Tecnologie Web
URI e URL

Questioni di fondo

WWW = URL + HTTP + HTML

- Il primo termine della "formula del web" fa riferimento tre questioni principali:
  - Come identifichiamo il server in grado di fornirci un elemento dell'ipertesto (una pagina o una risorsa all'interno della pagina)?
  - Come identifichiamo la risorsa a cui vogliamo accedere?
  - Quali meccanismi possiamo utilizzare per accedere alla risorsa?
- La risposta a tutte queste domande sono gli **URI**

Uniform Resource Identifier

- Gli **URI** (**U**niform **R**esource **I**dentifier) forniscono un meccanismo semplice ed estensibile per **identificare una risorsa**
- Con il termine **risorsa** intendiamo qualunque cosa abbia una identità: un documento, un'immagine, un servizio, una collezione di altre risorse.
- Caratteristiche di un URI:
  - E' un concetto generale: non fa riferimento necessariamente ad entità disponibili in rete
  - E' un **mapping concettuale ad una entità**: non si riferisce necessariamente ad una particolare versione dell'entità esistente in un dato momento.

  ➔Il mapping può rimanere inalterato anche se cambia il contenuto della risorsa

U come Uniforme

- La sintassi degli URI rispetta una sintassi standard, semplice e regolare

  ➔ **gli identificatori sono uniformi**
- L'uniformità ha diversi vantaggi:
  - Convenzioni sintattiche comuni
  - Comune semantica per l'interpretazione
  - Possibilità di usare nello stesso contesto differenti tipologie di identificatori anche con meccanismi di accesso diversi
  - Facilità nell'introduzione di nuovi tipi di identificatori (estensibilità)

Sintassi degli URI

- Un identificatore è un frammento di informazione che fa riferimento ad una entità dotata di un'identità (risorsa).
- Nel caso degli URI gli identificatori sono stringhe con una sintassi definita, dipendente dallo schema, che può essere espressa nella forma più generale in questo modo:

  `<scheme>:<scheme-specific-part>`
- Per la componente <scheme-specific-part> non esiste una struttura o una semantica comune a tutti gli URI.
- Esiste però un sottoinsieme di URI che condivide una sintassi comune per rappresentare relazioni gerarchiche in uno spazio di nomi:

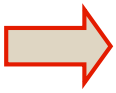
  `<scheme>://<authority><path>?<query>`
- A parte <scheme> le altre parti sono facoltative
- Alcuni schemi non prevedono la componente `<authority>` mentre altri non utilizzano `<query>`.

URN ed URL

- **Esistono due specializzazioni del concetto di URI:**
  - **Uniform Resource Name** (URN): identifica una risorsa per mezzo di un "nome" che deve essere globalmente unico e restare valido anche se la risorsa diventa non disponibile o cessa di esistere
  - **Uniform Resource Locator** (URL): identifica una risorsa per mezzo del suo meccanismo di accesso primario (es. la loro locazione nella rete) piuttosto che sulla base del suo nome o dei suoi attributi.
- **Applicando questi concetti ad una persona:**
  - L'URN è come il nome e cognome, o meglio il codice fiscale
  - L'URL è come l'indirizzo di casa o il numero di telefono

URN

- Un URN identifica una risorsa mediante un **nome** in un particolare dominio di nomi (**namespace**).
- Deve essere **unico** e **duraturo**
- Consente di "parlare" di una risorsa **prescindendo dalla sua ubicazione e dalle modalità con cui è possibile accedervi**
- Un esempio molto noto è il codice **ISBN** (International Standard Book Number) che identifica a livello internazionale in modo **univoco** e **duraturo** un libro o una edizione di un libro di un determinato editore.
- Non ci dice nulla su come procurarci il libro!



Namespace

`urn:isbn:0-9553010-9`

Nome univoco nel namespace

URL

- Un URL tiene conto anche della modalità per accedere alla risorsa
- Specifica il protocollo necessario per il trasferimento della risorsa stessa.
- Tipicamente il nome dello schema corrisponde al protocollo utilizzato
- La parte rimanente dipende dal protocollo
- Nella sua forma più comune (schema HTTP-like) la sintassi è

```
<protocol>://[<username>:<password>@]
<host>[:<port>][/<path>[?<query>][#fragment]]
```

