

SMTP

Introduzione

Scambio di messaggi asincrono

Basato su tecniche di store and forward

Utilizza il protocollo SMTP (Simple Mail Transfer Protocol) e l'estensione MIME (Multipurpose Internet Mail Extension) per

- definizione del formato degli indirizzi
- definizione del formato del messaggio
- definizione dei comandi

Protocollo: SMTP

Il protocollo SMTP (Simple Mail Transfer Protocol) regola il trasferimento della posta elettronica su Internet. E' regolato dall RFC 821.

Esso non è limitato alla posta su TCP-IP, ma si qualifica anzi come meccanismo di consegna anche tra canali diversi ed incompatibili (es. TCP-IP e X-25). L'unico requisito è che esista un host noto e connesso ad entrambi i canali.

Gli indirizzi

Sono composti da un identificativo utente, il carattere '@', ed un identificativo del mail host di base dell'utente. Le maiuscole non sono rilevanti.

`vitali@cs.unibo.it`

Il messaggio

È composto di un blocco di intestazione (header) ed un corpo (body), separati da una riga vuota (sequenza CRLF).

From: cianca@CS.UniBO.IT (Paolo Ciancarini)

Subject: stampante hp su mac

To: vitali@CS.UniBO.IT

Date: Wed, 17 Dec 97 18:42:14 MET

Fabio,

sul Mac manca il drive della stampante HP

Paolo

I comandi

Sono definiti un certo numero di comandi per la spedizione di messaggi ed informazioni tra applicazioni SMTP. L'host in possesso di un messaggio contatta l'host di destinazione o un host intermedio, si identifica, specifica il destinatario ed il mittente e manda il messaggio.

SMTP non viene utilizzato per accedere alla propria casella postale, quindi non ha meccanismi di identificazione via password.

```
HELO cs.unibo.it
FROM: giorgio@mail.alinet.it
RCPT-TO: cianca.cs.unibo.it
DATA
    .... messaggio
QUIT
```

Limiti di SMTP

- Solo ASCII 7-bit
- Lunghezza massima 1 Mb
- I messaggi possono essere forzati a capo dopo 80 caratteri

MIME

Per ovviare ai limiti di SMTP è stato elaborato MIME (Multipurpose Internet Mail Extensions): oggetti aventi un formato precisato nell RFC 1521-1522, come estensione del protocollo SMTP.

I messaggi MIME utilizzano la divisione in header e body di SMTP, e permettono di trasmettere messaggi multimediali o complessi tramite canali

SMTP, superando le note limitazioni per quel che riguarda formato dei dati, lunghezza dei messaggi, security.

Il protocollo MIME è stato realizzato per:

- Includere oggetti multipli all'interno di messaggi SMTP,
- Utilizzare codici di carattere diversi dallo standard US-ASCII
- Rappresentare documenti testuali formattati e multiformat
- Rappresentare materiale non-testuale come immagini, video, etc.
- Codificare secondo meccanismi adatti per la trasmissione su canali limitati in possibilità o imperfetti per sicurezza e privacy.
- Facilitare ulteriori estensioni come la definizione di nuovi tipi di dati o di codifica.

Un messaggio MIME rispetta il formato header-body di SMTP e codifica il body in maniera opportuna.

Definisce un criterio per categorizzare il formato di dati utilizzato (famiglia/formato)

`text/plain - text/html`

`image/gif - image/jpeg`

`application/x-msword, application/x-gzip`

Definisce alcuni header per specificare il formato dei dati (Content-Type) ed il formato della codifica (Content-Encoding), e l'eventuale utilizzo di messaggi multipli (multipart/*)

Alla fine il messaggio è codificato in modo da essere ASCII 7-bit, 75 caratteri per riga, spezzato o composto in modo da essere minore di 1 Mb.

HTTP

Introduzione

HTTP (HyperText Transfer Protocol):

Un protocollo stateless per la ricerca, il recupero e la manipolazione di risorse di rete (documenti)

HTTP è un protocollo con la leggerezza e la velocità necessari per un sistema informativo ipertestuale distribuito e collaborativo. È un protocollo generico, stateless, object-oriented, che può essere usato anche per scopi diversi dallo scambio di documenti ipertestuali, come name server, sistemi object-oriented distribuiti, etc.

