ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Uniform Resource Identifier (URI)

Fabio Vitali

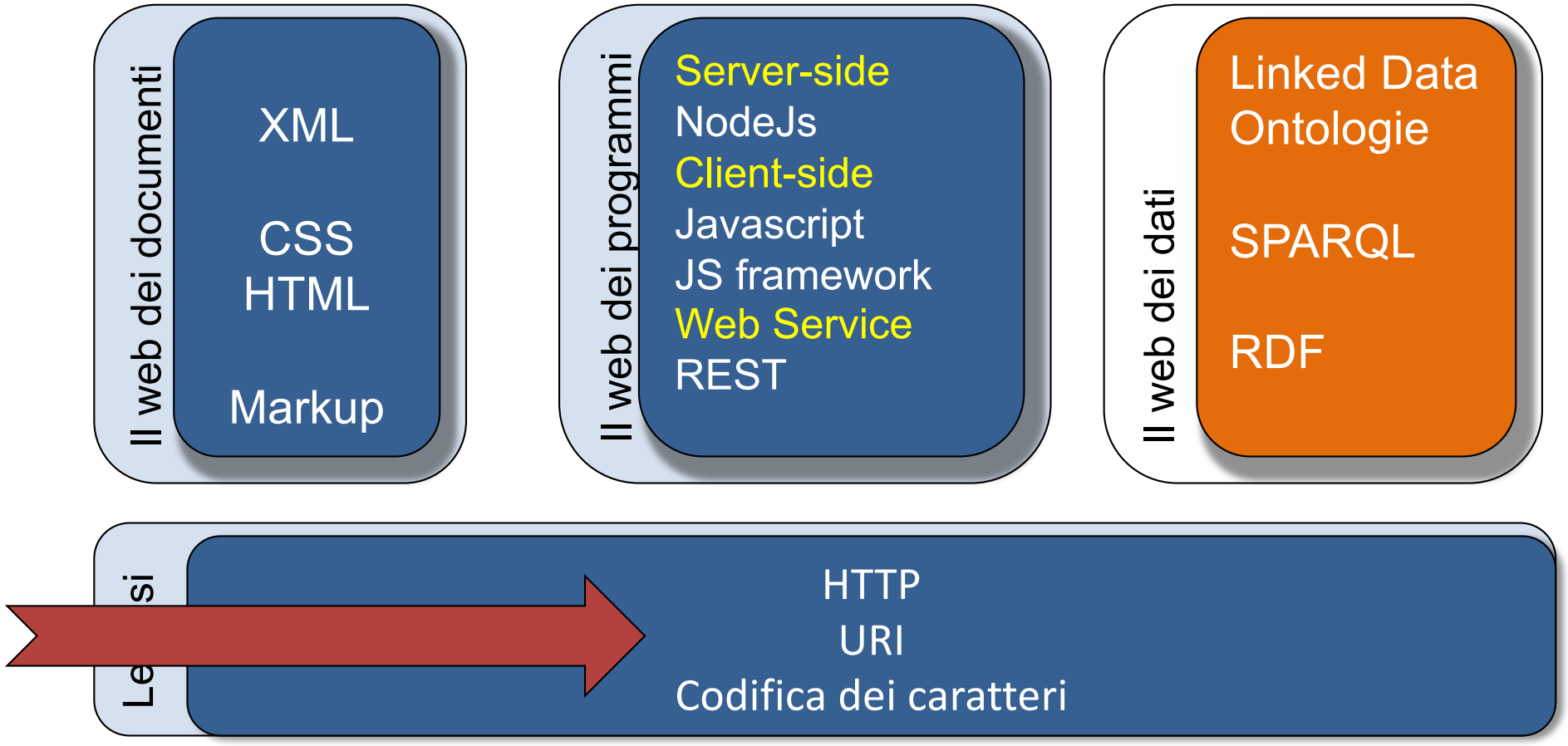
Corsi di laurea in Informatica e
Informatica per il Management
Alma Mater – Università di Bologna

Introduzione

Qui esaminiamo:

- Gli Uniform Resource Identifier (URI)
- Alcuni esempi di schemi
- Alcuni concetti derivati dagli URI
- Alcuni problemi degli URI

Argomenti delle lezioni



Principi architetturali del WWW

- Identificazione: Il WWW è uno spazio informativo (per umani e applicazioni) in cui ogni elemento di interesse è chiamato risorsa ed è identificato da un identificatore globale chiamato URI.
- Interazione: Un protocollo di comunicazione chiamato HTTP permette di scambiare messaggi su una rete informatica (nel nostro caso TCP/IP)
- Formato: la disponibilità di apposite applicazioni sul computer ricevente e la corretta identificazione del formato dati con cui la risorsa è stata comunicata permette di accedere al suo contenuto in maniera chiara e non ambigua.I formati sono molti, tra cui XHTML, PNG, RDF, ecc.)

Il concetto di risorsa

- Il concetto di risorsa è indipendente dal meccanismo di memorizzazione effettiva o dal tipo di contenuto. Una risorsa è qualunque struttura che sia oggetto di scambio tra applicazioni all'interno del World Wide Web.
- Anche se molti identificatori fanno riferimento a file memorizzati in un file system gerarchico, questo tipo di risorsa non è né necessario né frequente:
  - 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 Uniforme
  - 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.

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, 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.

L’esigenza di identificatori (2)

- Il WWW utilizza gli identificatori in una varietà di modi:
  - Immagini ed altri oggetti inclusi nei documenti HTML.
  - Link ipertestuali posti nel documenti ipertestuali
  - Identificatori di namespace per documenti XML (i.e., un vocabolario di termini a cui fare riferimento);
  - Identificatori di risorsa su cui esprimere affermazioni
  - 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.

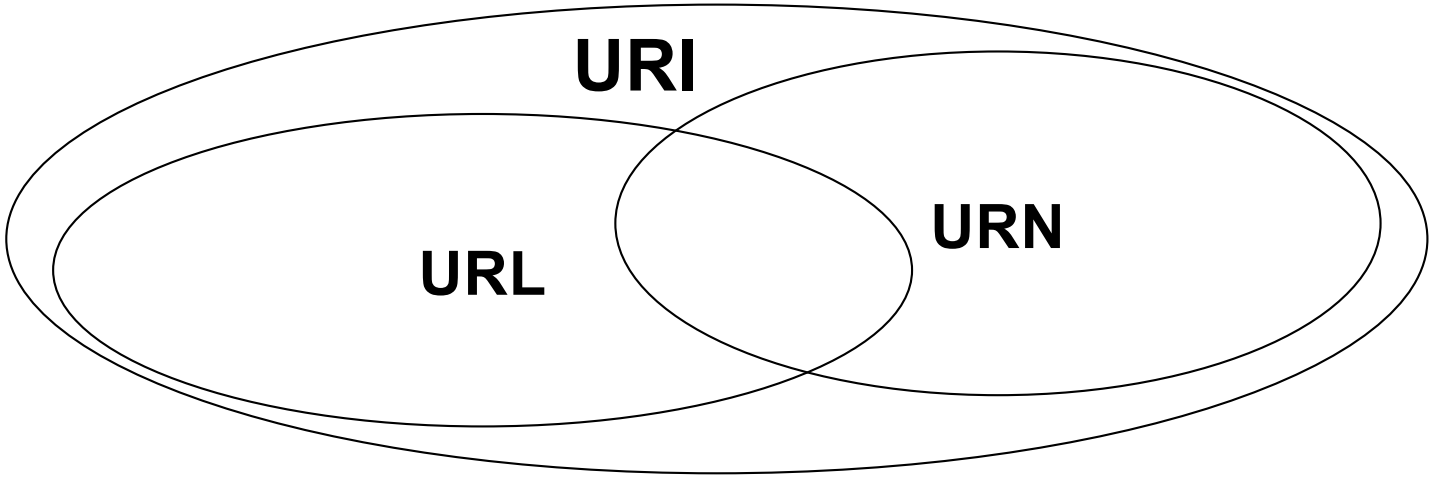
URI (1)

- Gli URI (Uniform 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

URI (2)

Gli Uniform Resource Identifier (URI) sono, per definizione:

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



URI (3)

- Gli URL sono un indirizzo della risorsa che possa essere immediatamente utilizzato da un programma per accedervi.
- Gli URL contengono tutte le informazioni necessarie per accedere all'informazione, ma sono fragili e soggetti 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.

