nodiscard

ENTITY

This attribute can be used to indicate that the return value of a function or a method should not be discarded. If the return value is not used in any way, a warning will be emitted.

This is useful for functions where not checking the return value is likely to be a bug.

To intentionally discard the return value of such a function, use (void) cast to suppress the warning.

> Since attributes are designed to be backwards compatible, `#[\NoDiscard]` can be added to functions and methods even when PHP 8.4 or below are supported, it just won't do anything. On PHP 8.5 and above a warning will be emitted if the result is unused. To suppress the warning without using `(void)`, which is not supported before PHP 8.5, consider using a variable like `$_`.

> `#[\NoDiscard]` applies to the specific function or method declaration it is written on, and the warning is emitted based on the declaration that is actually called. As a result, adding `#[\NoDiscard]` to an interface method or an abstract method does not emit a warning, because the method that is invoked is the implementing or overriding method. Likewise, a method that overrides a `#[\NoDiscard]` method does not emit the warning unless it is itself marked with the attribute. In contrast, a method imported from a trait keeps the attribute, because the trait method is copied into the using class as if declared there.

ENTITY

#[\Attribute(\Attribute::TARGET_METHOD | \Attribute::TARGET_FUNCTION)]

final

NoDiscard

ENTITY

public

readonly

string

null

message

ENTITY

ENTITY

`message`
An optional message explaining why the return value should not be discarded.

ENTITY

Basic usage

```
<?php

/**
 * Processes all the given items and returns an array with the results of the
 * operation for each item. `null` indicates success and an Exception indicates
 * an error. The keys of the result array match the keys of the $items array.
 *
 * @param array<string> $items
 * @return array<null|Exception>
 */
#[\NoDiscard("as processing might fail for individual items")]
function bulk_process(array $items): array {
    $results = [];

    foreach ($items as $key => $item) {
        if (\random_int(0, 9999) < 9999) {
            // Pretend to do something useful with $item,
            // which will succeed in 99.99% of cases.
```

```
            echo "Processing {$item}", PHP_EOL;
            $error = null;
        } else {
            $error = new \Exception("Failed to process {$item}.");
        }

        $results[$key] = $error;
    }

    return $results;
}

bulk_process($items);

?>
```

ENTITY

```
Warning: The return value of function bulk_process() should either be used or intentionally
ignored by casting it as (void), as processing might fail for individual items
```

Intentionally discarding the return value

```
<?php

#[\NoDiscard]
function some_command(): int {
    return 1;
}

// Suppress the warning using (void) - PHP 8.5+
(void) some_command();

// For compatibility with PHP versions before 8.5, use a temporary variable
$_ = some_command();

?>
```

ENTITY

Attributes overview

ENTITY

□□□ #2
□ □□□ □□□ 2026-10-31 17:42:09 UTC
□ □□□ □□□ 2026-10-31 17:42:10 UTC