



CS 4/55231 Internet Engineering	Kent State University Dept. of Computer Science LECT-10

Today's Topic

World Wide Web..




 INTERNET
 ENGINEERING


LECT-10, S-2
IN2004S_javed@kent.edu
Javed I. Khan@2004

World Wide Web Technology

WEB =

URL+ MIME + HTTP + HTML

+ INTERNET


 INTERNET
 ENGINEERING

LECT-10, S-3
IN2004S_javed@kent.edu
Javed I. Khan@2004


URL

4

General URL Syntax

<scheme>:<scheme-specific-part>

- A URL contains the name of the scheme being used (<scheme>) followed by a colon and then a string (the <scheme-specific-part>) whose interpretation depends on the scheme.
- Scheme names consist of a sequence of characters. The lower case letters "a"-"z", digits, and the characters plus ("+"), period ("."), and hyphen ("-") are allowed. For resiliency, programs interpreting URLs should treat upper case letters as equivalent.
- The characters ";", "/", "?", ":", "@", "=", and "&" are the characters which may be reserved for special meaning within a scheme. No other characters may be reserved within a scheme.



 INTERNET
 ENGINEERING

LECT-10, S-5
IN2004S_javed@kent.edu
Javed I. Khan@2004

Some Schemes

- List:


ftp	File Transfer protocol
http	Hypertext Transfer Protocol
gopher	The Gopher protocol
mailto	Electronic mail address
news	USENET news
nntp	USENET news using NNTP access
telnet	Reference to interactive sessions
wais	Wide Area Information Servers
file	Host-specific file names
prospero	Prospero Directory Service
- Other schemes may be specified by future specifications.


 INTERNET
 ENGINEERING

LECT-10, S-6
IN2004S_javed@kent.edu
Javed I. Khan@2004

Internet URL Scheme

- `scheme://<user>:<password>@<host>:<port>/<url-path>`
- user & password
 - optional; required by some protocol
- password
 - If present, it follows the user name separated from it by a colon. The user name (and password), if present, are followed by commercial-at-sign "@". Within the user and password field, any ";", "@", or "/" must be encoded.
- host
 - The fully qualified domain name of a network host, or its IP address
- port
 - Most schemes designate protocols that have a default port number. Another port number may optionally be supplied.
- url-path
 - The rest of the locator consists of data specific to the scheme, and is known as the "url-path". It supplies the details of how the specified resource can be accessed.




INTERNET
ENGINEERING

LECT-10, S-7
IN2004S_javed@kent.edu
Javed I. Khan@2004

Internet URL Scheme

- url-path of a FTP URL has the following syntax:
 - `<cwd1>/<cwd2>/.../<cwdN>/<name>;type=<typecode>`
- HTTP URL Syntax:
 - `http://<host>:<port>/<path>?<searchpart>`
- A telnet URL takes the form:
 - `telnet://<user>:<password>@<host>:<port>/`

[Click Here for Examples](#)




INTERNET
ENGINEERING

LECT-10, S-8
IN2004S_javed@kent.edu
Javed I. Khan@2004

What is Relative URL?

- Base URL + Embedded URL
- Embedded URL is specified explicitly.
- and this is how the Base URL is obtained:
 - (in order of precedence):
 - Base URL embedded in the document's content
 - Base URL of the encapsulating entity (message, document, or none)
 - URL used to retrieve the entity
 - Base URL = "" (undefined)



INTERNET
ENGINEERING

LECT-10, S-9
IN2004S_javed@kent.edu
Javed I. Khan@2004

Example: Base URL in document Content

```


<!doctype html public "-//IETF//DTD HTML//EN">
<HTML><HEAD>
<TITLE>An example HTML document</TITLE>
<BASE href="http://www.ics.uci.edu/Test/a/b/c">
</HEAD><BODY>
... <A href="..</x">a hypertext anchor</A> ...
</BODY></HTML>

```

A parser reading the example document should interpret the given relative URL "..</x" as representing the absolute URL

<URL: http://www.ics.uci.edu/Test/a/x>

regardless of the context in which the example document was obtained.



INTERNET
ENGINEERING


LECT-10, S-10
IN2004S_javed@kent.edu
Javed I. Khan@2004

Algorithm for absolute URL (1)

Step 1: The base URL is established according to the rules of Section 3. If the BASE URL is the empty string (unknown), the embedded URL is interpreted as an absolute URL and we are done.

