Introduzione agli URI

Fabio Vitali

Introduzione

Qui esaminiamo:

- Gli Universal Resource Identifier (URI)
- Alcuni esempi di schemi
- Il problema degli URN (che verrà approfondito in una ulteriore lezione)

URI

- Gli URI (Universal Resource Identifier) sono una sintassi usata in WWW per definire i nomi e gli indirizzi di oggetti (risorse) su Internet.
- Questi oggetti sono considerati accessibili tramite l'utilizzo di protocolli esistenti, inventati appositamente, o ancora da inventare.
- Gli URI si orientano a risolvere il problema di creare un meccanismo ed una ***sintassi di accesso unificata*** alle risorse di dati disponibili via rete.
- Tutte le istruzioni d'accesso ai vari specifici oggetti disponibili secondo un dato protocollo sono codificate come una stringa di indirizzo

L’esigenza di identificatori (1)

Gli URI sono stati verosimilmente il fattore determinante per il successo del WWW.

Attraverso gli URI, il WWW è stato in grado di identificare risorse accessibili tramite il proprio protocollo, HTTP, e tramite tutti gli altri protocolli esistenti (FTP, Telnet, Gopher, WAIS, ecc.).

Il punto principale a cui gli altri sistemi non erano arrivati era una sintassi universale, indipendente dal protocollo e facilmente memorizzabile o scambiabile con cui identificare le risorse di rete.

A seguire: L’esigenza di identificatori (2)

L’esigenza di identificatori (2)

Il WWW utilizza gli identificatori in una varietà di modi:

- Immagini ed altri oggetti inclusi nel documento HTML (che è un formato solo testo); Connessioni e relazioni globali tra documenti (ad esempio, script e link possono essere messi esternamente al documento HTML e da esso riferiti globalmente.
- Link ipertestuali disponibili nel documenti HTML
- Identificatori di namespace per documenti XML; Identificatori di risorsa su cui esprimere meta-informazioni in RDF; identificatori di risorse di cui fornire firme crittografiche o valori hash.

Nel primo caso, è importante che l'applicazione sia in grado di accedere ad una risorsa che è tipicamente nello stesso spazio di nomi della risorsa base; nel secondo caso, può essere in uno spazio di nomi diverso; nel terzo, interessa solo identificarla, non accedervi; tipicamente è in uno spazio di nomi diverso dalla risorsa base.

A seguire: L’esigenza di identificatori (3)

L’esigenza di identificatori (3)

Quindi due sono i fattori chiave nella valutazione della architettura degli URI:

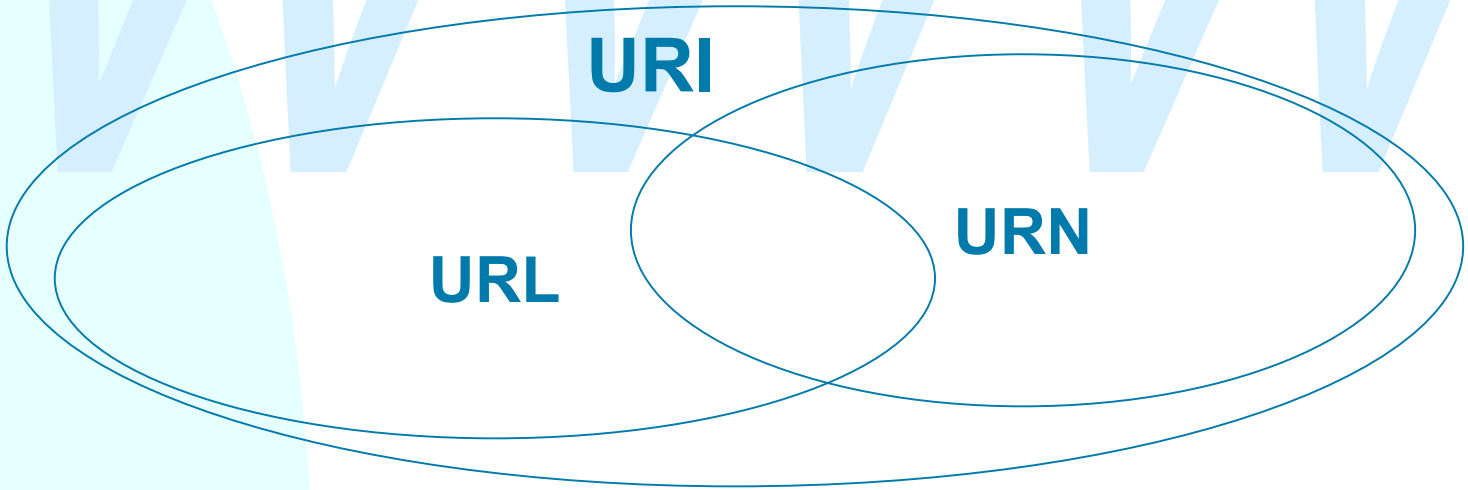
- Accessibilità: l'URI contiene informazioni su come fisicamente accedere alla risorsa indirizzata, che possa essere usata direttamente da un'applicazione (N.B.: NON sto dicendo che *accessibile* è meglio di *non accessibile*)
- Controllo: l'estensore dell'URI ha il controllo sulla mappatura dall'URI alla risorsa (ad esempio, sulla posizione di memorizzazione della risorsa) e gli interessa che le cose vengano mantenute semplici e corrette

Questi due fattori a loro volta comportano differenze in aspetti come permanenza e robustezza dell'URI, meccanismi di accesso alla risorsa, ecc.

Gli URI

Gli *Universal Resource Identifier* (URI) sono, per definizione:

- *Universal Resource Locator* (URL): una sintassi che contiene informazioni immediatamente utilizzabili per accedee alla risorsai (ad esempio, il suo indirizzo di rete)
- *Universal Resource Names* (URN: una sintassi che permetta una etichettatura permanente e non ripudiabile della risorsa, indipendentemente dal riportare informazioni sull'accesso.



Gli URI (2)

Nella visione classica, i due insiemi erano disgiunti: un URI era o un URL, che riportava informazioni sul protocollo, sul nome dell'host e sul nome locale della risorsa, oppure era un URN, che non riportava queste informazioni perché non persistenti.

Nella visione moderna, la distinzione tra URL e URN è secondaria rispetto al concetto di schemi: ogni URI appartiene ad uno schema (la parte della stringa che precede i due punti.

Alcuni schemi sono per natura persistenti, e possono essere usati per identificare in maniera non ripudiabile delle risorse; altri per natura contengono informazioni dirette per l'accesso, e allora possono essere considerati dei locatori.

Ma le cose non si escludono, e in ogni caso dipendono dalla natura dello schema.

A seguire: Gli URI (3)

Gli URI (3)

- Gli URL sono un indirizzo della risorsa che possa essere immediatamente utilizzato da un programma per accedere alla risorsa.
- Gli URL contengono tutte le informazioni necessarie per accedere all'informazione, ma sono fragili a modifiche non sostanziali del meccanismo di accesso (es. cambio del nome di una directory).
- Gli URN sono un nome stabile e definitivo di una risorsa, che possa fornire un informazione certa ed affidabile sulla sua esistenza ed accessibilità.
- Gli URN debbono essere trasformati da un apposito servizio, negli URL attualmente associati alla risorsa. Inoltre la mappa deve essere aggiornata ogni volta che la risorsa viene spostata.

Il concetto di risorsa

Gli URI sono pensati per essere indipendenti dal meccanismo di memorizzazione effettiva sottostante.

