:mod:`http.server` --- HTTP servers

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 4)

Unknown directive type "module".

```
.. module:: http.server
   :synopsis: HTTP server and request handlers.
```

Source code: :source:`Lib/http/server.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 7); *backlink*

Unknown interpreted text role "source".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 9)

Unknown directive type "index".

```
.. index::
   pair: WWW; server
   pair: HTTP; protocol
   single: URL
   single: httpd
```

This module defines classes for implementing HTTP servers.

Warning

:mod:`http.server` is not recommended for production. It only implements basic security checks.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 22); *backlink*

Unknown interpreted text role "mod".

One class, :class:`HTTPServer`, is a :class:`socketserver.TCPServer` subclass. It creates and listens at the HTTP socket, dispatching the requests to a handler. Code to create and run the server looks like this:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 25); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 25); *backlink*

Unknown interpreted text role "class".

```
def run(server_class=HTTPServer, handler_class=BaseHTTPRequestHandler):
    server_address = ('', 8000)
    httpd = server_class(server_address, handler_class)
    httpd.serve_forever()
```

This class builds on the :class:`~socketserver.TCPServer` class by storing the server address as instance variables named :attr:`server_name` and :attr:`server_port`. The server is accessible by the handler, typically through the handler's :attr:`server` instance variable.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 37); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 37); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 37); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 37); *backlink*

Unknown interpreted text role "attr".

This class is identical to HTTPServer but uses threads to handle requests by using the :class:`~socketserver.ThreadingMixIn`. This is useful to handle web browsers pre-opening sockets, on which :class:`HTTPServer` would wait indefinitely.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 44); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 44); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 49)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7
```

The :class:`HTTPServer` and :class:`ThreadingHTTPServer` must be given a *RequestHandlerClass* on instantiation, of which this module provides three different variants:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 52); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 52); *backlink*

Unknown interpreted text role "class".

This class is used to handle the HTTP requests that arrive at the server. By itself, it cannot respond to any actual HTTP requests; it must be subclassed to handle each request method (e.g. GET or POST). :class:`BaseHTTPRequestHandler` provides a number of class and instance variables, and methods for use by subclasses.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 58); *backlink*

Unknown interpreted text role "class".

The handler will parse the request and the headers, then call a method specific to the request type. The method name is constructed from the request. For example, for the request method `SPAM`, the :meth:`do_SPAM` method will be called with no arguments. All of the relevant information is stored in instance variables of the handler. Subclasses should not need to override or extend the :meth:`__init__` method.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 64); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 64); *backlink*

Unknown interpreted text role "meth".

:class:`BaseHTTPRequestHandler` has the following instance variables:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 71); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 73)

Unknown directive type "attribute".

```
.. attribute:: client_address

   Contains a tuple of the form ``(host, port)`` referring to the client's
   address.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 78)

Unknown directive type "attribute".

```
.. attribute:: server

   Contains the server instance.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 82)

Unknown directive type "attribute".

```
.. attribute:: close_connection

   Boolean that should be set before :meth:`handle_one_request` returns,
   indicating if another request may be expected, or if the connection should
   be shut down.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 88)

Unknown directive type "attribute".

```
.. attribute:: requestline

   Contains the string representation of the HTTP request line. The
   terminating CRLF is stripped. This attribute should be set by
   :meth:`handle_one_request`. If no valid request line was processed, it
   should be set to the empty string.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 95)

Unknown directive type "attribute".

```
.. attribute:: command

   Contains the command (request type). For example, ``'GET'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 99)

Unknown directive type "attribute".

```
.. attribute:: path

   Contains the request path. If query component of the URL is present,
   then ``path`` includes the query. Using the terminology of :rfc:`3986`,
   ``path`` here includes ``hier-part`` and the ``query``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 105)

Unknown directive type "attribute".

```
.. attribute:: request_version

   Contains the version string from the request. For example, ``'HTTP/1.0'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 109)

Unknown directive type "attribute".

```
.. attribute:: headers

   Holds an instance of the class specified by the :attr:`MessageClass` class
   variable. This instance parses and manages the headers in the HTTP
   request. The :func:`~http.client.parse_headers` function from
   :mod:`http.client` is used to parse the headers and it requires that the
   HTTP request provide a valid :rfc:`2822` style header.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 117)

Unknown directive type "attribute".