Caratteristica importante in HTTP è la negoziazione del formato di dati utilizzato, per garantire l'indipendenza del sistema dal formato di rappresentazione dei dati.

Le applicazioni HTTP

- **Client:** una applicazione che stabilisce una connessione HTTP
- **Server:** una applicazione che accetta connessioni HTTP
- **User agent:** il client che inizia una richiesta HTTP
- **Origin server:** il server che possiede la risorsa richiesta
- **Proxy:** Un'applicazione intermediaria che agisce sia da client che da server. Se richieste sono soddisfatte autonomamente, o passandole ad altri server, con possibile trasformazione, controllo, verifica delle richieste
- **Gateway:** un'applicazione che agisce da intermediario per qualche altro server. A differenza del proxy, il gateway riceve le richieste come fosse l'origin server: il client può non essere al corrente che si tratta del gateway.
- **Tunnel:** un programma intermediario che agisce da trasmettitore passivo di una richiesta HTTP. Il tunnel non fa parte della comunicazione HTTP, anche se può essere stato attivato da una connessione HTTP.

- **Cache:** memoria locale di un'applicazione e il sistema che controlla i meccanismi della sua gestione ed aggiornamento. Qualunque client o server può utilizzare una cache, ma non un tunnel.

La connessione HTTP

La comunicazione avviene su connessioni TCP-IP secondo un protocollo stateless, direzionale ed atomico basato su quattro stadi della transazione:

- La connessione
- La richiesta
- La risposta
- La chiusura

La richiesta

La richiesta è un messaggio MIME formato da una riga di richiesta e da dati ulteriori facoltativi

La richiesta semplice è:

```
GET URI CrLf
```

dove

URI è un identificatore univoco di risorse specificato dal protocollo URI

La richiesta completa è la seguente:

```
Method URI ProtocolVersion CrLf
[MIME Header]*
CrLf
Body
```

dove:

- Method è uno dei possibili metodi di comunicazione tra client e server.
- URI è un identificativo univoco di risorse
- ProtocolVersion è “HTTP/V1.0”
- MIME Header sono linee così formate:
- Fieldname: Value CrLf
- Body è un messaggio MIME

I metodi

I metodi indicano l'azione da eseguire sull'oggetto identificato dall'URI. Di seguito una lista dei metodi più importanti, nel bene o nel male:

GET: recupera la risorsa associata all'URI specificato e lo trasmette come risposta. Se all'URI non corrisponde un documento, ma un processo di produzione di dati o uno script, il protocollo restituisce l'output dell'esecuzione del processo o dello script, non il codice corrispondente alla risorsa stessa.

HEAD: come GET, ma restituisce i soli header MIME della risposta

POST: si richiede che la destinazione accetti il contenuto del messaggio come nuovo subordinato della risorsa specificata dall'URI. Questo vuol dire che la risorsa può diventare una annotazione all'URI specificato, un messaggio per una bulletin board o simili, dati in input per un'applicazione specificata, un'estensione ad un database.

La risposta

La risposta è un messaggio MIME formato da un codice di stato e da un messaggio MIME facoltativo, formato da header e body.

Ha cioè il seguente formato:

```
HTTPVersion StatusCode ReasonLine CrLf
[MIME Header]*
CrLf
Body
```

dove:

- HTTPVersion è “HTTP/V1.0”
- StatusCode è un codice numerico di 3 cifre che rappresenta il risultato (di successo o di errore) della richiesta effettuata.

- ReasonLine è una stringa leggibile da un essere umano che contiene una spiegazione verbale del codice restituito

I codici di risposta

Informativa: 1xx

Solo sperimentale

Successo: 2xx

200 Ok: la richiesta è stata eseguita

201 Created: l'oggetto trasmesso dalla richiesta POST è stato creato e memorizzato nell'URI specificato

202 Accepted: la richiesta è stata accettata, ma la processazione è incompleta

204 No content: la richiesta è stata accettata, ma non c'è risposta.

Redirezione: 3xx

Indica azioni che dovrebbe compiere il client (automaticamente) per la corretta esecuzione della richiesta

300 Multiple choices: la risorsa richiesta è disponibile in più locazioni. Da usare insieme ad una lista di locazioni possibili.

