

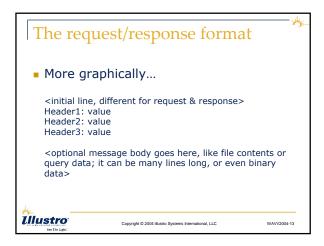


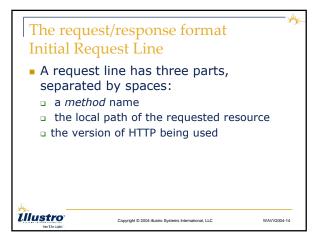
Multipurpose Internet Mail Extensions RFC 2045, November 1996 (part 1) Defines standards for text message formats and character sets Including many of the message headers used in HTTP messages HTTP messages are "MIME-like" but not completely MIME compliant

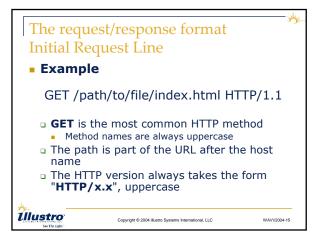
Structure of an HTTP Transaction • The client opens a connection and sends a request $message \ to \ an \ HTTP$ The server returns a server response message, usually containing The client receives the resource that was requested and process the received data Ullustro Copyright © 2004 illustro Systems International, LLC

The request/response format The format of the request and of the response are very similar Both messages are organized as First line Headers (optional) Empty line Body (optional)











The request/response format Initial Response Line Initial response line is called the Status line and also has three parts separated by spaces the HTTP version a response status code that gives the result of the request an English reason phrase describing the status code Typical status lines are: HTTP/1.1 200 OK HTTP/1.1 404 Not Found Copyright © 2004 illustric Systems International, LLC WANY2004-16

The request/response format Initial Response Line The status code is meant to be computer-readable It is a three-digit integer, and the first digit identifies the general category of response: Inx indicates an informational message only Xx indicates an informational message only Xx indicates success of some kind Xx redirects the client to another URL Xx indicates an error on the client's part Xx indicates an error on the server's part The most common status codes are: 200 OK 404 Not Found The reason phrase is meant to be human-readable, and may vary by server

The request/response format Header Lines Header lines provide information about the request or response, or about the object sent in the message body Header lines are in the text header format one line per header of the form "Header-Name: value" ending with CRLF

The request/response format The Message Body An HTTP message may have a body of data sent after the header lines In a request, this is where user-entered data or uploaded files are sent to the server In a response, this is where the requested resource is returned to the client

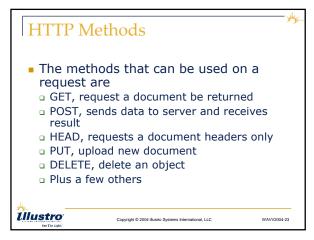
Copyright © 2004 illustro Systems International, LLC

The request/response format The Message Body If an HTTP message includes a body, there are usually header lines in the message that describe the body Content-Type: header gives MIME-type of the data in the body, such as text/html or image/gif Content-Length: header gives the number of bytes in the body

The request/response format Sample HTTP Exchange To retrieve the file at the URL http://www.somehost.com/path/file.html Send a request to www.somehost.com like: GET /path/file.html HTTP/1.1 User-Agent: HTTPTool/1.1 [empty line here]

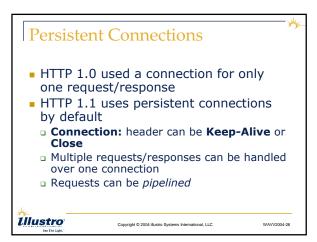
Illustro

The request/response format Sample HTTP Exchange The server should respond on the same connection with something like HTTP/1.1 200 OK Date: Mon. 05 Apr 2004 23:59:59 GMT Content-Type: text/html Content-Length: 1354 <html> <body> <h1>Web Page</h1> (more file contents) ... </body> </html> After sending the response, the server may close the connection

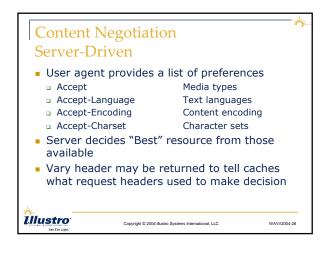




HTTP Methods POST Method ■ Different from a GET request in that... □ request URI is not a resource to retrieve; it's usually a program to handle the data you're sending □ HTTP response is normally program output, not a static file ■ The most common use of POST is to submit HTML form data to CGI scripts







Content Negotiation Agent-Driven Server returns status code 300 Multiple Choices and list of available resources User-agent decides which one is "Best" and issues new request for it Requires multiple trips to the server



Content Negotiation Transparent Combination of server-driven and agent-driven when the resource is cached When a cache contains the possible resources and is aware of the variances, it can provide the server-driven selection Offloads work from the server

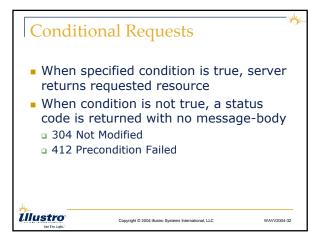
Copyright © 2004 illustro Systems International, LLC



lllustro

WAVV2004-30

Conditional Requests • Optional request headers can be included to make a request conditional • If-Modified-Since/If-Unmodified-Since • Compares date and time (Date: header) • If-Match/If-None-Match • Compares entity tags (ETAG: header)



Proxy Servers The HTTP protocol specification provides specific support for proxy servers Proxy servers are both a client and a server Accept requests from other clients Forward the request to a server (or another proxy) Cache the response Respond to the original client

Or, they respond immediately with a cached page instead of sending the request over the Internet Clients must be configured to use a proxy The client connects to the proxy instead of the host specified in the URL The request must specify the Absolute-URI instead of the Request-URI

Proxy Servers Headers specifically for proxies Proxy-Authorization: Proxy-Authenticate: Max-Forwards: Via:

Caching Caching of resources can be done by a cache server (proxy), the HTTP client, or both The goal of caching in HTTP is to: Eliminate the need to send requests Reduce the number of trips to the server Uses an expiration mechanism Eliminate the need to send full responses Reduce network bandwidth requirements Uses a validation mechanism