```
.. attribute:: rfile

   An :class:`io.BufferedIOBase` input stream, ready to read from
   the start of the optional input data.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 122)

Unknown directive type "attribute".

```
.. attribute:: wfile

   Contains the output stream for writing a response back to the
   client. Proper adherence to the HTTP protocol must be used when writing to
   this stream in order to achieve successful interoperation with HTTP
   clients.

   .. versionchanged:: 3.6
      This is an :class:`io.BufferedIOBase` stream.
```

:class:`BaseHTTPRequestHandler` has the following attributes:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 132); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 134)

Unknown directive type "attribute".

```
.. attribute:: server_version

   Specifies the server software version.  You may want to override this. The
   format is multiple whitespace-separated strings, where each string is of
   the form name[/version]. For example, ``'BaseHTTP/0.2'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 140)

Unknown directive type "attribute".

```
.. attribute:: sys_version

   Contains the Python system version, in a form usable by the
   :attr:`version_string` method and the :attr:`server_version` class
   variable. For example, ``'Python/1.4'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 146)

Unknown directive type "attribute".

```
.. attribute:: error_message_format

   Specifies a format string that should be used by :meth:`send_error` method
   for building an error response to the client. The string is filled by
   default with variables from :attr:`responses` based on the status code
   that passed to :meth:`send_error`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 153)

Unknown directive type "attribute".

```
.. attribute:: error_content_type

   Specifies the Content-Type HTTP header of error responses sent to the
   client.  The default value is ``'text/html'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 158)

Unknown directive type "attribute".

```
.. attribute:: protocol_version

   This specifies the HTTP protocol version used in responses.  If set to
   ``'HTTP/1.1'``, the server will permit HTTP persistent connections;
   however, your server *must* then include an accurate ``Content-Length``
   header (using :meth:`send_header`) in all of its responses to clients.
   For backwards compatibility, the setting defaults to ``'HTTP/1.0'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 166)

Unknown directive type "attribute".

```
.. attribute:: MessageClass

   Specifies an :class:`email.message.Message`\ -like class to parse HTTP
   headers.  Typically, this is not overridden, and it defaults to
   :class:`http.client.HTTPMessage`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 172)

Unknown directive type "attribute".

```
.. attribute:: responses

   This attribute contains a mapping of error code integers to two-element tuples
   containing a short and long message. For example, ``{code: (shortmessage,
   longmessage)}``. The *shortmessage* is usually used as the *message* key in an
   error response, and *longmessage* as the *explain* key.  It is used by
   :meth:`send_response_only` and :meth:`send_error` methods.
```

A :class:`BaseHTTPRequestHandler` instance has the following methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 180); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 182)

Unknown directive type "method".

```
.. method:: handle()

   Calls :meth:`handle_one_request` once (or, if persistent connections are
   enabled, multiple times) to handle incoming HTTP requests. You should
   never need to override it; instead, implement appropriate :meth:`do_\*`
   methods.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 189)

Unknown directive type "method".

```
.. method:: handle_one_request()

   This method will parse and dispatch the request to the appropriate
   :meth:`do_\*` method.  You should never need to override it.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 194)

Unknown directive type "method".

```
.. method:: handle_expect_100()

   When a HTTP/1.1 compliant server receives an ``Expect: 100-continue``
   request header it responds back with a ``100 Continue`` followed by ``200
   OK`` headers.
   This method can be overridden to raise an error if the server does not
   want the client to continue.  For e.g. server can choose to send ``417
   Expectation Failed`` as a response header and ``return False``.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 205)

Unknown directive type "method".

```
.. method:: send_error(code, message=None, explain=None)

   Sends and logs a complete error reply to the client. The numeric *code*
   specifies the HTTP error code, with *message* as an optional, short, human
   readable description of the error.  The *explain* argument can be used to
   provide more detailed information about the error; it will be formatted
   using the :attr:`error_message_format` attribute and emitted, after
```

```
   a complete set of headers, as the response body.  The :attr:`responses`
   attribute holds the default values for *message* and *explain* that
   will be used if no value is provided; for unknown codes the default value
   for both is the string ``???``. The body will be empty if the method is
   HEAD or the response code is one of the following: ``1xx``,
   ``204 No Content``, ``205 Reset Content``, ``304 Not Modified``.

   .. versionchanged:: 3.4
      The error response includes a Content-Length header.
      Added the *explain* argument.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 223)

Unknown directive type "method".

```
.. method:: send_response(code, message=None)

   Adds a response header to the headers buffer and logs the accepted
   request. The HTTP response line is written to the internal buffer,
   followed by *Server* and *Date* headers. The values for these two headers
   are picked up from the :meth:`version_string` and
   :meth:`date_time_string` methods, respectively. If the server does not
   intend to send any other headers using the :meth:`send_header` method,
   then :meth:`send_response` should be followed by an :meth:`end_headers`
   call.

   .. versionchanged:: 3.3
      Headers are stored to an internal buffer and :meth:`end_headers`
      needs to be called explicitly.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 238)

Unknown directive type "method".