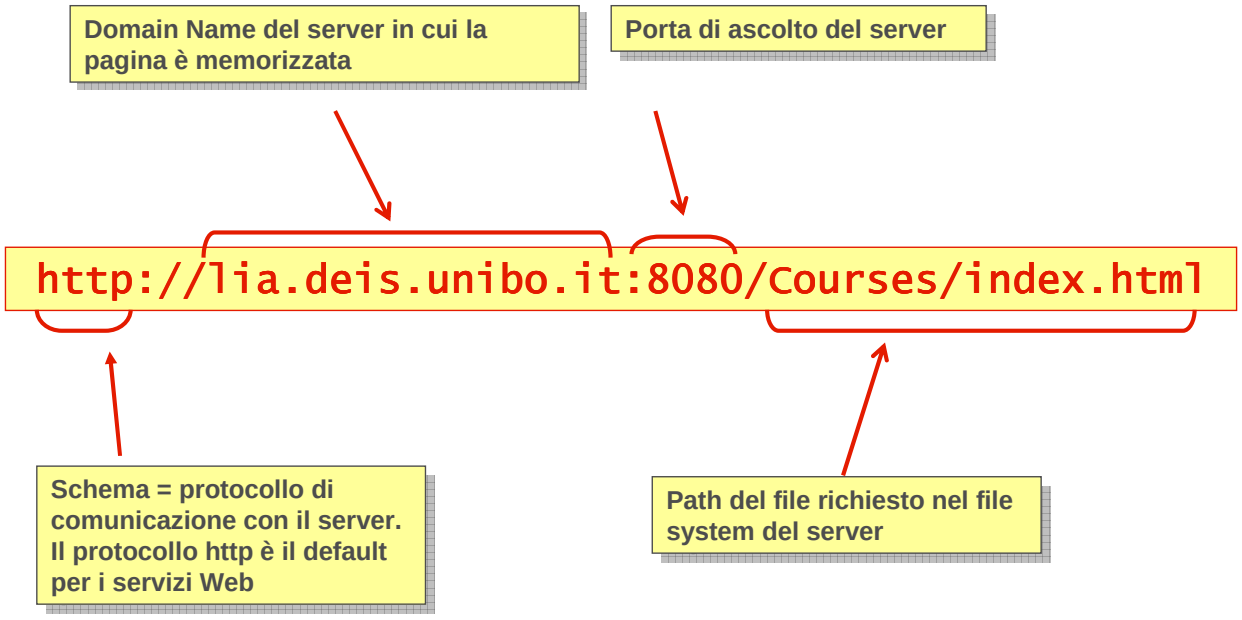
- Questa forma vale per diversi protocolli di uso comune: HTTP, HTTPS, FTP, WAP…
- Ma non, ad esempio, per la posta elettronica

Componenti di un URL con schema HTTP-like

- `<protocol>`: Descrive il protocollo da utilizzare per l'accesso al server (HTTP, HTTPS, FTP, MMS…)
- `<username>:<password>@`: credenziali per l'autenticazione
- `<host>`: indirizzo del server su cui risiede la risorsa. Può essere un nome di dominio o un indirizzo IP.
- `<port>`: definisce la porta TCP da utilizzare. Se non viene indicata si usa la porta standard per il protocollo specificato (per HTTP è la 80)
- `<path>`: percorso (pathname) nel file system del server che identifica la risorsa. Se manca tipicamente si accede alla risorsa predefinita (p.es home page).
- `<query>`: una stringa di caratteri che consente di passare al server uno o più parametri. Di solito ha questo formato:

  `parametro1=valore&parametro2=valore2…`

Esempio di URL con schema HTTP



Altri esempi di URL

- Schema per servizi **FTP**
  `ftp://ftp.FreeBSD.org/pub/FreeBSD/`
- Schema per indirizzi di **posta elettronica**:
  `mailto:enrico.lodolo@unibo.it`
- Schema per newsgroup e articoli **Usenet**
  `news:comp.infosystems.www.servers.unix`
- Schema per servizi **Telnet**
  `telnet://melvyl.ucop.edu`
- Schema per **IRC**
  `irc://irc.freenode.net/wikipedia-it`

URL opache e URL gerarchiche

- Le URI sono anche classificate come opache o gerarchiche
- **URL opaca**: non è soggetta ad ulteriori operazioni di parsing.
  - `mailto:paolo.rossi@deis.unibo.it`
- **URL gerarchica**: è soggetta a ulteriori operazioni di parsing, per esempio per separare l'indirizzo del server dal percorso all'interno file system
  - `http://deis.unibo.it/`
  - `docs/guide/collections/designfaq.html#28`
  - `../../../lab/examples/ant/build.xml`
  - `file:///~/calendar`

Parsing delle URL gerarchiche

- Una URL gerarchica è sottoposta ad un ulteriore parsing secondo la sintassi:

  `[scheme:][//authority]path[?query][#fragment]`

- La componente authority, se specificata, può essere **server-based** o **registry-based**. Una authority server-based subisce il seguente processo di parsing:

  `[user-info@]host[:port]`

- Una authority non server-based è detta registry-based.
- La componente path di una URI gerarchica è detta **assoluta** se inizia con "/", **relativa** in caso contrario.
- Il path di una URI gerarchica assoluta o che specifica una authority è sempre assoluto.

Operazioni sulle URL gerarchiche

- **Normalizzazione:** processo di rimozione dei segmenti "." e ".. " dal path di una URI gerarchica.
  - La normalizzazione su una URI opaca non ha effetto.
- **Risoluzione:** è il processo che a partire da una URI originaria porta all'ottenimento di una URI risultante.
  - La URI originaria viene risolta basandosi su una terza URI, detta base URI.
- **Relativizzazione** è il processo inverso alla risoluzione.

Esempio di risoluzione

- URL originale:
  `docs/guide/collections/designfaq.html#28`
- Base URL:
  `http://deis.unibo.it/`
- Risultato:
  `http://deis.unibo.it/docs/guide/collections/designfaq.html#28`

Riferimenti

- RFC2396, “Uniform Resource Identifiers (URI): Generic Syntax”, http://www.ietf.org/rfc/rfc2396.txt
- RFC1738, “Uniform Resource Locators (URL)”, http://www.ietf.org/rfc/rfc1738.txt
- C. D. Manning, P. Raghavan and Hinrich Schütze, “Introduction to Information Retrieval”, Cambridge University Press. 2008. (http://www-csli.stanford.edu/~hinrich/information-retrieval-book.html)