Anche se molti URI fanno riferimento a file memorizzati in un file system gerarchico, questo non è né necessario:

- Potrebbe essere in un file system relazionale (VM di IBM)
- Potrebbe essere in un database, e l'URI essere la chiave di ricerca
- Potrebbe essere il risultato dell'elaborazione di un'applicazione, e l'URI essere i parametri di elaborazione.
- Potrebbe essere una risorsa non elettronica (un libro, una persona, un pezzo di produzione industriale), e l'URI essere il suo nome universale
- Potrebbe essere un concetto astratto (la grammatica di un linguaggio)

Per questo si usa il termine Risorsa, invece che File, e si fornisce una sintassi indipendente dal sistema effettivo di memorizzazione. Mai assumere che si stia lavorando con un file system!

Criteri di design degli URI (1)

La sintassi degli URI é progettata per essere

- **Estensibile**: si possono aggiungere nuovi schemi, al fine di mantenere l'accessibilità delle risorse anche se nuovi protocolli vengono inventati
- **Completa**: tutti i nomi esistenti sono codificabili e nuovi protocolli sono comunque esprimibili tramite URI
- **Stampabile**: é possibile esprimere URI con caratteri ASCII a 7-bit, così da permettere scambi lungo qualunque canale, per quanto limitato o inefficiente, inclusi carta e penna.

Lo standard URI definisce alcune regole per la generazione di schemi di naming (insiemi di nomi caratterizzati dalla dipendenza da un protocollo di accesso comune), per la definizione dei caratteri accettabili e del carattere di escape.

La sintassi degli URI

Un URI è diviso in due parti:

- `uri = schema ":" parte-specifica`

Lo schema di naming (in pratica, il protocollo) é identificato da una stringa arbitraria (ma registrata) usata come prefisso. Il carattere di due punti separa il prefisso dal resto. La decodifica del resto dell'URI é funzione del prefisso.

Ogni schema ha una sua sintassi, ma esistono delle regole che tutti gli schemi debbono rispettare.

Caratteri ammessi negli URI (1)

I caratteri degli URI sono

curi : unreserved | reserved | escaped

I caratteri non riservati sono alfanumerici e alcuni caratteri di punteggiatura privi di ambiguità

unreserved: uppercase | lowercase | digit | punctuation

punctuation: -_.!~*'()

I caratteri riservati sono caratteri che hanno delle funzioni particolari in uno o più schemi di URI. In questo caso vanno usati direttamente quando assolvono alle loro funzioni, e escaped quando sono invece parte della stringa identificativa naturale

reserved: ;/?:@&=+$,

Caratteri ammessi negli URI (2)

I caratteri escaped fanno riferimento alle seguenti tipologie di caratteri:

- I caratteri non US-ASCII (cioè ISO Latin-1 > 127)
- I caratteri di controllo: US-ASCII < 32
- I caratteri *unwise*: { } | \ ^ []`
- I delimitatori: spazio<>#%"
- I caratteri riservati quando usati in contesto diverso dal loro uso riservato

In questo caso i caratteri vanno posti in maniera escaped, secondo la seguente sintassi:

- escaped: %XX, dove XX è il codice esadecimale del carattere

A seguire: Caratteri riservati negli URI (1)

Caratteri riservati negli URI (1)

% Il carattere “%” é il codice di escape, e serve per l’utilizzo di caratteri particolari nell’URI, precedendone il codice esadecimale. Ad esempio, per utilizzare un carattere “%” nel’URI bisogna usare la stringa “%25”

/ Il carattere “/” é utilizzato unicamente per l’identificazione di sottoparti di uno schema gerarchico, e non può essere usato per altri scopi.

. Il punto singolo “.” o il punto punto “..” hanno anch’essi un significato gerarchico riservato, per indicare ovviamente risorse allo stesso livello o al livello superiore.

A seguire: Caratteri riservati negli URI (2)

Caratteri riservati negli URI (2)

- # Il carattere di hash “#” serve per delimitare l’URI di un oggetto da un identificatore di un frammento interno alla risorsa considerata. Questo permette ad un URI di far riferimento non soltanto ad una risorsa (oggetto di interesse del server), ma anche a frammenti interni alla risorsa (che verranno identificati dal client).
- ? Il punto interrogativo “?” serve per separare l’URI di un oggetto su cui é possibile fare una query (un database, per esempio), dalla stringa usata per specificare la query.
- + All’interno della query, il segno più “+” é usato al posto dello spazio (che non é mai usato per nessuna ragione).