Step 2: Both the base and embedded URLs are parsed into their component parts as described in Section 2.4.

- a) If the embedded URL is entirely empty, it inherits the entire base URL (i.e., is set equal to the base URL) and we are done.
- b) If the embedded URL starts with a scheme name, it is interpreted as an absolute URL and we are done.
- c) Otherwise, the embedded URL inherits the scheme of the base URL.




INTERNET
ENGINEERING

LECT-10, S-11
IN2004S_javed@kent.edu
Javed I. Khan@2004

Algorithm for absolute URL (2)

Step 3: If the embedded URL's <net_loc> is non-empty, we skip to Step 7. Otherwise, the embedded URL inherits the <net_loc> (if any) of the base URL.

Step 4: If the embedded URL path is preceded by a slash "/", the path is not relative and we skip to Step 7.



INTERNET
ENGINEERING

LECT-10, S-12
IN2004S_javed@kent.edu
Javed I. Khan@2004

Algorithm for absolute URL (3)

Step 5: If the embedded URL path is empty (and not preceded by a slash), then the embedded URL inherits the base URL path, and

a) if the embedded URL's <params> is non-empty, we skip to step 7; otherwise, it inherits the <params> of the base URL (if any) and

b) if the embedded URL's <query> is non-empty, we skip to step 7; otherwise, it inherits the <query> of the base URL (if any) and we skip to step 7.



INTERNET
ENGINEERING

LECT-10, S-13
IN20045_javed@kent.edu
Javed I. Khan@2004

Algorithm for absolute URL (4)

Step 6: The last segment of the base URL's path (anything following the rightmost slash "/", or the entire path if no slash is present) is removed and the embedded URL's path is appended in its place. The following operations are then applied, in order, to the new path:

a) All occurrences of ".", where "." is a complete path segment, are removed.

b) If the path ends with "." as a complete path segment, that "." is removed.

c) All occurrences of "<segment>/.", where <segment> is a complete path segment not equal to "..", are removed. Removal of these path segments is performed iteratively, removing the leftmost matching pattern on each iteration, until no matching pattern remains.

d) If the path ends with "<segment>/.", where <segment> is a complete path segment not equal to "..", that "<segment>/." is removed.



INTERNET
ENGINEERING

LECT-10, S-14
IN20045_javed@kent.edu
Javed I. Khan@2004

Algorithm for absolute URL (5)

Step 7: The resulting URL components, including any inherited from the base URL, are recombined to give the absolute form of the embedded URL.



INTERNET
ENGINEERING

LECT-10, S-15
IN20045_javed@kent.edu
Javed I. Khan@2004

Example of Relative URLs

Base: <URL:http://a/b/c/d;p?q#f>

- g = <URL:http://a/b/c/g>
- ./g = <URL:http://a/b/c/g>
- g/ = <URL:http://a/b/c/g/>
- //g = <URL:http://g>
- ?y = <URL:http://a/b/c/d;p?y>
- g?y = <URL:http://a/b/c/g?y>
- g?y/./x = <URL:http://a/b/c/g?y/./x>
- g;x = <URL:http://a/b/c/g;x>
- g;x?y#s = <URL:http://a/b/c/g;x?y#s>
- ../ = <URL:http://a/>
- .././g = <URL:http://a/g>
- g/./h = <URL:http://a/b/c/h>



INTERNET
ENGINEERING

LECT-10, S-16
IN20045_javed@kent.edu
Javed I. Khan@2004

MIME

17

Objective of MIME Types

- The purpose of the Content-Type field is to describe the data contained in the body fully enough that the receiving user agent can pick an appropriate agent or mechanism to present the data to the user, or otherwise deal with the data in an appropriate manner.

```
Content-Type: type/sub-type
Content-length:0
```

- After the type and subtype names, the remainder of the header field is simply a set of parameters
 - specified in an attribute/value notation. The set of meaningful parameters differs for the different types. In particular, there are NO globally-meaningful parameters that apply to all content-types.
- The ordering of parameters is not significant.




