SkillNewton

PHP Errors and Fixes Cheat Sheet

Common PHP 8 error messages with their usual cause and fix: parse errors, warnings, fatal and type errors, PDO, sessions, headers and limits.

This cheat sheet matches the PHP 8 error messages beginners see most often with their usual cause and fix. Find the message by its opening words, then read the cause and the fix. Messages are grouped by type: parse errors, warnings and notices, fatal and type errors, database errors, sessions and headers, and configuration limits. The last section shows a safe error-reporting setup for development and production, so you always see the real message.

How to use it

1. **Make the error visible.** On your development machine, turn on `display_errors` and `error_reporting(E_ALL)` as shown in the last section. On a live site, read the error log instead.
2. **Read the file and line number.** Every message ends with a file path and line number. For parse errors, also check the line before, where the real mistake often is.
3. **Match the opening words.** Find the row that starts with the same words as your message. Names and numbers in your message (variables, classes, sizes) will differ.
4. **Fix the cause, not the symptom.** Avoid hiding errors with the @ operator or by switching reporting off. Fix the code, then check the log stays clean.

Parse and syntax errors

`Parse error: syntax error, unexpected end of file`	A {, bracket or quote was opened and never closed. Use your editor's bracket matching to find the unclosed block.
`Parse error: Unclosed '{' on line 12`	PHP 8 tells you where the unclosed brace started. Add the missing } at the end of that block.
`Parse error: syntax error, unexpected variable "$name"`	Usually a missing semicolon or operator on the line before. Check the end of the previous statement.
`Parse error: syntax error, unexpected token "}"`	An extra closing brace, or a missing semicolon just before it.
`Parse error: syntax error, unexpected token "else"`	An `if` block was closed early, or a semicolon was put after `if (...)`.
`Parse error: syntax error, unexpected identifier …`	A missing semicolon, comma or concatenation dot, or an unescaped quote that ended a string early.
`No syntax errors detected in index.php`	Output of `php -l index.php`, which checks a file for parse errors without running it. Run it after editing.

Warnings, notices and deprecations

`Warning: Undefined variable $total`	The variable was never set, is misspelt, or is outside the function scope. Initialise it (`$total = 0;`) or pass it in as a parameter.

`Warning: Undefined array key "email"`	The key does not exist, often a form field that was not submitted. Use `$_POST['email'] ?? ''` or `isset()` first.
`Notice: Undefined index: email`	PHP 7 wording of the undefined array key warning above. Same cause and fix.
`Warning: Undefined property: User::$phone`	Reading a property that was never declared or set. Declare it in the class or check for typos.
`Warning: Trying to access array offset on … null`	You used `$row['name']` but `$row` is null, often because a query returned no row. Check the result before using it.
`Warning: foreach() argument must be of type array|object, null given`	The value you loop over is not an array. Check where it was set, or default it with `$items ?? []`.
`Warning: include(config.php): Failed to open stream: No such file or directory`	The path is wrong relative to the current working folder. Use an absolute path: `require __DIR__ . '/config.php';`
`Deprecated: htmlspecialchars(): Passing null to parameter #1 ($string) of type string is deprecated`	PHP 8.1+ warns when null is passed to built-in string functions. Pass a string instead, for example `$value ?? ''`.
`Deprecated: Creation of dynamic property User::$age is deprecated`	PHP 8.2+: assigning a property that is not declared in the class. Declare the property.

Fatal errors and type errors

`Fatal error: Uncaught TypeError: add(): Argument #1 ($a) must be of type int, string given`	The function has a type declaration and received the wrong type. Cast or validate input first, for example `(int) $_GET['qty']`.
`Fatal error: Uncaught Error: Call to a member function fetch() on null`	The variable holding the object is null (or `on bool`: a function returned false). Check the earlier call that should have created it.
`Fatal error: Uncaught Error: Class "App\Models\User" not found`	Wrong namespace or missing `use` statement, file not loaded, or the file name case does not match on Linux. With Composer, run `composer dump-autoload`.
`Fatal error: Uncaught Error: Call to undefined function mysql_connect()`	The old mysql_* functions were removed in PHP 7. Rewrite the code with PDO or mysqli.
`Fatal error: Uncaught Error: Call to undefined method User::getName()`	Typo in the method name, or the method is defined in a different class.
`Fatal error: Uncaught ArgumentCountError: Too few arguments to function greet(), 0 passed`	The call is missing required arguments. Pass them, or give the parameter a default value.
`Fatal error: Uncaught Error: Undefined constant "DB_HOST"`	The constant was not defined before use (often a config file not included), or a string is missing its quotes.
`Fatal error: Uncaught DivisionByZeroError: Division by zero`	PHP 8 throws on division or modulo by zero. Check the divisor before dividing.
`Fatal error: Cannot redeclare … (previously declared in …)`	The same function or file was loaded twice. Use `require_once` and avoid duplicate function names.
`Fatal error: Uncaught TypeError: Unsupported operand types: array + int`	Arithmetic on an array. Use the right element, such as `$prices[0] + 1`, or `array_sum()`.