A seguire: Caratteri riservati negli URI (3)

Caratteri riservati negli URI (3)

* L'asterisco "*" ha un significato speciale all'interno di schemi specifici.

! Analogamente il punto esclamativo "!" ha un significato all'interno di uno schema.

%XX Caratteri speciali o riservati o in generale non sicuri (es. quelli superiori al codice ASCII 127) possono essere specificati tramite codifica esadecimale introdotta dal carattere di escape.

A seguire: Caratteri riservati negli URI (4)

Caratteri riservati negli URI (4)

Esempio: i due URI

- `http://www.alpha.edu/a/b/c/d`
- `http://www.alpha.edu/a/b/c%2Fd`

non sono uguali, perché, benché il codice esadecimale corrisponda al carattere “/”, nel primo caso esso ha significato gerarchico, e nel secondo fa parte del nome dell’ultima sottoparte della gerarchia, “c/d”.

Gli elementi di un URI

Gli URI hanno tipicamente la seguente forma:

- <schema>**:**<autorità><percorso>**?**<query>**#<**frammento>

Non sempre ci sono tutti (a parte schema): alcuni non hanno autorità (ad esempio, le news), altri non hanno la query, ecc.

Alcuni URI hanno uno spazio di nomi gerarchico. In questo caso la parte autorità inizia con un "//", e la parte path contiene dei caratteri "/" a separare le varie parti del percorso gerarchico

Se l'URI non è gerarchico, non c'è la sequenza "//" e il carattere "/" non può essere usato nella parte rimanente dell'URI.

- <schema>**://**<autorità>[**/**<el>](0-n)**?**<query>**#**<frammento>

URI assoluti e relativi

Un URI assoluto contiene tutte le parti predefinite dal suo schema, esplicitamente precisate.

Un URI gerarchico può però anche essere relativo, ed in questo caso riportare solo una parte dell'URI assoluto corrispondente. Sicuramente manca la parte schema.

Un URI relativo fa sempre riferimento ad un URI di base (ad esempio, l'URI assoluto del documento ospitante l'URI relativo) rispetto al quale fornire elementi di differenza.

Risolvere un URI relativo

Risolvere un URI relativo significa identificare l'URI assoluto corrispondente sulla base dell'URI relativo stesso e dell'URI di base. Questo avviene come segue:

- Se inizia con "#", è un frammento interno allo stesso documento di base
- Se inizia con uno schema, è un URL assoluto
- Se inizia con "/", allora è un path assoluto all'interno della stessa autorità del documento di base, e gli va applicata la stessa parte autorità.
- Altrimenti, si estrae il path assoluto dell'URI di base, meno l'ultimo elemento, e si aggiunge in fondo l'URI relativo.
- Si procede infine a semplificazioni per le sequenze:
  - "./" (stesso livello di gerarchia): viene cancellata
  - "../" (livello superiore di gerarchia): viene eliminato insieme all'elemento precedente.
- Ciò che rimane (se rimane qualcosa) è l'URL assoluto risolto. Ogni ostacolo incontrato genera un errore.