URI (4)

- 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). Esiste uno schema urn: che è un URN, ma non è l'unico tipo di URN.
- 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.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Sintassi degli URI

Criteri di design degli URI

Gli URI sono progettati per essere:

- ***Trascrivibili***: gli URI sono sequenze di caratteri tratti da un set molto limitato (lettere, numeri, alcuni - pochi - caratteri speciali), per permettere la trascrizione degli URI su canali anche non elettronici (un pezzo di carta, un cartellone, ecc.)
- ***Fornire identificazione, non interazione***: le operazioni eseguibili sulle risorse ed i protocolli per eseguirli non sono indicati nell'URI, ma nell'interpretazione che degli URI si fa all'interno dello specifico schema.
- ***Fornire spazi di nomi organizzati gerarchicamente***: i caratteri ":", "/", "?" e "#" separano ambiti diversi all'interno degli URI. Il primo, onnipresente ed obbligatorio, è lo schema di interpretazione del resto. Schemi gerarchici usano il carattere "/" per separare livelli di specificità diversi del nome locale. "?" separa la parte di identificazione della risorsa da qualunque parametro necessario per accedervi, e "#" separa l'identificativo della risorsa da quello del frammento interno a cui si fa riferimento.

Organizzazione degli URI (1)

Un URI è diviso nelle parti seguenti:

URI = *schema* : [// *authority*] *path* [? *query*] [# *fragment*]

- Esempi:
  - http://www.ietf.org/rfc/rfc2396.txt
    ftp://ftp.is.co.za/rfc/rfc1808.txt
    https://purl.oclc.org/OCLC/PURL/FAQ
    file: ///Documenti/corsi/tw04/slides/l1.html
    telnet: //melvyl.ucop.edu/
    mailto: John.Doe@example.com

    AAUAAAAFCAYAAACNbyblAAAAHElEQVQI12P4//8/w38GI
    AXDIBKE0DHxgljNBAAO9TXL0Y4OHwAAAABJRU5ErkJggg=
    =

Componenti degli URI

schema : [// *authority*] *path* [? *query*] [# *fragment*]

- Lo ***schema*** (negli URL è il protocollo) é identificato da una stringa registrata presso IANA usata come prefisso.
- La presenza esplicita di ***un'authority*** individua un'organizzazione gerarchica dello spazio dei nomi a cui sono delegati i nomi (che possono essere ulteriormente delegati). L'autorità è a sua volta divisa in:
  - authority = [userinfo @] host [: port]
- La parte ***userinfo*** non deve essere presente se lo schema non prevede identificazione personale. La parte ***host*** è o un nome di dominio o un indirizzo IP. La ***port*** può essere omessa se ci si riferisce ad una well-known port (per http è la porta 80).

Organizzazione degli URI (2)

schema : [// authority] path [? query] [# fragment]

- La parte ***path*** è la parte identificativa della risorsa all'interno dello spazio di nomi identificato dallo schema e (se esistente) dalla authority.
- Poiché la presenza di una parte authority evidenzia uno spazio di nomi gerarchico sotto la gestione dell'authority, in questi casi la parte path è divisa in blocchi separati da slash "/", ciascuno dei quali è un componente del path organizzato in gerarchia.
- In questo caso diventano significativi gli pseudo componenti "." e "..".

Organizzazione degli URI (2)

schema : [// *authority*] *path* [? *query*] [# *fragment*]

- La parte ***query*** individua un'ulteriore specificazione della risorsa all'interno dello spazio di nomi identificato dallo schema e dall'URI precedente. Di solito questi sono parametri passati all'URI (un processo) per specificare un risultato dinamico (es. l'output di una query su un motore di ricerca). La parte query è tutto quello che sta dopo il punto interrogativo e prima del carattere hash. Tipicamente ha la forma *nome1=valore1&nome2=valore+in+molte+parole*
- La parte ***fragment*** individua una risorsa secondaria (una risorsa associata, dipendente o in molti casi un frammento) della risorsa primaria. E' tutta la parte che sta dopo al carattere di hash "#".

