Learn Session Management using php

Topics : PHP
Written on December 31, 2020

Learn Session Management using php Tutorial
Below Session Configuration is used in php.ini.
session.save_handler
This parameter specifies the method used by PHP to store and retrieve session variables. The default value is files, to indicate the use of session files, as described in the previous sections. The other values that this parameter can have are: mm to store and retrieve variables from shared memory, and user to store and retrieve variables with user-defined handlers. In Appendix D we describe how to create user-defined handlers to store session variables in a MySQL database.

session.save_path:
This parameter specifies the directory in which session files are saved when the session.save_handler is set to files. The default value is /tmp. When implementing user-defined save_handler methods, the value of this parameter is passed as an argument to the function that opens a session.

session.use_cookies:
This parameter determines if PHP sets a cookie to hold the session ID. Setting this parameter to 0 stops PHP from setting cookies and may be considered for the reasons discussed in the previous section. The default value is 1, meaning that a cookie stores the session ID.

session.name:
This parameter controls the name of the cookie, GET attribute, or POST attribute that is used to hold the session ID. The default is PHPSESSID, and there is no reason to change this setting unless there is a name collision with another variable.

session.auto_start:
With the default value of 0 for this setting, PHP initializes a session only when a session call such as session_start( ) or session_register( ) is made. If this parameter is set to 1, sessions are automatically initialized if a session ID is found in the request. Allowing sessions to autostart adds unnecessary overhead if session values aren't required for all scripts.

session.cookie_lifetime :
This parameter holds the life of a session cookie in seconds and is used by PHP when setting the expiry date and time of a cookie. The default value of 0 sets up a session cookie that lasts only while the browser program is running. Setting this value to a number of seconds other than 0 sets up the cookie with an expiry date and time. The expiry date and time of the cookie is set as an absolute date and time, calculated by adding the cookie_lifetime value to the current date and time on the server machine.[2]

session.cookie_path:
This parameter sets the valid path for a cookie. The default value is /, which means that browsers include the session cookie in requests for resources in all paths for the cookie's domain. Setting this value to the path of the session-based scripts can reduce the number of requests that need to include the cookie. For example, setting the parameter to /winestore instructs the browser to include the session cookie only with requests that start with http://www.webdatabasebook.com/winestore/.

session.cookie_domain:
This parameter can override the domain for which the cookie is valid. The default is a blank string, meaning that the cookie is set with the domain of the machine running the web server, and the browser includes the cookie only in requests sent to that domain.

session.cookie_secure:
This parameter sets the secure flag of a cookie, which prevents a browser from sending the session cookie over nonencrypted connections. When this setting is 1, the browser sends the session cookie over a network connection that is protected using the Secure Sockets Layer, SSL. We discuss SSL in the next chapter and provide installation instructions in Appendix A. The default value of 0 allows a browser to send the session cookie over encrypted and nonencrypted services.

session.serialize_handler:
This parameter sets up the method by which variables are serialized, that is, how they are converted into a stream of bytes suitable for the chosen session store. The default value is php, which indicates use of the standard PHP serialization functions. An alternative is wddx, which uses the WDDX libraries that encode variables as XML.
session.gc_probability
This parameter determines the probability that the garbage collection process will be performed when a session is initialized. The default value of 1 sets a 1% chance of garbage collection. See the discussion in the previous section for a full explanation of garbage collection.

session.gc_maxlifetime :
This parameter sets the life of a session in number of seconds. The default value is 1440, or 24 minutes. Garbage collection destroys a session that has been inactive for this period. See the discussion in the previous section for a full explanation of garbage collection.
session.referer_check
This parameter can restrict the creation of sessions to requests that have the HTTP Referer: header field set. This is a useful feature if access to an application is allowed only by following a hypertext link from a particular page such as a welcome page. If the HTTP Referer header field doesn't match the value of this parameter, PHP creates a session, but the session is marked as invalid and unusable. The default value of a blank string applies no restriction.

session.entropy_file :
PHP generates the session IDs from a random number seeded by the system date and time. Because the algorithm is known--it can be looked up in the PHP source code--it makes guessing session IDs a little easier. If this parameter is set to the name of a file, the first n bytes from that file (where n is specified by the session.entropy_length parameter) make the ID less predictable. The default value is left blank, meaning the default seeding method is used. One alternative is to use /dev/urandom, a special Unix device that produces a pseudorandom number.

session.entropy_length :
This parameter is the number of bytes to use when generating a session ID from the file specified by session.entropy_file. The default value is 0, the required value when no entropy file is set.

session.cache_limiter:
This parameter controls how responses can be cached by the browser. The default is nocache, meaning that PHP sets up the HTTP response to avoid browser caching. PHP sets the HTTP/1.1-defined header field Cache-Control to no-cache, the HTTP/1.0 header field Pragma to no-cache, and--for good measure--the Expires header field to Thu, 19 Nov 1981 08:52:00 GMT. Applications that use sessions--and even stateless web database applications--can be adversely affected when browsers cache pages. The other values allowed, private and public, allow responses to be cached. The distinction between private and public is apparent when a proxy server caches responses. See Appendix B for more details about HTTP caching.

session.cache_expire :
This parameter is used when caching is allowed; it sets the expiry date and time of the response to be the current system time plus the parameter value in minutes. The default value is 180.

Output :