna codificació al cos de l'entitat (per exemple una compressió de dades).
- **Last-Modified:** hora i data en què el recurs contingut en el cos de l'entitat ha sigut modificat per última vegada.

HTTP Headers: Camps entitat

- **Expires:** Indica la data i l'hora a partir de la qual el contingut del cos es pot considerar obsolet o caducat per al seu emmagatzematge en la memòria cau.
- **Allow:** Indica els mètodes HTTP que es poden aplicar al recurs sol·licitat.
- **Content-Language:** llenguatge utilitzat.
- **Content-Location:** l'URI de l'entitat, en cas que el recurs corresponent en disposi de més d'una. Per exemple un recurs disponible en diferents idiomes.

HTTP Headers: Camps entitat

- **Content-MD5:** seqüència de bits per a comprovar la integritat del contingut.
- **Content-Range:** per si una entitat s'envia en diferents fragments. Indica el fragment enviat.
- **Etag:** etiqueta associada a l'entitat, per si el recurs en disposa de més d'una. Normalment s'utilitza per identificar els recursos desats a les memòries cau.

HTTP Headers: Camps petició

- **From:** a aquest camp conté l'adreça electrònica de l'usuari que sol·licita el recurs.
- **User-Agent:** Aquest camp permet identificar la implementació del client.
- **Referer:** indica en la petició actual l'adreça de la pàgina web (URL) que ha referenciat o enllaçat al recurs demandat. És a dir d'on s'ha obtingut el recurs. El servidor pot utilitzar aquesta informació per generar estadístiques, detectar des de quin recurs es referencia una adreça incorrecta o obsoleta, "hotlinking", etc.

HTTP Headers: Camps petició

- **If-Modified-Since:** Quan el client ja disposa d'una còpia del recurs sol·licitat en la memòria cau, pot utilitzar aquest camp per a dur a terme una operació GET condicional: si el recurs s'ha modificat posteriorment a la data indicada, s'efectuarà l'operació de manera normal i, si no, el servidor enviarà una resposta sense cos i amb el codi 304
- **Authorization:** amb aquest camp, un usuari pot presentar les seves credencials a un servidor perquè li permeti accedir a recursos d'accés restringit.

HTTP Headers: Camps petició

- **Accept:** tipus de contingut que accepta el client.
- **Accept-Charset**
- **Accept-Encoding**
- **Accept-Language**
- **Host:** nom del servidor a qui va dirigida la petició, per si en té més d'un. Permet que un servidor (una sola IP) amb diferents noms actuï com si fossin diferents servidors alhora.

HTTP Headers: Camps petició

- **If-Match:** permet fer una petició condicional d'una entitat partir de proporcionar el llistat de versions que té el client (identificadors d'entitat) de peticions anteriors. S'utilitza per minimitzar el tràfic de xarxa i actualitzar les memòries cau.
- **If-None-Match**
- **If-Unmodified-Since**
- **If-Range**
- **Max-Forwards**
- **Range:** sol·licitar un fragment d'una entitat.

HTTP Headers: Camps petició

- **Max-Forwards:** indica el número de salts que un missatge pot ser redireccionat. En cada salt (proxy o gateway) aquest número disminueix en una unitat, al arribar a 0 el missatge no podrà ser redirigit i no arribarà a destí. S'utilitza amb el mètode TRACE.
- **Proxy-Authorization:** amb aquest camp, un usuari pot presentar les seves credencials a un proxy que requereix autenticació.

HTTP Headers: Camps resposta

- **Server:** Aquest camp és similar a User-Agent, però referit al servidor.
- **Location:** Indica que el recurs sol·licitat s'ha d'obtenir en una altra adreça. Aquest serà el cas quan el codi de resposta sigui 301 o 302. Per norma general, quan el client rebi una resposta que inclogui aquest camp, generarà de manera automàtica una nova petició amb l'adreça indicada.

