# HTTP HEADERS

- Accept
- Accept-Encoding
- Authorization
- Accept-Language
- Content-Type
- Content-Location
- Content-Encoding
- Content-Length
- Content-Language
- Cache-Control

**srikanth tekmudi**
@srikanth tekmudi

Client and server can pass the extra bit of information with the request and response using HTTP headers.

The HTTP headers are divided into four categories:

1 Request headers: Client to Server

2 Response headers: Server to Client

3 Representation headers: Information about the body of the resource

4 Payload headers: Information about the payload data

**srikanth tekmudi**
@srikanth tekmudi

## 📌 Accept

`Accept` header is used when the client wants to inform the server about the type of data that the client can understand.

For example,
Accept: image/png

## 📌 Accept-Encoding

The `Accept-Encoding` header is usually attached with a request to the server, indicating which encoding method is understandable by the client.

For example,
Accept-Encoding: gzip


**srikanth tekmudi**
@srikanth tekmudi

## 📌 Authorization

As the header name suggests, the `Authorization` request header is used to pass the credentials so that the server can authenticate the client.

For example,

Authorization: Basic dgfhWUytzgdfhgSYG

## 📌 Accept-Language

The `Accept-Language` request header is used to describe which language is understood by the client.

For example,
Accept-Language: en-US


**srikanth tekmudi**
@srikanth tekmudi

## 📌 Content-Type

`Content-Type` representational header specifies the media type of the resource. This header indicates the content type of the returned data when the client didn't mention any content encoding.

For example,
Content-Type: text/html; charset=UTF-8

## 📌 Content-Location

`Content-Location` header indicates the alternate location for the response. It is used when an API can return data in different formats depending on the `Accept` header.

For example,
Content-Location: /examples/foo.json
Content-Location: /examples/foo.xml


**srikanth tekmudi**
@srikanth tekmudi

## 📌 Content-Encoding

The `Content-Encoding` header is used to compress the message data or payload.

For example,
Content-Encoding: gzip

## 📌 Content-Length

As the header name suggests, it indicates the size of the resource or message body in bytes. It can be used with both request and response.

For example,
Content-Length: 148


**srikanth tekmudi**
@srikanth tekmudi

## 📌 Content-Language

`Content-Language` indicates the human logical language of the response. This header is used to deliver resources in multiple languages so that users can choose their preferred language.

For example,
Content-Language: en-US

## 📌 Cache-Control

`Cache-Control` header is used as a caching mechanism for both the request and response.
There are several directives(values) that we can pass with this header.

For example,
Cache-Control: max-age


**srikanth tekmudi**
@srikanth tekmudi