

**NAME**

http – Send and process HTTP requests.

**DESCRIPTION**

The Hypertext Transfer Protocol (HTTP) (RFC 1945) is a client/server protocol used to retrieve documents in the World Wide Web. Documents are addressed using Uniform Resource Locators (URL) (RFC 1738). The **http** command allows to implement HTTP clients and a simple HTTP server.

**HTTP COMMAND**

The following list of http commands is useful for HTTP clients:

**http proxy** [*url*]

The **http proxy** command allows to define a proxy HTTP server which will be used in subsequent client commands. Using this command without an *url* argument will return the URL to the currently used proxy or an empty string. Providing a *url* argument sets the proxy server. Setting the proxy to an empty string turns the proxy feature off.

**http head** *url*

The **http head** command retrieves the HTTP header for the document located at *url*. The header is returned as a list of keys and values which can be converted into a Tcl array using the array set command.

**http get** *url fileName*

The **http get** command retrieves the document located at *url*. The body of the document is written to the file named *fileName*. The command returns the HTTP header as described for the **http head** command above.

**http post** *url docFileName fileName*

The **http post** command posts the document in *docFileName* to the location *url*. The body of the returned document is written to the file named *fileName*. The command returns the HTTP header as described for the **http head** command above.

**http put** *url docFileName*

The **http put** command puts the document in *docFileName* to the location *url*. The command returns the HTTP header as described for the **http head** command above.

**http delete** *url*

The **http delete** command deletes the document at the location defined by *url*. The command returns HTTP status information.

The following set of http commands is useful for a simple HTTP server:

**http server** [*port*]

The **http server** command starts a simple HTTP server listening on port *port*. If called without a *port* argument, the currently used port number or an empty string is returned. An empty string indicates that no server is currently active.

**http mime** *type extension*

The **http mime** command is used to add mime type definitions to the internal table which maps file name extension to mime types. If called without an argument, the list of all mime type mappings is returned. A new mapping is created by supplying a mime *type* and the file name *extension*.

**http bind** *pattern method [script]*

The **http bind** command binds a Tcl script to the invocation of a HTTP *method* where the URL matches *pattern*. Whenever a HTTP request of type *method* is received that matches the pattern, *script* is evaluated. The return value of the script must be a name of a file which will be send back

to the client. Errors during the evaluation of the *script* will result in an error response.

The following % sequences are substituted before a script bound to a URL is evaluated:

- %A** The network address of the client.
- %P** The URL path requested by the client.
- %S** The search path contained in the URL path.

**SEE ALSO**

scotty(1), Tnm(n), Tcl(n)

**AUTHORS**

Juergen Schoenwaelder <schoenw@cs.utwente.nl>