HTTP Headers: Camps resposta

- **WWW-Authenticate:** Aquest camp és obligatori quan el codi de les respostes és 401. Indica que és necessari presentar una credencial per a accedir al recurs sol·licitat, o que la que s'ha presentat no és vàlida.

Un servidor pot agrupar els seus recursos d'accés restringit en regnes (**realms**), cada un amb el seu esquema d'autenticació i conjunt d'usuaris autoritzats (amb una mateixa credencial s'hauria de poder accedir a tots els recursos d'un regne).

HTTP Headers: Camps resposta

- **Age:** indica el temps que ha passat des de què s'ha generat el recurs servit.
- **Proxy-Authenticate:** el mateix que WWW-Authenticate però per servidors proxy.
- **Public:** llista de mètodes suportats pel servidor.
- **Retry-After**
- **Vary:** llista de camps de la petició que s'han utilitzat per a seleccionar una entitat, quan el recurs en disposa de més d'una.
- **Warning:** informació addicional sobre l'estatus de la resposta.

Mètodes principals HTTP

- El **mètode GET**: serveix per a obtenir l'entitat corresponent a l'URI especificat en la línia de petició. Normalment el servidor traduirà el camí de l'URI a un nom de fitxer o programa:
 - En el primer cas, **el cos de l'entitat serà el contingut del fitxer.**
 - En el segon cas, el servidor executarà el programa i **l'entitat serà el resultat que generi.**

Mètodes principals HTTP

- **El mètode HEAD:** és igual que el GET, excepte que en la resposta el cos serà buit i, per tant, només tindrà capçalera (que haurà de ser idèntica a la que s'hauria enviat si el mètode fos el GET).
 - Per norma general, s'utilitza el mètode HEAD, per exemple, per a comprovar si un URL és vàlid o per a obtenir informació sobre un recurs sense necessitat de transferir-ne el contingut.

Mètodes principals HTTP

- El **mètode POST**: serveix per a enviar una entitat que el servidor ha d'incorporar en el recurs identificat per l'URI de la línia de petició. La semàntica d'aquest mètode depèn del tipus de recurs. Per exemple, es pot utilitzar per:
 - Afegir contingut a un recurs existent,
 - Enviar un missatge a un grup de notícies
 - Crear un registre nou en una base de dades
 - Passar dades a un programa que s'ha d'executar al servidor. **Un cas típic d'aquest últim exemple són les dades d'un formulari HTML.**

Altres Mètodes HTTP

- El mètode **PUT**: per a crear un recurs amb l'URI especificat en la petició.
- El mètode **DELETE**: per a esborrar un recurs.
- El mètode **OPTIONS**: per a obtenir informació sobre les opcions de transferència.
- El mètode **TRACE**: per a obtenir una còpia del missatge com ha arribat a la seva destinació final.

Propietats dels mètodes

- **Mètodes segurs:** Mètodes de tipus només lectura i que no fan canvis en els recursos que es troben en el servidor. Són segurs els mètodes GET, HEAD, OPTIONS i TRACE.
- **Mètodes Idempotents:** Mètodes que l'efecte de realitzar la mateixa petició sobre el servidor una o múltiples vegades causa el mateix efecte. No genera un recurs diferent per cada petició, si la petició és la mateixa. Són d'aquest tipus: PUT, DELETE i els mètodes segurs.

Propietats dels mètodes

- **Mètodes "Cacheables"**: Les respostes del servidor es poden emmagatzemar en el client per ser utilitzades en el futur. Són d'aquest tipus els mètodes GET, HEAD i POST però a la pràctica una grandíssima majoria d'implementacions en servidors i clients del protocols HTTP només permeten emmagatzemar en la memòria cau les respostes a mètodes GET i HEAD, de manera que la resposta a un POST generalment no es pot afegir a la memòria cau.

Autors:

- Pau López
- Ricard Torrell
- Daniel Collados