INTERNET
ENGINEERING

LECT-10, S-18
IN20045_javed@kent.edu
Javed I. Khan@2004

Example MIME Types

- MIME EXAMPLES
 - text/plain
 - text/html
 - application/postscript
 - image/gif
 - image/jpeg
 - audio/basic
 - x-world/x-vrml
 - audio/x-realaudio




INTERNET
ENGINEERING

LECT-10, S-19
IN2004S, javed@kent.edu
Javed I. Khan@2004

MIME Types

- In general, the top-level "Content-Type" is used to declare the general type of data while the subtype specifies a specific format for that type of data.
 - Thus, a Content-Type = "image/xyz" is enough to tell a user agent that the data is an image, even if the user agent has no knowledge of the specific image format "xyz". Such information can be used, for example, to decide whether or not to show a user the raw data from an unrecognized subtype
- However, registered subtypes of audio, image, text, and video, should not contain embedded information that is really of a different type.
- Such compound types should be represented using the "multipart" or "application" types.




INTERNET
ENGINEERING

LECT-10, S-20
IN2004S, javed@kent.edu
Javed I. Khan@2004

MIME Types

- An initial set of seven Content-Types is defined by this document.
- It is expected that additions to the larger set of supported types can generally be accomplished by the creation of new subtypes of these initial types.
- In the future, more top-level types may be defined only by an extension to this standard. If another primary type is to be used for any reason, it must be given a name starting with "X-" to indicate its non-standard status and to avoid a potential conflict with a future official name.




INTERNET
ENGINEERING

LECT-10, S-21
IN2004S, javed@kent.edu
Javed I. Khan@2004

MIME Type Format

- content := "Content-Type" ":" type "/" subtype *(";" parameter);
- type := "application" | "audio" | "image" | "message" | "multipart" | "text" | "video" | extension-token
- extension-token := x-token | iana-token
- iana-token := <a public token, must be registered with IANA>
- x-token := <The two characters "X-" or "x-" by any token>
- subtype := token ;
- The type, subtype, and parameter names are not case sensitive.




INTERNET
ENGINEERING

LECT-10, S-22
IN2004S, javed@kent.edu
Javed I. Khan@2004

Various Types

- message --
 - an encapsulated message.
 - A body of Content-Type "message" is itself all or part of a fully formatted RFC 822 conformant message which may contain its own different Content-Type header field.
 - The "primary" subtype is "rfc822".
 - The "partial" subtype is defined for partial messages, to permit the fragmented transmission of bodies that are thought to be too large to be passed through mail transport facilities.
 - Subtype, "External-body", is defined for specifying large bodies by reference to an external data source.




INTERNET
ENGINEERING

LECT-10, S-23
IN2004S, javed@kent.edu
Javed I. Khan@2004

Various Types

- multipart --
 - data consisting of multiple parts of independent data types. Four initial subtypes are defined, including the
 - primary "mixed" subtype,
 - "alternative" for representing the same data in multiple formats, "parallel" for parts intended to be viewed simultaneously, and
 - "digest" for multipart entities in which each part is of type "message".
- application --
 - some other kind of data, typically either uninterpreted binary data or information to be processed by a mail-based application.
 - primary subtype, "octet-stream", is to be used in the case of uninterpreted binary data, in which case the simplest recommended action is to offer to write the information into a file for the user.



INTERNET
ENGINEERING

LECT-10, S-24
IN2004S, javed@kent.edu
Javed I. Khan@2004

HTTP

HTTP/1.1 RFC 2068

HTTP/1.0 RFC 1945

25

HTTP

- HyperText Transfer Protocol (HTTP): The Language of communication Language between a Web-Server and a Web Client.
- The first version of HTTP/0.9 was developed in 1990 by Tim Berners-Lee, Roy Fielding, and Henrik Frystuk Nielson. It was a simple protocol for data transfer across Internet.
- Later on its modified version HTTP/1.0 became the defacto language of Web and sparked the Web Revolution.
- In november 1995, HTTP/1.1 was announced by its original authors which contained many more functionalities and stringent specifications.
- The next one on working is HTTP-ng.



INTERNET
ENGINEERING

LECT-10, S-26
IN2004S_javed@kent.edu
Javed I. Khan@2004

Some Definitions (Set-1)

- Message
 - The basic unit of HTTP communication, consisting of a structured sequence of octets matching the syntax defined in HTTP and transmitted via the connection.
- Request
 - An HTTP request message.
- Response
 - An HTTP response message.
- Resource
 - A network data object or service that can be identified by a URI. Resources may be available in multiple representations (e.g. multiple languages, data formats, size,resolutions) or vary in other ways.