301 Moved Permanently: la risorsa è stata spostata ad un URI differente. Alla risposta vengono aggiunte una o più righe di header contenenti gli URI correnti per la risorsa richiesta. Poiché lo spostamento è definitivo, il richiedente dovrebbe aggiornare l'indirizzo nelle proprie risorse.

302 Moved Temporarily: la risorsa è stata sposta ad un URI diverso. Poiché lo spostamento è temporaneo, il richiedente dovrebbe mantenere il vecchio URI nelle proprie risorse.

304 Not modified: la risorsa non è stata modificata dalla data ed ora specificata nel If-Modified-Since.

Errori del client: 4xx

400 Bad request: Errore di sintassi nella richiesta

401 Unauthorized: mancano i criteri di autorizzazione alla richiesta

403 Forbidden: Il server ha compreso la richiesta ma non la eseguirà.
Fornire credenziali di autorizzazione non serve e non è opportuno ripetere la richiesta.

404 Not Found: la richiesta non esiste all'URI specificato.

Errori sul server: 5xx

500 Internal Error: un errore inaspettato nel server ha impedito il completamento della richiesta.

501 Not Implemented: il server non implementa il servizio richiesto.

502: Bad gateway: il server, agendo da gateway o proxy, ha ricevuto una risposta non valida dall'applicazione seguente nella catena di richieste.

503: Service unavailable: il server non è in grado di fornire il servizio a causa di malfunzionamento o sovraccarico dell'applicazione.

Gli header

Authorization: Uno user agent che voglia autenticarsi presso un server (di solito dopo un codice 401) può farlo includendo uno header "Authorization:" con delle credenziali per farsi identificare e per fornire i criteri necessari per avere accettata la richiesta.

Content-Type: Il tipo di dato (MIME) utilizzato per il documento. Esso rappresenta l'UNICO modo per un client di determinare il tipo di dato della risorsa richiesta: né l'estensione, né particolarità del tipo di dati lo autorizzano altrimenti.

Es.: `Content-Type: text/html`

Content-Encoding: A volte il dato può essere codificato in qualche maniera per superare limiti di comunicazione o diminuire i tempi di trasmissione. Con content-encoding viene specificato il metodo di codifica utilizzato (MIME).

Es.: `Content-Encoding: application/x-gzip`

Content-Length: indica la lunghezza del body (in byte espressi in formato decimale). E' obbligatorio per le risposte e rappresenta l'unico mezzo per determinare la dimensione del documento ricevuto.

Es. : Content-Length: 3514

Date: la data e l'ora in cui il messaggio è stato creato.

Expires: la data e l'ora in cui il messaggio non è più valido. Serve per determinare la validità di una copia di cache e per alcuni trucchetti (in particolare il *client-pull*).

From: fornisce l'indirizzo Internet dell'umano che controlla lo user-agent.

If-Modified-Since: rappresenta il meccanismo principale per verificare la consistenza di una cache. Usato in un GET, ritorna 200 se la risorsa è stata modificata, e 304 (not modified) altrimenti.

Last-Modified: la data di ultima modifica della risorsa secondo il server. Questa viene usata dal client per verificare se la propria copia (in cache) è valida o meno.

Location: per risposte 3xx, fornisce l'URI attuale di una risorsa spostata.

Pragma: un campo generale per aggiungere nuove intestazioni utilizzabili da applicazioni specializzate. Per il momento l'unica direttiva ufficiale è "no-cache", per indicare che non è opportuno fare cache della risorsa specificata.

Es.: Pragma: no-cache

Referer: permette al client di specificare a vantaggio del server l'URI della risorsa da cui è partita la richiesta nuova.

Server: permette al server di specificare tipo e versione dell'applicazione utilizzata.

User-Agent: permette al client di specificare tipo e versione dell'applicazione utilizzata.