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

Caratteri riservati negli URI (4)

Esempio: i due URI

http://www.alpha.edu/uno/due/tre

http://www.alpha.edu/uno/due%2Ftre

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”.

Route

- Una route è un'associazione della parte *path* di un URI ad una risorsa gestita o restituita da un server web.
- ***Managed route***: il server associa ogni URI ad una risorsa o attraverso il file system locale (*risorse statiche*) oppure generate attraverso una computazione (*risorse* dinamiche).
  - Molto di moda oggi con node.js e express.js
- ***File-system route***: il server associa la radice della parte *path* ad una directory del file system locale e ogni filename valido all'interno di quella directory genera un URI corretto e funzionante.
  - Il vecchio approccio via web server come Apache e le applicazioni server-side di tipo LAMP.

Una *managed route*

```
var router = require("express").Router();

function getName(req, res) {
    res.send("<p>Fabio</p>");
}

function getEmail(req, res) {
    res.send("fabio.vitali@unibo.it");
}

router.get("/name", getName);
router.get("/email", getEmail);
app.use("/css", express.static('css'))
```

URI della richiesta	Risorsa restituita
`http://www.example.com/name`	`<p>Fabio</p>`
`http://www.example.com/email`	`fabio.vitali@unibo.it`
`http://www.example.com/css/style.css`	*contenuto del file* `css/style.css`

Una *file-system route*

Organizzazione del file system	URI disponibili
/	-
var/	-
www/	-
index.html	http://www.example.com/
alice/	-
index.html	http://www.example.com/alice/
bruce/	-
index.html	http://www.example.com/bruce/
...	...
fabio/	-
index.html	http://www.example.com/fabio/
pages/	
doc1.html	http://www.example.com/fabio/pages/doc1.html
doc2.html	http://www.example.com/fabio/pages/doc2.html
index.html	http://www.example.com/fabio/pages/
img/	
img1.gif	http://www.example.com/fabio/img/img1.gif
img2.jpg	http://www.example.com/fabio/img/img1.gif
css/	
style.css	http://www.example.com/fabio/css/style.css

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Uri references (uriref)

Uri ref (o URI reference)

- Un URI assoluto contiene tutte le parti predefinite dal suo schema, esplicitamente precisate.
- Un URI gerarchico può però anche essere relativo, (detto tecnicamente un URI reference) ed in questo caso riportare solo una parte dell'URI assoluto corrispondente, tagliando progressivamente cose da sinistra.
- Un URI reference fa sempre riferimento ad un URI di base (ad esempio, l'URI assoluto del documento ospitante l'URI reference) rispetto al quale fornisce porzioni differenti.
  - Es.: l'URL reference pippo.html posto dentro al documento di URI http://www.sito.com/uno/due/pluto.html
    fa riferimento al documento il cui URI assoluto è
    http://www.sito.com/uno/due/pippo.html

Operazioni su URI

- URI resolution
  - La generazione dell'URL assoluto corrispondente all'URI
  - Si esegue quando l'URI è un URI reference oppure un URI a cui non corrisponde una risorsa fisica (un URI che non è un URL)
  - Input: un URI – output: un URI
- URI dereferencing
  - La fornitura della risorsa identificata dall'URI (ad esempio, il documento cercato)
  - Input: un URL – output: una risorsa

Risolvere un URI reference (1)

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

	Dato il base URI http://www.site.com/dir1/doc1.html
Se inizia con "#", è un frammento interno allo stesso documento di base	#anchor1 si risolve come http://www.site.com/dir1/doc1.html #anchor1
Se inizia con uno schema, è un URI assoluto	http://www.site.com/dir2/doc2.html si risolve come http://www.site.com/dir2/doc2.html
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.	/dir3/doc3.html si risolve come http://www.site.com/dir3/doc3.html