INTERNET
ENGINEERING

LECT-10, S-27
IN2004S_javed@kent.edu
Javed I. Khan@2004

More Definitions (Set-1)

- Client
 - A program that establishes connections for the purpose of sending requests.
- Server
 - An application program that accepts connections in order to service requests by sending back responses. Any given program may be capable of being both a client and a server



INTERNET
ENGINEERING

LECT-10, S-28
IN2004S_javed@kent.edu
Javed I. Khan@2004

What is HTTP?

- HTTP defines a simple "request-response" conversation.
- A requesting (client) program establishes connection with a receiving (server) program. The client sends a request to the server. HTTP defines the rules to phrase the requests.
- The server replies to a request with a response. The rules of the response is also defined by HTTP.
- HTTP does not define how the connection is made or managed, nor how the information is actually transmitted. This is done by TCP/IP.
- HTTP does not know, what information is passed as content of its messages. This is done by HTML.



INTERNET
ENGINEERING

LECT-10, S-29
IN2004S_javed@kent.edu
Javed I. Khan@2004

Format of a Request

- Request=method+URI+version+(optionally) request modifier info.
- Format:

```
<method><URI><HTTP version><crLF>  
[<header> :<value>] <crLF>  
[<header> :<value>] <crLF>  
[<header> :<value>] <crLF>  
blank line <crLF>  
[entity body]
```

- Example:

```
GET /path/file.html HTTP/1.0  
Accept: text/html  
Accept: audio/x  
User-agent: NCSA Mosaic/2.5
```



request



INTERNET
ENGINEERING

LECT-10, S-30
IN2004S_javed@kent.edu
Javed I. Khan@2004

HTTP Methods at a Glance

method	1.0	1.1	Explanation
GET	Y	Y	Retrieve specified URL.
HEAD	Y	Y	Retrieve the header section of GET.
POST	Y	Y	Send this data to specified URL.
PUT	N	Y	Replace the old value or URL with the content.
PATCH	N	Y	Similar to PUT except the attached entity is the difference.
COPY	N	Y	Copy the URL resource to the specified location(s).
MOVE	N	Y	Move the URL resource to the specified location(s).
DELETE	N	Y	Delete the URL resource.
LINK	N	Y	Establish link between the URL and other resources.
UNLINK	N	Y	Detach link between the URL and other resources.
TRACE	N	Y	Echo back what you have received.
OPTION	N	Y	Request information about communication options.
WRAPPED	N	Y	Allow request(s) to be wrapped or encrypted.

INTERNET ENGINEERING

HTTP/1.0 has 3 methods, and HTTP/1.1 has 13.

LECT-10, S-31
IN20045_javed@kent.edu
Javed I. Khan@2004

HTTP Header Fields

- HTTP/1.0 has 17 header fields and HTTP/1.1 has over 41 header fields. These can be grouped into 4 major classes:
 - General Headers
 - Request Only Headers
 - Response Only headers
 - Entity Headers (describes the content and can be in request and response)

INTERNET ENGINEERING

LECT-10, S-32
IN20045_javed@kent.edu
Javed I. Khan@2004

HTTP General Headers

Header	1.0	1.1	Explanation
Cache Control	N	Y	Caching Directions.
Connection	N	Y	Connection Maintenance routine
Date	Y	Y	Date of time of message origination.
Forwarded	N	Y	Used by gateways to trace intermediate steps and avoid loops
Keep-Alive	N	Y	Diagnostic Information.
MIME-version	Y	Y	Contains the mime version used to encode the message.
Pragma	Y	Y	Contains implementation directives (such as no caching)
Upgrade	N	Y	Lists additional protocols a client supports and would like to use if server agrees.

INTERNET ENGINEERING

LECT-10, S-33
IN20045_javed@kent.edu
Javed I. Khan@2004

HTTP Request Headers

Header	1.0	1.1	Explanation
Accept	Y	Y	Lists acceptable MIME type/subtype contents.
Accept-Charset	Y	Y	Lists acceptable character sets.
Accept-encoding	Y	Y	Lists acceptable encoding such as compress or zip.
Accept-language	Y	Y	Lists acceptable languages.
Authorization	Y	Y	Passes user authentication and encryption scheme.
From	Y	Y	Contains users email address.
Host	N	Y	Contains name fo target host.
If-Modified-since	Y	Y	Contains data/time used by GET to conditionally download documents.
Proxy-authorization	N	Y	Allow client to present their identity to a proxy.
Refer	Y	Y	URL of document from this request originated.
Unless	N	Y	List header conditions which must be met before a method is applied to a resource.
User-Agent	Y	Y	Provides client browser information.