```
.. method:: send_header(keyword, value)

   Adds the HTTP header to an internal buffer which will be written to the
   output stream when either :meth:`end_headers` or :meth:`flush_headers` is
   invoked. *keyword* should specify the header keyword, with *value*
   specifying its value. Note that, after the send_header calls are done,
   :meth:`end_headers` MUST BE called in order to complete the operation.

   .. versionchanged:: 3.2
      Headers are stored in an internal buffer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 249)

Unknown directive type "method".

```
.. method:: send_response_only(code, message=None)

   Sends the response header only, used for the purposes when ``100
   Continue`` response is sent by the server to the client. The headers not
   buffered and sent directly the output stream.If the *message* is not
   specified, the HTTP message corresponding the response *code*  is sent.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 258)

Unknown directive type "method".

```
.. method:: end_headers()

   Adds a blank line
   (indicating the end of the HTTP headers in the response)
   to the headers buffer and calls :meth:`flush_headers()`.

   .. versionchanged:: 3.2
```

```
The buffered headers are written to the output stream.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 267)

Unknown directive type "method".

```
.. method:: flush_headers()

   Finally send the headers to the output stream and flush the internal
   headers buffer.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 274)

Unknown directive type "method".

```
.. method:: log_request(code='-', size='-')

   Logs an accepted (successful) request. *code* should specify the numeric
   HTTP code associated with the response. If a size of the response is
   available, then it should be passed as the *size* parameter.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 280)

Unknown directive type "method".

```
.. method:: log_error(...)

   Logs an error when a request cannot be fulfilled. By default, it passes
   the message to :meth:`log_message`, so it takes the same arguments
   (*format* and additional values).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 287)

Unknown directive type "method".

```
.. method:: log_message(format, ...)

   Logs an arbitrary message to ``sys.stderr``. This is typically overridden
   to create custom error logging mechanisms. The *format* argument is a
   standard printf-style format string, where the additional arguments to
   :meth:`log_message` are applied as inputs to the formatting. The client
   ip address and current date and time are prefixed to every message logged.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 295)

Unknown directive type "method".

```
.. method:: version_string()

   Returns the server software's version string. This is a combination of the
   :attr:`server_version` and :attr:`sys_version` attributes.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 300)

Unknown directive type "method".

```
.. method:: date_time_string(timestamp=None)

   Returns the date and time given by *timestamp* (which must be ``None`` or in
   the format returned by :func:`time.time`), formatted for a message
```

```
header. If *timestamp* is omitted, it uses the current date and time.

The result looks like ``'Sun, 06 Nov 1994 08:49:37 GMT'``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 308)

Unknown directive type "method".

```
.. method:: log_date_time_string()

   Returns the current date and time, formatted for logging.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 312)

Unknown directive type "method".

```
.. method:: address_string()

   Returns the client address.

   .. versionchanged:: 3.3
      Previously, a name lookup was performed. To avoid name resolution
      delays, it now always returns the IP address.
```

This class serves files from the directory *directory* and below, or the current directory if *directory* is not provided, directly mapping the directory structure to HTTP requests.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 327)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7
   The *directory* parameter.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 330)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.9
   The *directory* parameter accepts a :term:`path-like object`.
```

A lot of the work, such as parsing the request, is done by the base class :class:`BaseHTTPRequestHandler`. This class implements the :func:`do_GET` and :func:`do_HEAD` functions.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 333); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 333); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 333); *backlink*

Unknown interpreted text role "func".

The following are defined as class-level attributes of :class:`SimpleHTTPRequestHandler`:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 337); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 340)

Unknown directive type "attribute".

```
.. attribute:: server_version

   This will be ``"SimpleHTTP/" + __version__``, where ``__version__`` is
   defined at the module level.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 345)

Unknown directive type "attribute".

```
.. attribute:: extensions_map

   A dictionary mapping suffixes into MIME types, contains custom overrides
   for the default system mappings. The mapping is used case-insensitively,
   and so should contain only lower-cased keys.

   .. versionchanged:: 3.9
      This dictionary is no longer filled with the default system mappings,
      but only contains overrides.
```

The :class:`SimpleHTTPRequestHandler` class defines the following methods:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 355); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 357)

Unknown directive type "method".