Risolvere un URI reference (2)

Altrimenti:	Dato il base URI http://www.site.com/dir1/doc1.html
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:	doc4.html si risolve come http://www.site.com/dir1/doc4.html dir5/doc5.html si risolve come http://www.site.com/dir1/dir5/doc5.html
"./" (stesso livello di gerarchia): viene cancellata	./doc6.html si risolve come http://www.site.com/dir1/./doc6.html che è equivalente a http://www.site.com/dir1/doc6.html
"../" (livello superiore di gerarchia): viene eliminato insieme all'elemento precedente.	../doc7.html si risolve come http://www.site.com/dir1/../doc7.html che è equivalente a http://www.site.com/doc7.html

Esempi di risoluzione

Dato l'URI base: http://www.sito.com/uno/due/tre, i seguenti URI relativi diventano:

pippo:pluto	pippo:pluto
pippo	http://www.sito.com/uno/due/pippo
./pippo	http://www.sito.com/uno/due/pippo
#pluto	(documento attuale)#pluto
pippo#pluto	http://www.sito.com/uno/due/pippo#pluto
.	http://www.sito.com/uno/due/
./	http://www.sito.com/uno/due/
..	http://www.sito.com/uno/
../	http://www.sito.com/uno/
../pippo	http://www.sito.com/uno/pippo
../..	http://www.sito.com/
../../	http://www.sito.com/
../../pippo	http://www.sito.com/pippo

Gli schemi 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 l'identificazione della risorsa
  - query é una frase che è 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.

Lo schema FILE (RFC 8089)

- Dà accesso ai file di un file system locale (cioè del computer su cui gira il browser)
- E' equivalente a scegliere "Apri file" nel menu del browser.
- Non girano applicazioni server-side, nessuna connessione HTTP. La sintassi è:

  file://host/path [#fragment]

- La parte host può essere eliminata, assumendo che sia localhost, che porta alla sintassi più frequente:

  file:///path

- Ad esempio:

  file:///c:/Users/mario/Pictures/img1.jpg

- MS Windows accetta anche "\", che però non è nello standard approvato!

Lo schema DATA (RFC 2397)

- Uno schema non gerarchico, che non fa riferimento ad una risorsa, ma CONTIENE la risorsa: tutti i dati della risorsa sono inseriti nell'URI vero e proprio.
- Usato per immagini inline su cui non si vuole attivare una connessione HTTP separata. La sintassi è:

data:[<media type>][;base64],<data>

- ***media type*** è un media type MIME registrato presso IANA
- In più si può condificare il dato, di soloto usando **base64;**
- Poi ci sono i ***data*** della risorsa. Per esempio:

data:text/plain;charset=UTF-8;some%20text%20for%20you

AFCAYAAACNbyblAAAAHElEQVQI12P4//8/w38GIAXDIBKE0DHxgljN
BAAO9TXL0Y4OHwAAAABJRU 5ErkJggg==

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'URI, 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.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Alcuni approfondimenti sugli URI

URI che iniziano con //

- Un URI reference può iniziare con un nome dominio, del tipo:

```
//www.sito.com/dir1/dir2/fig1.gif
```

- Essa deve essere risolta in un URI assoluto utilizzando le parti mancanti dell'URI base, che in questo caso sarebbe la pagina HTML ospitante il tag <img>.
- Quindi questo significa: "carica l'immagine fig1.gif utilizzando lo stesso protocollo della pagina HTML: HTTP se siamo fuori dall'aria protetta e HTTPS se siamo dentro l'area protetta".
- Ecco quindi che è prassi utilizzare questo tipo di URI nelle chiamate CDN:

```
//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js
//netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.j
s
```

Content Delivery Network (CDN)

In breve, una rete fortemente distribuita di server commerciali che collaborano tra loro per distribuire in maniera omogenea contenuti di grande successo senza inutili duplicazioni di occupazione e trasmissione di file.

Un uso corretto dei CDN permette di sfruttare al meglio anche le capacità di caching delle macchine end-user o dei nodi intermedi della rete.

