

HTTP - PARAMETERS

http://www.tutorialspoint.com/http/http_parameters.htm

Copyright © tutorialspoint.com

This chapter is going to list down few of the important HTTP Protocol Parameters and their syntax the way they are used in the communication. For example, format for date, format of URL, etc. This will help you in constructing your request and response messages while writing HTTP client or server programs. You will see the complete usage of these parameters in subsequent chapters while learning the message structure for HTTP requests and responses.

HTTP Version

HTTP uses a **<major>.<minor>** numbering scheme to indicate versions 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 specifying HTTP version number:

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

Example

```
HTTP/1.0
```

```
or
```

```
HTTP/1.1
```

Uniform Resource Identifiers

Uniform Resource Identifiers *URI* are simply formatted, case-insensitive string containing name, location, etc. to identify a resource, for example, a website, a web service, etc. A general syntax of URI used for HTTP is as follows:

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

Here if the **port** is empty or not given, port 80 is assumed for HTTP and an empty **abs_path** is equivalent to an **abs_path** of **"/"**. The characters other than those in the **reserved** and **unsafe** sets are equivalent to their **"%" HEX HEX** encoding.

Example

The following three URIs are equivalent:

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

Date/Time Formats

All HTTP date/time stamps **MUST** be represented in Greenwich Mean Time *GMT*, without exception. HTTP applications are allowed to use any of the following three representations of date/time stamps:

```
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 specify the character sets that the client prefers. Multiple character sets can be listed separated by commas. If a value is not specified, the default is the US-ASCII.

Example

Following are the valid character sets:

```
US-ASCII  
or  
ISO-8859-1  
or  
ISO-8859-7
```

Content Encodings

A content encoding value indicates that an encoding algorithm has been used to encode the content before passing it over the network. Content coding are primarily used to allow a document to be compressed or otherwise usefully transformed without losing the identity.

All content-coding values are case-insensitive. HTTP/1.1 uses content-coding values in the Accept-Encoding and Content-Encoding header fields which we will see in the subsequent chapters.

Example

Following are the valid encoding schemes:

```
Accept-encoding: gzip  
or  
Accept-encoding: compress  
or  
Accept-encoding: deflate
```

Media Types

HTTP uses Internet Media Types in the **Content-Type** and **Accept** header fields in order to provide open and extensible data typing and type negotiation. All the Media-type values are registered with the Internet Assigned Number Authority *IANA*. The general syntax to specify media type is as follows:

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

The type, subtype, and parameter attribute names are case--insensitive.

Example

```
Accept: image/gif
```

Language Tags

HTTP uses language tags within the **Accept-Language** and **Content-Language** fields. A language tag is composed of one or more parts: a primary language tag and a possibly empty series of subtags:

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

White spaces are not allowed within the tag and all tags are case- insensitive.

Example

Example tags include:

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

where any two-letter primary-tag is an ISO-639 language abbreviation and any two-letter initial subtag is an ISO-3166 country code

Loading [MathJax]/jax/output/HTML-CSS/jax.js