**Request header field:**

1. **Accept:**

Specify media types, which are acceptable for the response.

\*/\*: all media types

Media\_type/\*: mall subtypes of the media

If no Accept header field is present, then it is assumed that the client accepts all media types.

If an Accept header field is present, and if the server cannot send a response which is acceptable according to the combined Accept field value, then the server SHOULD send a **406 (not acceptable) response**.

Ex-1:

Accept: audio/\*; q=0.2, audio/basic

SHOULD be interpreted as "I prefer audio/basic, but send me any audio type if it is the best available after an 80% mark-down in quality."

Ex-2:

Accept: text/plain; q=0.5, text/html,

text/x-dvi; q=0.8, text/x-c

Verbally, this would be interpreted as "text/html and text/x-c are the preferred media types, but if they do not exist, then send the text/x-dvi entity, and if that does not exist, send the text/plain entity."

Accept: Text/html, application/xhtml+xml; q=0.9, image/avif, image/webp

1. **Accept-Encoding**: gzip, deflate, br

The Accept-Encoding request-header field restricts the content-codings (section 3.5) that are acceptable in the response, content-coding such as gzip, is a compression Algorithms.

Accept-Encoding = "Accept-Encoding" ":"

1#( codings [ ";" "q" "=" qvalue ] )

codings = ( content-coding | "\*" )

Examples of its use are:

Accept-Encoding: compress, gzip

Accept-Encoding:

Accept-Encoding: \*

Accept-Encoding: compress;q=0.5, gzip;q=1.0

Accept-Encoding: gzip;q=1.0, identity; q=0.5, \*;q=0

A server tests whether a content-coding is acceptable, according to an Accept-Encoding field, using these rules:

1. If the content-coding is one of the content-codings listed in

the Accept-Encoding field, then it is acceptable, unless it is

accompanied by a qvalue of 0. (As defined in section [3.9](https://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.9), a

qvalue of 0 means "not acceptable.")

2. The special "\*" symbol in an Accept-Encoding field matches any

available content-coding not explicitly listed in the header

field.

3. **If multiple content-codings are acceptable, then the acceptable**

**content-coding with the highest non-zero qvalue is preferred**.

4. The "identity" content-coding is always acceptable, unless

specifically refused because the Accept-Encoding field includes

"identity;q=0", or because the field includes "\*;q=0" and does

not explicitly include the "identity" content-coding. If the

Accept-Encoding field-value is empty, then only the "identity"

encoding is acceptable.

If an Accept-Encoding field is present in a request, and if the server cannot send a response which is acceptable according to the Accept-Encoding header, then the server SHOULD send an **error response** with the **406 (Not Acceptable)** status code.

If no Accept-Encoding field is present in a request, the server MAY assume that the client will accept any content coding. In this case, if "identity" is one of the available content-codings, then the server SHOULD use the "identity" content-coding, unless it has additional information that a different content-coding is meaningful to the client.

**Response Headers**

1. **Age**

The Age response-header field conveys the sender's estimate of the

amount of time since the response (or its revalidation) was

generated at the origin server.

A cached response is "fresh" if

its age does not exceed its freshness lifetime. Age values are

calculated as specified in section [13.2.3](https://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html#sec13.2.3).

Age = "Age" ":" age-value

age-value = delta-seconds

**Age values are non-negative decimal integers, representing time in**

**seconds.**

### Allow

The Allow entity-header field lists the set of methods **supported**

**by the resource**. The purpose of this

field is strictly to inform the recipient of valid methods

associated with the resource. An Allow header field MUST be

present in a **405 (Method Not Allowed)** response.

Allow = "Allow" ":" #Method

Example of use:

Allow: GET, HEAD, PUT