Ad esempio ogni utente che naviga su molti siti web, che usano le stesse librerie importanti (ad esempio, Bootstrap, JQuery etc.) ma non le prendono da un CDN, deve scaricare tutte le volte gli stessi identici file.

Se questi siti invece fanno riferimento a librerie sullo stesso CDN, esse verranno scaricate una volta sola e rimarranno in cache anche per gli altri siti.

Il pericolo delle illazioni a partire dalla forma degli URI

- Gli URI possono parlare di risorse sia in senso astratto (organizzazione semantica del sito) sia concreto (dettagli fisici di memorizzazione e architettura del sistema).
- Le tecnologie attuali permettono si svincolare la forma dell'URI da tutti i dettagli connessi con la sua memorizzazione.
- Questo è fondamentale per parlare di risorse (che sopravvivono a cambiamenti di organizzazione interna) piuttosto che di file e processi (che sono fortemente dipendenti dall'organizzazione interna del sistema)
- In quest'ottica, prestate particolare attenzione ai moduli mod_alias e mod_rewrite di Apache.

URL permanenti

- Esistono molti schemi di URL che forniscono (o quantomeno promettono) le garanzie richieste ad un URL per essere un URN, in particolare la permanenza.
- Ad esempio PURL (permanent URL), o i permalink offerti da molte piattaforme di blog ecc.
- In tutti i casi, un sistema server-side prende un URL e.g. permanente) a cui non corrisponde una risorsa fisica e identifica quale sia l'URL fisico corretto per quella risorsa.
- L'accesso avviene per
  - Dereferenziazione: il sistema converte l'URI virtuale nell'URI fisico, accede alla risorsa e la restituisce al richiedente
  - Redirezione: il sistema fornisce al richiedente una risposta speciale (302 temporary redirect) e l'URI che oggi (ma solo oggi) permette di accedere alla risorsa corretta.

URI rewriting

- Anche un routing di tipo file system può essere trasformato in un routing gestito, attraverso helper di routing.
- Ad esempio Apache, come anche molti altri server web, fornisce un modulo di riscrittura degli URL, chiamato *mod_rewrite*
- Trasforma un URI visibile in un URL fisico sulla base di regole
  - http://example.com/wiki/index.php?title=Argomento

  può essere esposto all'esterno semplicemente come
  - http://example.com/Argomento
- Vantaggi:
  - Permette di nascondere dettagli dell'implementazione
  - Permette di realizzare sistemi di nomi perduranti oltre la vita utile del software utilizzato
  - Permette di esprimere informazioni semantiche sulla struttura del sito invece che sulle tecnologie usate

URI shortener

- Con la nascita di twitter e il limite dei 140 caratteri, sorge il problema di inserire nei tweet link ad URL anche piuttosto lunghi.
- Servizi come bit.ly, tr.im, o goo.gl permettono di creare URL molto brevi corrispondenti a URL molto lunghi
- Il servizio è un semplice rewriter (redirect, per essere più precisi) che automaticamente suggerisce un nome opaco molto breve
  - goo.gl/T655I -> http://www.ted.com/talks/tim_berners_lee_the_year_open_data_went_worldwide.html

application/x-www-form-urlencoded

E' un estensione della codifica degli URI applicata anche a risorse trasmesse su un canale HTTP (ad esempio il contenuto di un POST)

- i codici non alfanumerici sono sostituiti da ‘%HH’ (HH: codice esadecimale del carattere),
- gli spazi sono sostituiti da ‘+’,
- i nomi dei controlli sono separati da ‘&’,
- il valore è separato dal nome da ‘=‘

application/x-www-form-urlencoded (2)

- Ad esempio, la parte query di un URI usa il formato urlencoded;

  http://mysite.com/serversidescript?user=Fabio+Vitali&pwd=%40%40%40

- Anche i form che trasmettono dati al server (metodo POST) possono usare questo tipo di formato dati nel body della richiesta:

