

# HTTP - Responses

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

- A Status-line
- Zero or more header (General|Response|Entity) fields followed by CRLF
- An empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields
- Optionally a message-body

The following sections explain each of the entities used in an HTTP response message.

## Message Status-Line

A Status-Line consists of the protocol version followed by a numeric status code and its associated textual phrase. The elements are separated by space SP characters.

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

## HTTP Version

A server supporting HTTP version 1.1 will return the following version information:

HTTP-Version = HTTP/1.1

Explore our [latest online courses](#) and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## Status Code

The Status-Code element is a 3-digit integer where first digit of the Status-Code defines the class of response and the last two digits do not have any categorization role. There are



5 values for the first digit:

S.N.	Code and Description
1	<b>1xx: Informational</b> It means the request was received and the process is continuing.
2	<b>2xx: Success</b> It means the action was successfully received, understood, and accepted.
3	<b>3xx: Redirection</b> It means further action must be taken in order to complete the request.
4	<b>4xx: Client Error</b> It means the request contains incorrect syntax or cannot be fulfilled.
5	<b>5xx: Server Error</b> It means the server failed to fulfill an apparently valid request.

HTTP status codes are extensible and HTTP applications are not required to understand the meaning of all registered status codes. A list of all the status codes has been given in a separate chapter for your reference.

## Response Header Fields

We will study General-header and Entity-header in a separate chapter when we will learn HTTP header fields. For now, let's check what Response header fields are.

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.

- Accept-Ranges
- Age
- ETag
- Location
- Proxy-Authenticate
- Retry-After
- Server
- Vary
- WWW-Authenticate



You can introduce your custom fields in case you are going to write your own custom Web Client and Server.

## Examples of Response Message

Now let's put it all together to form an HTTP response for a request to fetch the **hello.htm** page from the web server running on tutorialspoint.com

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
Content-Length: 88
Content-Type: text/html
Connection: Closed
```

```
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

The following example shows an HTTP response message displaying error condition when the web server could not find the requested page:

```
HTTP/1.1 404 Not Found
Date: Sun, 18 Oct 2012 10:36:20 GMT
Server: Apache/2.2.14 (Win32)
Content-Length: 230
Connection: Closed
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
  <title>404 Not Found</title>
</head>
<body>
  <h1>Not Found</h1>
  <p>The requested URL /t.html was not found on this server.</p>
```

```
</body>  
</html>
```

Following is an example of HTTP response message showing error condition when the web server encountered a wrong HTTP version in the given HTTP request:

```
HTTP/1.1 400 Bad Request  
Date: Sun, 18 Oct 2012 10:36:20 GMT  
Server: Apache/2.2.14 (Win32)  
Content-Length: 230  
Content-Type: text/html; charset=iso-8859-1  
Connection: Closed
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html>  
<head>  
  <title>400 Bad Request</title>  
</head>  
<body>  
  <h1>Bad Request</h1>  
  <p>Your browser sent a request that this server could not understand.</p>  
  <p>The request line contained invalid characters following the protocol string.</p>  
</body>  
</html>
```