```
.. method:: do_HEAD()

   This method serves the ``'HEAD'`` request type: it sends the headers it
   would send for the equivalent ``GET`` request. See the :meth:`do_GET`
   method for a more complete explanation of the possible headers.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 363)

Unknown directive type "method".

```
.. method:: do_GET()

   The request is mapped to a local file by interpreting the request as a
   path relative to the current working directory.

   If the request was mapped to a directory, the directory is checked for a
   file named ``index.html`` or ``index.htm`` (in that order). If found, the
   file's contents are returned; otherwise a directory listing is generated
   by calling the :meth:`list_directory` method. This method uses
   :func:`os.listdir` to scan the directory, and returns a ``404`` error
   response if the :func:`~os.listdir` fails.

   If the request was mapped to a file, it is opened. Any :exc:`OSError`
   exception in opening the requested file is mapped to a ``404``,
   ``'File not found'`` error. If there was a ``'If-Modified-Since'``
   header in the request, and the file was not modified after this time,
   a ``304``, ``'Not Modified'`` response is sent. Otherwise, the content
   type is guessed by calling the :meth:`guess_type` method, which in turn
   uses the *extensions_map* variable, and the file contents are returned.
```

```
        A ``'Content-type:'`` header with the guessed content type is output,
        followed by a ``'Content-Length:'`` header with the file's size and a
        ``'Last-Modified:'`` header with the file's modification time.

        Then follows a blank line signifying the end of the headers, and then the
        contents of the file are output. If the file's MIME type starts with
        ``text/`` the file is opened in text mode; otherwise binary mode is used.

        For example usage, see the implementation of the :func:`test` function
        invocation in the :mod:`http.server` module.

        .. versionchanged:: 3.7
           Support of the ``'If-Modified-Since'`` header.
```

The :class:`SimpleHTTPRequestHandler` class can be used in the following manner in order to create a very basic webserver serving files relative to the current directory:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 397); *backlink*

Unknown interpreted text role "class".

```
import http.server
import socketserver

PORT = 8000

Handler = http.server.SimpleHTTPRequestHandler

with socketserver.TCPServer(("", PORT), Handler) as httpd:
    print("serving at port", PORT)
    httpd.serve_forever()
```

:mod:`http.server` can also be invoked directly using the :option:`-m` switch of the interpreter. Similar to the previous example, this serves files relative to the current directory:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 414); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 414); *backlink*

Unknown interpreted text role "option".

```
python -m http.server
```

The server listens to port 8000 by default. The default can be overridden by passing the desired port number as an argument:

```
python -m http.server 9000
```

By default, the server binds itself to all interfaces. The option `-b/--bind` specifies a specific address to which it should bind. Both IPv4 and IPv6 addresses are supported. For example, the following command causes the server to bind to localhost only:

```
python -m http.server --bind 127.0.0.1
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 432)

Unknown directive type "versionadded".

```
.. versionadded:: 3.4
    ``--bind`` argument was introduced.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 435)

Unknown directive type "versionadded".

```
.. versionadded:: 3.8
    ``--bind`` argument enhanced to support IPv6
```

By default, the server uses the current directory. The option `-d/--directory` specifies a directory to which it should serve the files. For example, the following command uses a specific directory:

```
python -m http.server --directory /tmp/
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 444)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7
   ``--directory`` argument was introduced.
```

This class is used to serve either files or output of CGI scripts from the current directory and below. Note that mapping HTTP hierarchic structure to local directory structure is exactly as in :class:`SimpleHTTPRequestHandler`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 449); *backlink*

Unknown interpreted text role "class".

Note

CGI scripts run by the :class:`CGIHTTPRequestHandler` class cannot execute redirects (HTTP code 302), because code 200 (script output follows) is sent prior to execution of the CGI script. This pre-empts the status code.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 455); *backlink*

Unknown interpreted text role "class".

The class will however, run the CGI script, instead of serving it as a file, if it guesses it to be a CGI script. Only directory-based CGI are used --- the other common server configuration is to treat special extensions as denoting CGI scripts.

The :func:`do_GET` and :func:`do_HEAD` functions are modified to run CGI scripts and serve the output, instead of serving files, if the request leads to somewhere below the `cgi_directories` path.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 465); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 465); *backlink*

Unknown interpreted text role "func".

The :class:`CGIHTTPRequestHandler` defines the following data member:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 469); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 471)

Unknown directive type "attribute".

```
.. attribute:: cgi_directories

   This defaults to ``['/cgi-bin', '/htbin']`` and describes directories to
   treat as containing CGI scripts.
```

The :class:`CGIHTTPRequestHandler` defines the following method:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 476); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 478)

Unknown directive type "method".

```
.. method:: do_POST()

   This method serves the ``'POST'`` request type, only allowed for CGI
   scripts.  Error 501, "Can only POST to CGI scripts", is output when trying
   to POST to a non-CGI url.
```

Note that CGI scripts will be run with UID of user nobody, for security reasons. Problems with the CGI script will be translated to error 403.

:class:`CGIHTTPRequestHandler` can be enabled in the command line by passing the `--cgi` option:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]http.server.rst, line 487); *backlink*

Unknown interpreted text role "class".

```
python -m http.server --cgi
```