HTTP Status Codes

Keywords: httpd server status codes wof

The values of the numeric status code to HTTP requests are as follows. The data sections of messages Error, Forward and redirection responses may be used to contain human-readable diagnostic information.

Success 2xx

These codes indicate success. The body section if present is the object returned by the request. It is a MIME format object. It is in MIME format, and may only be in text/plain, text/html or one fo the formats specified as acceptable in the request.

OK 200

The request was fulfilled.

CREATED 201

Following a POST command, this indicates success, but the textual part of the response line indicates the URI by which the newly created document should be known.

Accepted 202

The request has been accepted for processing, but the processing has not been completed. The request may or may not eventually be acted upon, as it may be disallowed when processing actually takes place. there is no facility for status returns from asynchronous operations such as this.

Partial Information 203

When received in the response to a GET command, this indicates that the returned metainformation is not a definitive set of the object from a server with a copy of the object, but is from a private overlaid web. This may include annotation information about the object, for example.

No Response 204

Server has received the request but there is no information to send back, and the client should stay in the same document view. This is mainly to allow input for scripts without changing the document at the same time.

Error 4xx, 5xx

The 4xx codes are intended for cases in which the client seems to have erred, and the 5xx codes for the cases in which the server is aware that the server has erred. It is impossible to distinguish these cases in general, so the difference is only informational.

The body section may contain a document describing the error in human readable form. The document is in MIME format, and may only be in text/plain, text/html or one for the formats specified as acceptable in the request.

Bad request 400

The request had bad syntax or was inherently impossible to be satisfied.

Unauthorized 401

The parameter to this message gives a specification of authorization schemes which are acceptable. The client should retry the request with a suitable Authorization header.

PaymentRequired 402

The parameter to this message gives a specification of charging schemes acceptable. The client may retry the request with a suitable ChargeTo header.

Forbidden 403

The request is for something forbidden. Authorization will not help.

Not found 404

The server has not found anything matching the URI given

Internal Error 500

The server encountered an unexpected condition which prevented it from fulfilling the request.

Not implemented 501

The server does not support the facility required.

Service temporarily overloaded 502 (TO BE DISCUSSED)

The server cannot process the request due to a high load (whether HTTP servicing or other requests). The implication is that this is a temporary condition which maybe alleviated at other times.

Gateway timeout 503 (TO BE DISCUSSED)

This is equivalent to Internal Error 500, but in the case of a server which is in turn accessing some other service, this indicates that the respose from the other service did not return within a time that the gateway was prepared to wait. As from the point of view of the clientand the HTTP transaction the other service is hidden within the server, this maybe treated identically to Internal error 500, but has more diagnostic value.

Redirection 3xx

The codes in this section indicate action to be taken (normally automatically) by the client in order to fulfill the request.

Moved 301

The data requested has been assigned a new URI, the change is permanent. (N.B. this is an optimisation, which must, pragmatically, be included in this definition. Browsers with link editing capabiliy should automatically relink to the new reference, where possible)

The response contains one or more header lines of the form

URI: <url> String CrLf

Which specify alternative addresses for the object in question. The String is an optional comment field. If the response is to indicate a set of variants which each correspond to the requested URI, then the multipart/alternative wrapping may be used to distinguish different sets

Found 302

The data requested actually resides under a different URL, however, the redirection may be altered on occasion (when making links to these kinds of document, the browser should default to using the Udi of the redirection document, but have the option of linking to the final document) as for "Forward".

The response format is the same as for Moved.

Method 303

        Method: <method> <url>
        body-section

Note: This status code is to be specified in more detail. For the moment it is for

discussion only.

Like the found response, this suggests that the client go try another network address. In this case, a different method may be used too, rather than GET.

The body-section contains the parameters to be used for the method. This allows a document to be a pointer to a complex query operation.

The body may be preceded by the following additional fields _as listed_.

Not Modified 304

_If the client has done a conditional GET and access is allowed, but the document has not been modified since the date and time specified in _If-Modified-Since_ field, the server responds with a 304 status code and does not send the document body to the client.

Response headers are as if the client had sent a HEAD request, but limited to only those headers which make sense in this context. This means only headers that are relevant to cache managers and which may have changed independently of the document's Last-

Modified date. Examples include Date , Server and Expires .

The purpose of this feature is to allow efficient updates of local cache information (including relevant metainformation) without requiring the overhead of multiple HTTP requests (e.g. a HEAD followed by a GET) and minimizing the transmittal of information already known by the requesting client (usually a caching proxy).