

CHEAT SHEET # 1

STATUS CODES DEFINITIONS FOR BEGINNERS



When it comes to designing a website, we now all know that certain processes should be followed, which allows Google (the biggest and most successful search engine) to lend a hand.

After you’ve set up your new website, clearly identified your audience and created valuable content, it’s then time to “organize your sites structure”.

First, we must learn how Google understands our websites, which this course showed you today, in order to fully understand Google’s Search Console, AKA Google Webmaster Tools.

Once we have a basic understanding of how Google understands our websites, it is then time to use Google’s FREE resources:

[Webmaster Central Blog](#)

[Webmaster Help Forum](#)

[Google Webmasters Google+](#)

[Google Webmasters Twitter](#)

[Google Webmasters YouTube Channel](#)

[How Search Works](#)

HTTP STATUS CODES FOR BEGINNERS

ALL VALID HTTP 1.1 STATUS CODES SIMPLY EXPLAINED

HTTP, or Hypertext Transfer Protocol, is the method by which clients (so, this is you), and servers communicate. When someone clicks on a link, types in the url or fills out a form onsite, like a contact request for example; this is all directed at the server. Basically, you the clicker/browser/buyer and so forth, who visits a site and likes what they see, will be using it...obviously. But, you are not really interacting with the website itself, you’re requesting, buying, submitting and so on to the SERVER side of things.

HTTP response codes for dummies.

50X: We royally screwed up.

40X: You royally screwed up.

30X: ask that dude over there

20X: cool beans

FOR MORE TECHNICAL ADVANCED OR GEEK LEVEL 10 – YOU CAN CHECK OUT THE FOLLOWING URL:

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Informational

100 - Continue

A status code of 100 indicates that (usually the first) part of a request has been received without any problems, and that the rest of the request should now be sent.

101 - Switching Protocols

HTTP 1.1 is just one type of protocol for transferring data on the web, and a status code of 101 indicates that the server is changing to the protocol it defines in the "Upgrade" header it returns to the client. For example, when requesting a page, a browser might receive a status code of 101, followed by an "Upgrade" header showing that the server is changing to a different version of HTTP.

Successful

200 - OK

The 200 status code is by far the most common returned. It means, simply, that the request was received and understood and is being processed.

201 - Created

A 201 status code indicates that a request was successful and as a result, a resource has been created (for example a new page).

202 - Accepted

The status code 202 indicates that server has received and understood the request, and that it has been accepted for processing, although it may not be processed immediately.

203 - Non-Authoritative Information

A 203 status code means that the request was received and understood, and that information sent back about the response is from a third party, rather than the original server. This is virtually identical in meaning to a 200 status code.

204 - No Content

The 204 status code means that the request was received and understood, but that there is no need to send any data back.

205 - Reset Content

The 205 status code is a request from the server to the client to reset the document from which the original request was sent. For example, if a user fills out a form, and submits it, a status code of 205 means the server is asking the browser to clear the form.

206 - Partial Content

A status code of 206 is a response to a request for part of a document. This is used by advanced caching tools, when a user agent requests only a small part of a page, and just that section is returned.

Redirection

300 - Multiple Choices

The 300 status code indicates that a resource has moved. The response will also include a list of locations from which the user agent can select the most appropriate.

301 - Moved Permanently

A status code of 301 tells a client that the resource they asked for has permanently moved to a new location. The response should also include this location. It tells the client to use the new URL the next time it wants to fetch the same resource.

302 - Found

A status code of 302 tells a client that the resource they asked for has temporarily moved to a new location. The response should also include this location. It tells the client that it should carry on using the same URL to access this resource.

303 - See Other

A 303 status code indicates that the response to the request can be found at the specified URL, and should be retrieved from there. It does not mean that something has moved - it is simply specifying the address at which the response to the request can be found.

304 - Not Modified

The 304 status code is sent in response to a request (for a document) that asked for the document only if it was newer than the one the client already had. Normally, when a document is cached, the date it was cached is stored. The next time the document is viewed, the client asks the server if the document has changed. If not, the client just reloads the document from the cache.

