Tecnologie Web T
URI e URL

Home Page del corso: http://lia.disi.unibo.it/Courses/twt2021-info/
Versione elettronica: 1.02.URI.pdf
Versione elettronica: 1.02.URI-2p.pdf

URL: problematiche fondamentali

WWW = URL + HTTP + HTML

- Il primo termine della "formula del Web" fa riferimento a 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** (elemento dell'ipertesto) a cui vogliamo accedere?
  - Quali meccanismi (ad es. in termini di protocollo) 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 entità abbia una identità: un documento, un'immagine, un servizio, una collezione di altre risorse
- Caratteristiche di un URI:
  - È un concetto generale: non fa riferimento necessariamente a risorse accessibili tramite HTTP o a entità disponibili in rete
  - È **mapping concettuale ad una entità**: non si riferisce necessariamente ad una particolare versione dell'entità esistente in un dato momento
  - ➔ Mapping può rimanere inalterato anche se cambia il contenuto della risorsa

U come Uniforme

- Gli URI rispettano 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 (protocolli) 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** (domanda: esempi di spazi di nomi che conoscete?):

  `<scheme>://<authority><path>?<query>`
- A parte `<scheme>`, le altre parti possono talora essere omesse, come nei casi in cui non è inclusa la componente `<authority>` o non è inclusa la componente `<query>`

5

5

2 specializzazioni di URI: 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. locazione nella rete) piuttosto che sulla base del suo nome o dei suoi attributi

Applicando questi concetti ad una persona:

- URN è come identificazione basata su nome+cognome, o meglio codice fiscale
- URL è come indirizzo di casa o numero di telefono (se univoci)

6

6

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 e dove procurarci il libro

schema

`urn:isbn:0-9553010-9`

Nome univoco nel namespace

7

7

URL

- Un URL tiene conto anche della *modalità per accedere alla risorsa*
- Specifica il *protocollo necessario* per il trasferimento della risorsa stessa (non solo HTTP, quindi…)
- Tipicamente il **nome dello schema corrisponde al protocollo utilizzato**
- La parte rimanente del nome dipende dal protocollo
- Nella sua forma più comune (schema HTTP-like) 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

8

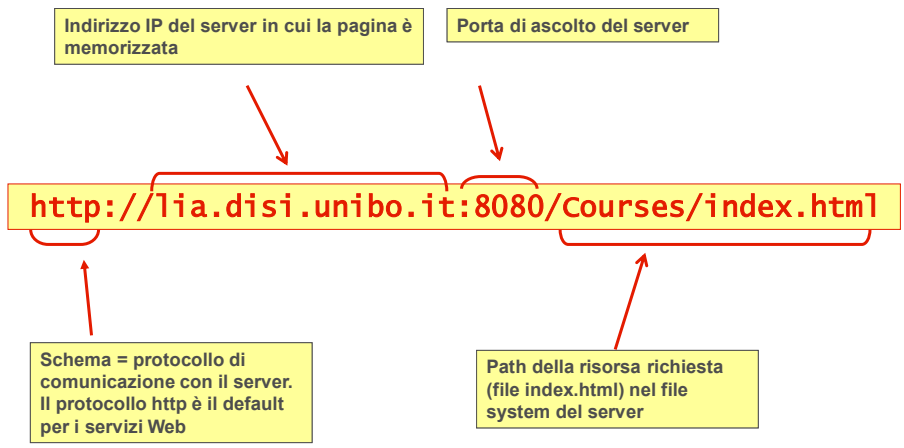
8

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 server*** su cui risiede la risorsa. Può essere un indirizzo IP logico o fisico
- `<port>`: definisce la ***porta da utilizzare*** (TCP come protocollo di trasporto per HTTP, che vedremo è a livello applicativo). Se non viene indicata, si usa porta standard per il protocollo specificato (per HTTP è 80)
- `<path>`: ***percorso (pathname) che identifica la risorsa*** nel file system del server. Se manca, tipicamente si accede alla risorsa predefinita (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 URI (alcuni di questi non sono URL, quali?)

- Schema per servizi **FTP**
  `ftp://ftp.FreeBSD.org/pub/FreeBSD/`
- 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`
- Schema per indirizzi di **posta elettronica**:
  `mailto:paolo.bellavista@unibo.it`

URI opache e URI gerarchiche

Le URI possono essere anche classificate come opache o gerarchiche

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

Operazioni sulle URI gerarchiche

- **Normalizzazione:** processo di rimozione dei segmenti "." e ".." (e altri caratteri speciali) dal path di una URI gerarchica
  - Normalizzazione si applica solo a URI gerarchiche, su URI opache 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

Semplice esempio di risoluzione (base URI)

- URI originaria:
  `docs/guide/collections/designfaq.html#28`
- Base URI:
  `http://disi.unibo.it/`
- Risultato:
  `http://disi.unibo.it/docs/guide/collections/designfaq.html#28`

Riferimenti bibliografici

- 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, H. Schütze, “Introduction to Information Retrieval”, Cambridge University Press, 2008 (http://www-csli.stanford.edu/~hinrich/information-retrieval-book.html)