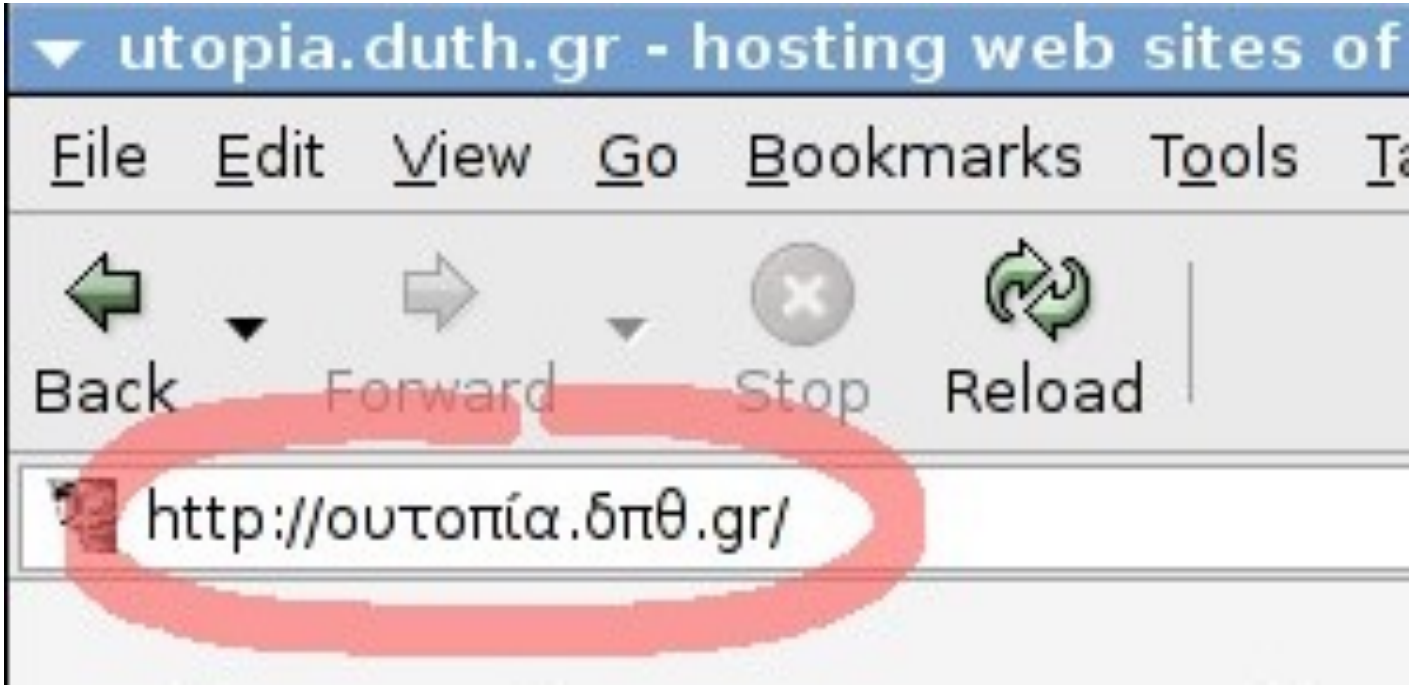
  user=Fabio+Vitali&pwd=%40%40%40

IRI

- Gli URI possono usare solo caratteri ASCII 7 bit (e neanche tutti). Ma attraverso escaping è possibile accedere a tutti i caratteri ISO Latin-1, anche se non visibili a schermo.
- Ovviamente questo è sgradevole per tutte le culture che usano caratteri che non usano alfabeti latini o per cui le estensioni di caratteri latini siano fondamentali.
- IRI (Internationalized Resource Identifier), RFC 3987 di Gennaio 2005, fornisce una sintassi per inserire caratteri di UCS-4 in un URI e per risolverli. In realtà non tutti i caratteri sono ammessi: tutti i codici di controllo e tutti i codici non corrispondenti a caratteri usati nelle lingue non sono ammessi.
- Per poter far funzionare gli IRI dobbiamo però discutere dei nomi di dominio internazionalizzati (IDN)

IDN

Internazionalized Domain Name (IDN), o anche Internazionalized Domain Name for Application (IDNA), è uno standard IETF per estendere i nomi di dominio a tutti i caratteri non ASCII di Unicode (RFC 3490). Questi nuovi nomi di dominio vengono detti IDN.