305 - Use Proxy

A 305 status code tells the client that the requested resource has to be reached through a proxy, which will be specified in the response.

307 - Temporary Redirect

307 is the status code that is sent when a document is temporarily available at a different URL, which is also returned. There is very little difference between a 302 status code and a 307 status code. 307 was created as another, less ambiguous, version of the 302 status code.

Client Error

400 - Bad Request

A status code of 400 indicates that the server did not understand the request due to bad syntax.

401 - Unauthorized

A 401 status code indicates that before a resource can be accessed, the client must be authorized by the server.

402 - Payment Required

The 402 status code is not currently in use, being listed as "reserved for future use".

403 - Forbidden

A 403 status code indicates that the client cannot access the requested resource. That might mean that the wrong username and password were sent in the request, or that the permissions on the server do not allow what was being asked.

404 - Not Found

The best known of them all, the 404 status code indicates that the requested resource was not found at the URL given, and the server has no idea how long for.

405 - Method Not Allowed

A 405 status code is returned when the client has tried to use a request method that the server does not allow. Request methods that are allowed should be sent with the response (common request methods are POST and GET).

406 - Not Acceptable

The 406 status code means that, although the server understood and processed the request, the response is of a form the client cannot understand. A client sends, as part of a request, headers indicating what types of data it can use, and a 406 error is returned when the response is of a type not in that list.

407 - Proxy Authentication Required

The 407 status code is very similar to the 401 status code, and means that the client must be authorized by the proxy before the request can proceed.

408 - Request Timeout

A 408 status code means that the client did not produce a request quickly enough. A server is set to only wait a certain amount of time for responses from clients, and a 408 status code indicates that time has passed.

409 - Conflict

A 409 status code indicates that the server was unable to complete the request, often because a file would need to be edited, created or deleted, and that file cannot be edited, created or deleted.

410 - Gone

A 410 status code is the 404's lesser known cousin. It indicates that a resource has permanently gone (a 404 status code gives no indication if a resource has gone permanently or temporarily), and no new address is known for it.

411 - Length Required

The 411 status code occurs when a server refuses to process a request because a content length was not specified.

412 - Precondition Failed

A 412 status code indicates that one of the conditions the request was made under has failed.

413 - Request Entity Too Large

The 413 status code indicates that the request was larger than the server is able to handle, either due to physical constraints or to settings. Usually, this occurs when a file is sent using the POST method from a form, and the file is larger than the maximum size allowed in the server settings.

414 - Request-URI Too Long

The 414 status code indicates the URL requested by the client was longer than it can process.

415 - Unsupported Media Type

A 415 status code is returned by a server to indicate that part of the request was in an unsupported format.

416 - Requested Range Not Satisfied

A 416 status code indicates that the server was unable to fulfill the request. This may be, for example, because the client asked for the 800th-900th bytes of a document, but the document was only 200 bytes long.

417 - Expectation Failed

The 417 status code means that the server was unable to properly complete the request. One of the headers sent to the server, the "Expect" header, indicated an expectation the server could not meet.

Server Error

500 - Internal Server Error

A 500 status code (all too often seen by Perl programmers) indicates that the server encountered something it didn't expect and was unable to complete the request.

501 - Not Implemented

The 501 status code indicates that the server does not support all that is needed for the request to be completed.

502 - Bad Gateway

A 502 status code indicates that a server, while acting as a proxy, received a response from a server further upstream that it judged invalid.

503 - Service Unavailable

A 503 status code is most often seen on extremely busy servers, and it indicates that the server was unable to complete the request due to a server overload.

504 - Gateway Timeout

A 504 status code is returned when a server acting as a proxy has waited too long for a response from a server further upstream.

505 - HTTP Version Not Supported

A 505 status code is returned when the HTTP version indicated in the request is not supported. The response should indicate which HTTP versions are supported.