A seguire: Esempi di risoluzione

Esempi di risoluzione

Dato l'URI base: **http://a/b/c/d**, i seguenti URI relativi diventano:

g:h	**g:h**
g	**http://a/b/c/g**
./g	**http://a/b/c/g**
#s	**(current document)#s**
g#s	**http://a/b/c/g#s**
.	**http://a/b/c/**
./	**http://a/b/c/**
..	**http://a/b/**
../	**http://a/b/**
../g	**http://a/b/g**
../..	**http://a/**
../../	**http://a/**
../../g	**http://a/g**

HTTP e HTTPS

I protocolli più usati nel WWW. HTTPS prevede una crittografazione in entrambi i sensi del contenuto del messaggio. Per il resto sono identici. La sintassi di questo schema è:

```
http://host[:port]/path[?query][#fragment]
https://host[:port]/path[?query][#fragment]
```

dove:

- **host** é l'indirizzo TCP-IP o DNS, dell'host su cui si trova la risorsa
- **port** é la porta a cui il server é in ascolto per le connessioni. Per default, la porta è 80 per HTTP e 443 per HTTPS.
- **path** é un pathname gerarchico (per esempio, un filename parziale) per l'identificazione della risorsa
- **query** é una frase che costituisce l'oggetto di una ricerca sulla risorsa specificata.
- **fragment** é un identificativo di una sottoparte dell'oggetto. La definizione e il ritrovamento di queste sottoparti é a carico del client, e quindi la parte di fragment viene ignorata dal server, che restituisce l'intero oggetto.

FTP

La sintassi della parte specifica è:

```
ftp://[user[:password]@]host[:port]/path
```

dove:

- **User e password** sono utente e password per l'accesso ad un server FTP. La mancanza di user fa partire automaticamente una connessione anonima
- Si tende a scoraggiare l'uso della password nell'URL, in quanto evidente situazione di scarsa sicurezza. Tuttavia lo schema lo prevede come parte facoltativa.
- **Host, port e path** sono l'indirizzo del server, la porta di connessione ed il nome del file dell'oggetto ricercato, come per HTTP. La porta di default è 21.

SMTP e Telnet

SMTP

La sintassi della parte specifica è:

```
mailto:user@host
```

dove

- non esiste il prefisso “//” perché lo schema non è gerarchico
- User e host sono i componenti dell’indirizzo di e-mail del destinatario

Telnet

La sintassi della parte specifica è:

```
telnet:host:port
```

URN (1)

Ci sono ancora problemi per il successo degli URN. Non esiste ancora nessun meccanismo di URN sufficientemente affermato.

Gli scopi degli URN sono:

- **Ambito globale**: non viene indicata una locazione, ed ha lo stesso significato da ovunque lo si usi
- **Unicità globale**: non è possibile assegnare lo stesso URN a risorse diverse
- **Persistenza**: Non esiste ragione per la sua cessata esistenza a parte la cancellazione della risorsa a cui fa riferimento.
- **Scalabilità**: ogni risorsa sulla rete deve poter possedere per lungo tempo un URN

A seguire: URN (2)

URN (2)

- **Estensibilità**: nuove funzionalità emergeranno. E' necessario che lo schema di URN permetta estensioni per coprire le esigenze delle nuove funzionalità.
- **Supporto per i meccanismi esistenti**: esistono già dei meccanismi di naming globali: numeri ISBN per i libri, identificatori pubblici ISO per gli standard, codici UPC per i prodotti fisici. Lo schema di naming deve inglobare trasparentemente questi schemi di naming.
- **Risoluzione**: deve esistere un meccanismo semplice per la mappatura di un URN nell'URL più appropriato
- **Indipendenza**: ogni suddivisione gerarchica dell'autorità dei nomi deve essere autonoma (cioè gestisce in autonomia i nomi ad essa soggetti).

Conclusioni

Qui abbiamo parlato di

- La sintassi degli URI e degli URL
- Il probema degli URN (da approfondire)

Riferimenti

Wilde's WWW, capitolo 2

Altri testi:

- RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax. T. Berners-Lee, R. Fielding, L. Masinter. August 1998.
- RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June1999.
- RFC 2717 Registration Procedures for URL Scheme Names. R. Petke, I. King. November 1999.
- RFC 2718 Guidelines for new URL Schemes. L. Masinter, H. Alvestrand, D. Zigmond, R. Petke. November 1999
- RFC 3305 Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations. M. Mealling, Ed., R. Denenberg, Ed.. August 2002