Database errors (PDO and mysqli)

`Uncaught PDOException: SQLSTATE[HY000] [2002] Connection refused`	MySQL is not running, or the host or port is wrong. Start the MySQL service and check the host and port (3306 by default).
`SQLSTATE[HY000] [2002] No such file or directory`	With host `localhost`, PDO uses a Unix socket that was not found. Try `127.0.0.1` or set the correct socket path.
`SQLSTATE[HY000] [1045] Access denied for user 'root'@'localhost'`	Wrong username or password, or the user lacks access from that host. Check credentials and grants in MySQL.
`SQLSTATE[HY000] [1049] Unknown database 'shop'`	The database name in the DSN does not exist. Create it or fix the spelling.
`SQLSTATE[42S02]: Base table or view not found: 1146 Table 'shop.users' doesn't exist`	Table name is wrong or migrations or the SQL import were not run. Table names are case-sensitive on Linux.
`SQLSTATE[42S22]: Column not found: 1054 Unknown column 'emial' in 'field list'`	Typo in a column name, or the column has not been added yet.
`SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax`	Invalid SQL: a missing comma, unquoted value or reserved word as a name. Print the query and test it in MySQL directly.
`SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry`	A value already exists in a UNIQUE or PRIMARY KEY column, such as an email address. Check first or show a friendly message.
`SQLSTATE[HY093]: Invalid parameter number`	The number or names of placeholders do not match the values passed to `execute()`.
`PDOException: could not find driver`	The pdo_mysql extension is not enabled. Enable it in php.ini and restart the web server or PHP-FPM.
`Fatal error: Uncaught mysqli_sql_exception: …`	Since PHP 8.1, mysqli throws exceptions by default. Catch the exception and read its message, which is the MySQL error.

```php
<?php
// A PDO connection that reports errors clearly
$pdo = new PDO(
    'mysql:host=127.0.0.1;port=3306;dbname=shop;charset=utf8mb4',
    'shop_user',
    'change-me',
    [
        PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION, // default since PHP 8.0
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ]
);

$stmt = $pdo->prepare('SELECT name FROM users WHERE email = ?');
$stmt->execute([$email]);
$user = $stmt->fetch();          // false when no row is found
if ($user === false) {
    // handle "not found" before using $user['name']
}
```

Sessions and headers

`Warning: Cannot modify header information - headers already sent by (output started at /var/www/site/config.php:1)`	Something was output before `header()`, `setcookie()` or `session_start()`: spaces before `<?php`, a UTF-8 BOM, an echo, or a blank line after a closing `?>`. The message names the file and line; omit closing tags in PHP-only files.
`Warning: session_start(): Session cannot be started after headers have already been sent`	Same cause as above. Call `session_start()` at the very top of the script, before any output.
`Notice: session_start(): Ignoring session_start() because a session is already active`	The session was started twice. Use `if (session_status() === PHP_SESSION_NONE) { session_start(); }`.
`Warning: session_start(): open(/var/lib/php/sessions/sess_…, O_RDWR) failed: Permission denied (13)`	PHP cannot write to the session folder. Fix the folder ownership or set `session.save_path` to a writable folder.
`Session values are empty on the next page (no error)`	`session_start()` is missing on that page, the cookie path or domain differs, or the cookie is marked secure on a plain HTTP site.
`Warning: Undefined array key "user_id" (in $_SESSION)`	The user is not logged in or the session expired. Check `isset($_SESSION['user_id'])` and redirect to the login page.

Configuration and limits

`Fatal error: Allowed memory size of 134217728 bytes exhausted (tried to allocate 20480 bytes)`	The script used more than `memory_limit` (128M by default). Process large data in chunks, fetch rows one at a time or use generators; raise the limit only if the job really needs it.
`Fatal error: Maximum execution time of 30 seconds exceeded`	The script ran longer than `max_execution_time` (30 seconds by default for web requests; no limit on the CLI). Look for infinite loops or slow API and database calls; move long jobs to a CLI script or queue.
`Warning: POST Content-Length of … bytes exceeds the limit of 8388608 bytes`	The request is bigger than `post_max_size` (8M by default), so `$_POST` and `$_FILES` are empty. Raise it above `upload_max_filesize`.
`$_FILES['photo']['error'] is 1 (UPLOAD_ERR_INI_SIZE)`	The file is larger than `upload_max_filesize` (2M by default). Raise it in php.ini and restart PHP.
`Fatal error: Uncaught Error: Call to undefined function curl_init()`	The cURL extension is not enabled. Enable `extension=curl` (or install php-curl on Linux) and check with `php -m`.
`Warning: Module "mysqli" is already loaded`	The extension is enabled twice, in php.ini and in a conf.d file. Remove one of the lines.
`Times are 5 hours 30 minutes out (no error)`	PHP is using UTC. Set `date.timezone = Asia/Kolkata` in php.ini or call `date_default_timezone_set('Asia/Kolkata')`.
`php --ini`	Shows which php.ini file is loaded. The CLI and the web server often use different files, so check `phpinfo()` for the web one.

Debugging setup: development and production

`error_reporting(E_ALL);`	Report every error, warning, notice and deprecation. Use E_ALL in every environment.

`ini_set('display_errors', '1');`	Show errors in the browser. Development only.
`display_errors = Off`	php.ini setting for live sites, so visitors never see paths or queries.
`log_errors = On`	Write errors to the log. Keep this on everywhere.
`error_log = /var/log/php/error.log`	Choose where PHP writes its log. The folder must be writable by PHP.
`error_log('Order total: ' . $total);`	Write your own debug message to the error log without showing it to visitors.
`var_dump($row);`	Print a value with its type, for example to see whether it is null, false or an empty string.
`php -l index.php`	Check a file for syntax errors from the command line.

```
; php.ini on your DEVELOPMENT machine
error_reporting = E_ALL
display_errors = On
display_startup_errors = On
log_errors = On

; php.ini on a PRODUCTION server
error_reporting = E_ALL
display_errors = Off
display_startup_errors = Off
log_errors = On
error_log = /var/log/php/error.log

; Then watch the log while you test:
;   tail -f /var/log/php/error.log
```

Online version, with the matching free guide: skillnewton.com/resources/cheat-sheets/php-errors-and-fixes/
Learn this hands-on: PHP Web Development: Foundation & Practical Implementation – skillnewton.com/courses/php-foundation-course/