Rischi di IDNA

(attacco via omografi)

- Uno dei rischi più importanti di IDNA è il fatto che molti caratteri di vari script si assomigliano molto.
- Ad esempio, i seguenti caratteri sono in cirillico:

  а с е і ј о р ѕ х у

- Questo può permettere a utenti maliziosi di registrare nomi di dominio sospettosamente simili a nomi noti, e fare spoofing (cioè impersonificazione)
- Ad esempio il dominio “http://paypal.com” contiene 2 a in cirillico, ma è ovviamente indistinguibile da “http://paypal.com”
- Per ovviare a questi inconvenienti alcuni registri di nomi di dominio non accettano domini contenenti mix di caratteri di script diversi.

CURIE (1/2)

- In molti casi, la lunghezza degli URI può diventare considerevole, e magari URI molto simili possono essere presenti in molte parti diverse di una stessa risorsa, richiedendo uno spreco inutile di spazio e un rischio frequente di errori di compilazione.
- I CURIE (Compact URI) sono un modo per esprimere in maniera compatta famiglie di URI che condividono lo stesso prefisso.
- Hanno una sintassi del tipo

  [prefix:curie]

- (le parentesi quadre e i due punti sono obbligatori)

CURIE (2/2)

- Il prefisso è associato (in uno di molti modi, ad esempio XML namespaces che vedremo) ad un URI assoluto che deve essere sostituito nel CURIE per ottenere l'URI cercato.
- Ad esempio:

```
<html xmlns:dom="http://www.dominio.com/">
      <p>Si veda <a href="[dom:doc1]">doc 1</a></p>
</html>
```

- In questo caso, il CURIE [dom:doc1] viene risolto nell'URI http://www.dominio.com/doc1

Linked Data

- Oltre alle pagine web, oltre alle applicazioni web, esistono in rete quantità incredibili di informazioni strutturate che sarebbe utile mettere a disposizione non sotto forma di pagine HTML pensate per la lettura da parte di umani, ma come dati accedibili e manipolabili equivalentemente da umani e software.
- Il nome Linked Data è stato proposto da Tim Berners-Lee nel 2009 come modello concettuale per la caratterizzazione di tutte le risorse di dati che possono essere messi a disposizione ed incrociati da applicazioni ed umani.
- Per essere Linked Data, una collezione di dati deve:
  - Usare URI per identificare oggetti.
  - Usare HTTP URI in modo che questi oggetti possano essere referenziati e cercati da persone e user agent.
  - Fornire informazioni utili sull'oggetto quando la sua URI è dereferenziata, usando formati standard come RDF.
  - Includere link ad altre URI relative ai dati esposti per migliorare la ricerca di altre informazioni relative nel Web.

Linked Open Data

- I LOD sono Linked Data di tipo aperto (duh!).
  - "Linked Open Data (LOD) is Linked Data which is released under an open licence, which does not impede its reuse for free." (Tim Berners-Lee)

- Sono diventati un argomento politico di grande interesse da quando alcune organizzazioni politiche (di tutto lo spettro politico) ne hanno fatto il centro di campagne per favorire la trasparenza delle pubbliche amministrazioni e l'accesso della cittadinanza alle informazioni e ai processi decisionali delle pubbliche amministrazioni.

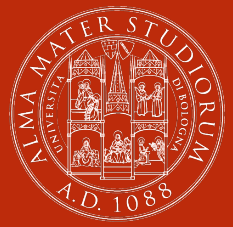
Conclusioni

Qui abbiamo parlato di

- Gli Uniform Resource Identifier (URI)
- Alcuni esempi di schemi
- Alcuni concetti derivati dagli URI
- Alcuni problemi degli URI

Riferimenti

- RFC 3986 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
- RFC 3490, IDN in Applications, Faltstrom, Hoffman, Costello, Internet Engineering Task Force (2003)



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Fabio Vitali

Corso di tecnologie web, A.A. 2017-18

Fabio.vitali@unibo.it

www.unibo.it