

# Parameters in HTTP

This chapter lists some of the HTTP protocol parameters and their syntax by which they are used in communication.

This chapter lists some of the HTTP protocol parameters and their syntax by which they are used in communication. For example, date format, URL format, etc. This will help you in building your requirements and feedback messages while writing Client or Server programs. You will see the complete usefulness of these parameters in the next chapters while learning about the message structure for HTTP requests and responses.

## HTTP version

HTTP uses a numbering diagram . to indicate the version of the protocol. The version of an HTTP message is indicated by an HTTP-Version field in the first line. Here is the general syntax of determining the HTTP version number:

```
HTTP - Version = "HTTP" "/" 1 * DIGIT "." 1 * DIGIT
```

### For example

```
HTTP / 1.0 or HTTP / 1.1
```

## Uniform Resource Identifiers - Identifier for uniform resources

The URI is a form-sensitive, case-insensitive string in a simple way containing a name, location, . to identify a source, for example, a website, a web service, . A common syntax of URI is used for HTTP as follows:

```
URI = "http:" "://" host [ ":" port ] [ abs_path [ "?" query ] ]
```

Here, if the **port** is empty or not provided, port 80 is said to be HTTP and an empty **abs\_path** is equivalent to an **abs\_path** as "/". Other characters in the set set and **unsafe** are equivalent to their "%" HEX HEX "encoding.

### For example

Here are 3 examples of URIs that are equivalent:

```
http://abc.com:80/~smith/home.html
http://ABC.com/%7Esmith/home.html
http://ABC.com:/%7esmith/home.html
```

## The Date / Time format

All labels Date / Time HTTP **Must** be represented in Greenwich Mean Time (GMT), without exception. HTTP applications are allowed to use the following 3 representative labels Date / Time:

```
Sun, 06 Nov 1994 08:49:37 GMT; RFC 822, updated by RFC 1123
Sunday, 06-Nov-94 08:49:37 GMT; RFC 850, obsoleted by RFC 1036
Sun Nov 6 08:49:37 1994; ANSI C's asctime () format
```

## Character sets

We use character sets to determine the character settings that the client likes. Many character sets can be listed separately by commas. If a value is not specified, the default is US-ASCII.

### For example

Below are valid character sets:

```
US-ASCII
```

```
or
```

```
ISO-8859-1
```

```
or
```

```
ISO-8859-7
```

## Content encoding

A content encryption value indicates that an encryption algorithm has been used to encrypt the content before passing it to the network. Content encryption is first used to allow a document to be compressed or in addition to be transmitted without misidentification.

All content encoding values are case-insensitive. HTTP / 1.1 uses content encoding values in Accept-Encoding and Content-Encoding Header fields that we will look at in the next chapters.

### For example

Below are valid encryption schemes:

```
Accept-encoding: gzip
```

or

```
Accept-encoding: compress
```

or

```
Accept-encoding: deflate
```

## Multimedia types (media types)

HTTP uses Internet Media Types in **Content-Type** and **Accept** fields to provide open and expandable data. All media type values are registered with IANA (Internet Assigned Number Authority). The general syntax for specifying the media type is as follows:

```
media-type = type "/" subtype * (";" parameter)
```

Type, subtype, and parameter attributes are case-insensitive.

### For example

```
Accept: image / gif
```

## Language tags

HTTP uses language tags in the **Accept-Language** and **Content-Language** fields. A language tag consists of one or more sections: A primary language card and a series of sub-tags:

```
language-tag = primary-tag * ("-" subtag)
```

Spaces are not allowed in the card and all tags are case-insensitive.

### For example

Example tags include:

```
en, en-US, en-cockney, i-cherokee, x-pig-latin
```

The two primary-tag words are an abbreviation for the language in ISO-639 and the first two characters in the subtag are the country code.

### According to Tutorialspoint

Previous post: What is HTTP

Next post: Message in HTTP

You finished reading the article "**Parameters in HTTP**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.