*part of Hypertext Transfer Protocol -- HTTP/1.1*
*RFC 2616 Fielding, et al.*

# 6 Response

After receiving and interpreting a request message, a server responds with an HTTP response message.

```
       Response      = Status-Line               ; Section 6.1
                       *(( general-header        ; Section 4.5
                        | response-header        ; Section 6.2
                        | entity-header ) CRLF)  ; Section 7.1
                       CRLF
                       [ message-body ]          ; Section 7.2
```

## 6.1 Status-Line

The first line of a Response message is the Status-Line, consisting of the protocol version followed by a numeric status code and its associated textual phrase, with each element separated by SP characters. No CR or LF is allowed except in the final CRLF sequence.

```
       Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF
```

### 6.1.1 Status Code and Reason Phrase

The Status-Code element is a 3-digit integer result code of the attempt to understand and satisfy the request. These codes are fully defined in section 10. The Reason-Phrase is intended to give a short textual description of the Status-Code. The Status-Code is intended for use by automata and the Reason-Phrase is intended for the human user. The client is not required to examine or display the Reason- Phrase.

The first digit of the Status-Code defines the class of response. The last two digits do not have any categorization role. There are 5 values for the first digit:

```
      - 1xx: Informational - Request received, continuing process

      - 2xx: Success - The action was successfully received,
        understood, and accepted

      - 3xx: Redirection - Further action must be taken in order to
        complete the request

      - 4xx: Client Error - The request contains bad syntax or cannot
        be fulfilled

      - 5xx: Server Error - The server failed to fulfill an apparently
        valid request
```

The individual values of the numeric status codes defined for HTTP/1.1, and an example set of corresponding Reason-Phrase's, are presented below. The reason phrases listed here are only recommendations -- they MAY be replaced by local equivalents without affecting the protocol.

```
      Status-Code    =
            "100"  ; Section 10.1.1: Continue
          | "101"  ; Section 10.1.2: Switching Protocols
          | "200"  ; Section 10.2.1: OK
          | "201"  ; Section 10.2.2: Created
          | "202"  ; Section 10.2.3: Accepted
          | "203"  ; Section 10.2.4: Non-Authoritative Information
          | "204"  ; Section 10.2.5: No Content
          | "205"  ; Section 10.2.6: Reset Content
          | "206"  ; Section 10.2.7: Partial Content
          | "300"  ; Section 10.3.1: Multiple Choices
          | "301"  ; Section 10.3.2: Moved Permanently
          | "302"  ; Section 10.3.3: Found
          | "303"  ; Section 10.3.4: See Other
          | "304"  ; Section 10.3.5: Not Modified
          | "305"  ; Section 10.3.6: Use Proxy
          | "307"  ; Section 10.3.8: Temporary Redirect
          | "400"  ; Section 10.4.1: Bad Request
          | "401"  ; Section 10.4.2: Unauthorized
          | "402"  ; Section 10.4.3: Payment Required
          | "403"  ; Section 10.4.4: Forbidden
          | "404"  ; Section 10.4.5: Not Found
          | "405"  ; Section 10.4.6: Method Not Allowed
          | "406"  ; Section 10.4.7: Not Acceptable

          | "407"  ; Section 10.4.8: Proxy Authentication Required
          | "408"  ; Section 10.4.9: Request Time-out
          | "409"  ; Section 10.4.10: Conflict
          | "410"  ; Section 10.4.11: Gone
          | "411"  ; Section 10.4.12: Length Required
          | "412"  ; Section 10.4.13: Precondition Failed
          | "413"  ; Section 10.4.14: Request Entity Too Large
          | "414"  ; Section 10.4.15: Request-URI Too Large
          | "415"  ; Section 10.4.16: Unsupported Media Type
          | "416"  ; Section 10.4.17: Requested range not satisfiable
          | "417"  ; Section 10.4.18: Expectation Failed
          | "500"  ; Section 10.5.1: Internal Server Error
          | "501"  ; Section 10.5.2: Not Implemented
          | "502"  ; Section 10.5.3: Bad Gateway
          | "503"  ; Section 10.5.4: Service Unavailable
          | "504"  ; Section 10.5.5: Gateway Time-out
          | "505"  ; Section 10.5.6: HTTP Version not supported
          | extension-code

      extension-code = 3DIGIT
      Reason-Phrase  = *<TEXT, excluding CR, LF>
```

HTTP status codes are extensible. HTTP applications are not required to understand the meaning of all registered status codes, though such understanding is obviously desirable. However, applications MUST understand the class of any status code, as indicated by the first digit, and treat any unrecognized response as being equivalent to the x00 status code of that class, with the exception that an unrecognized response MUST NOT be cached. For example, if an unrecognized status code of 431 is received by the client, it can safely assume that there was something wrong with its request and treat the response as if it had received a 400 status code. In such cases, user agents SHOULD present to the user the entity returned with the response, since that entity is likely to include human- readable information which will explain the unusual status.

## 6.2 Response Header Fields

The response-header fields allow the server to pass additional information about the response which cannot be placed in the Status- Line. These header fields give information about the server and about further access to the resource identified by the Request-URI.

```
       response-header = Accept-Ranges           ; Section 14.5
                       | Age                     ; Section 14.6
                       | ETag                    ; Section 14.19
                       | Location                ; Section 14.30
                       | Proxy-Authenticate      ; Section 14.33

                       | Retry-After             ; Section 14.37
                       | Server                  ; Section 14.38
```

```
                                | Vary                    ; Section 14.44
                                | WWW-Authenticate        ; Section 14.47
```

Response-header field names can be extended reliably only in combination with a change in the protocol version. However, new or experimental header fields MAY be given the semantics of response- header fields if all parties in the communication recognize them to be response-header fields. Unrecognized header fields are treated as entity-header fields.