INTERNET ENGINEERING

LECT-10, S-34
IN20045_javed@kent.edu
Javed I. Khan@2004

Format of a Response

- Request=status+description of info+actual info
- Format:


```
<HTTP version><result code> [<explanation>]<crlf>
[<header> : <value>] <crlf>
[<header> : <value>] <crlf>
[<header> : <value>] <crlf>
blank line <crlf>
[entity body]
```
- Example:


```
HTTP/1.0 200 OK
Server: NCSA/1.3
Mime_version:1.0
Content_type: text/html
Content_length:2000
<HTML>
...
</HTML>
```

INTERNET ENGINEERING

LECT-10, S-35
IN20045_javed@kent.edu
Javed I. Khan@2004

HTTP Response Headers


Header	1.0	1.1	Explanation
Location	Y	Y	Returns the exact location of the resource.
Proxy-authorization	N	Y	Returns the encryption/authorization scheme used in this session.
Public	N	Y	Lists all the non-standard methods supported by the server.
Retry-after	N	Y	Indicates (in seconds) when to retry a service.
Server	Y	Y	Returns information about server software.
WWW-authenticate	Y	Y	Returns the encryption/authorization scheme the server wants to use for all its Web interaction.

INTERNET ENGINEERING

LECT-10, S-36
IN20045_javed@kent.edu
Javed I. Khan@2004

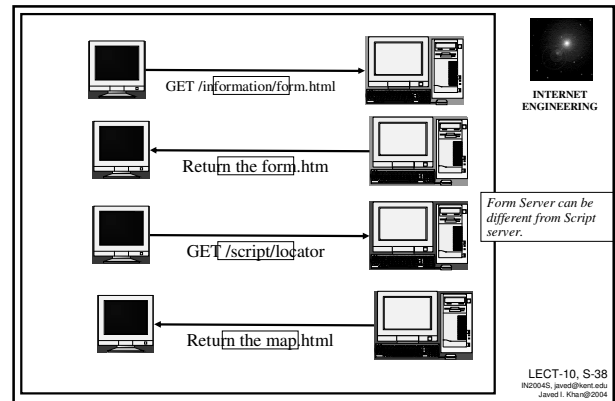
HTTP Entity Headers

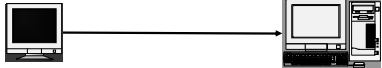

Header	1.0	1.1	Explanation
Allow	Y	Y	Lists methods supported by the URL resource.
Content-encoding	Y	Y	Specifies encoding of reply such as zip or compress.
Content-language	N	Y	Specifies natural language of reply.
Content-length	Y	Y	Length in bytes inside entity body.
Content-type	Y	Y	MIME content type of reply.
Content-version	N	Y	Contains a resource version number.
Derived-from	N	Y	Identifies the previous version of the resource.
Expires	Y	Y	Date/time after which the document is stale.
Last-modified	Y	Y	Last modification date of the content.
Link	N	Y	Contains document link information.
Title	N	Y	Contains the document title.
Transfer-encoding	N	Y	Identifies a transformation applied to the document or message body.
URL-Header	N	Y	Contains the resource name part of the URL.



INTERNET
ENGINEERING

LECT-10, S-37
IN2004S_jpv01@kent.edu
Javed I. Khan@2004




- 
- **Message Transmission**
 - Default: "keep connection open".
 - Header "connection: close" in a request means close the connection after servicing the request.
 - A set of requests can be pipelined. Client does not have to wait for previous responses.
 - For HTTP/1.0 default is not persistent.
 - The proxy server **MUST** signal persistent connections separately with its clients and the origin servers (or other proxy servers) that it connects to.
- 

INTERNET
ENGINEERING

LECT-10, S-39
IN2004S_jpv01@kent.edu
Javed I. Khan@2004

References

- **Principal References RFCs:**
 - URL: RFC 1728, 1808
 - MIME: RFC 1521
 - HTTP 1.0: RFC 1945
 - HTTP: 1.1 RFC 2068
- **Book: Web Server Technology, Yeager & McGrath.**



INTERNET
ENGINEERING

LECT-10, S-40
IN2004S_jpv01@kent.edu
Javed I. Khan@2004