constants

ENTITY ENTITY

The following constants indicate the type of error returned by `json_last_error` or stored as the `code` of a `JsonException`.

`JSON_ERROR_NONE` (`int`)
No error has occurred.

`JSON_ERROR_DEPTH` (`int`)
The maximum stack depth has been exceeded.

`JSON_ERROR_STATE_MISMATCH` (`int`)
Occurs with underflow or with the modes mismatch.

`JSON_ERROR_CTRL_CHAR` (`int`)
Control character error, possibly incorrectly encoded.

`JSON_ERROR_SYNTAX` (`int`)
Syntax error.

`JSON_ERROR_UTF8` (`int`)
Malformed UTF-8 characters, possibly incorrectly encoded.

`JSON_ERROR_RECURSION` (`int`)
The object or array passed to `json_encode` include recursive references and cannot be encoded. If the `JSON_PARTIAL_OUTPUT_ON_ERROR` option was given, ENTITY will be encoded in the place of the recursive reference.

`JSON_ERROR_INF_OR_NAN` (`int`)
The value passed to `json_encode` includes either `NAN` or `INF`. If the `JSON_PARTIAL_OUTPUT_ON_ERROR` option was given, `0` will be encoded in the place of these special numbers.

`JSON_ERROR_UNSUPPORTED_TYPE` (`int`)
A value of an unsupported type was given to `json_encode`, such as a ENTITY. If the `JSON_PARTIAL_OUTPUT_ON_ERROR` option was given, ENTITY will be encoded in the place of the unsupported value.

`JSON_ERROR_INVALID_PROPERTY_NAME` (`int`)
A key starting with \u0000 character was in the string passed to `json_decode` when decoding a

JSON object into a PHP object.

`JSON_ERROR_UTF16` (`int`)
Single unpaired UTF-16 surrogate in unicode escape contained in the JSON string passed to `json_decode`.

`JSON_ERROR_NON_BACKED_ENUM` (`int`)
The value passed to `json_encode` includes a non-backed enum which cannot be serialized. Available as of PHP 8.1.0.

The following constants can be combined to form options for `json_decode`.

`JSON_BIGINT_AS_STRING` (`int`)
Decodes large integers as their original string value.

`JSON_OBJECT_AS_ARRAY` (`int`)
Decodes JSON objects as PHP array. This option can be added automatically by calling `json_decode` with the second parameter equal to ENTITY.

The following constants can be combined to form options for `json_encode`.

`JSON_HEX_TAG` (`int`)
All ENTITY and ENTITY are converted to \u003C and \u003E.

`JSON_HEX_AMP` (`int`)
All ENTITY are converted to \u0026.

`JSON_HEX_APOS` (`int`)
All ' are converted to \u0027.

`JSON_HEX_QUOT` (`int`)
All " are converted to \u0022.

`JSON_FORCE_OBJECT` (`int`)
Outputs an object rather than an array when a non-associative array is used. Especially useful when the recipient of the output is expecting an object and the array is empty.

`JSON_NUMERIC_CHECK` (`int`)
Encodes numeric strings as numbers.

`JSON_PRETTY_PRINT` (`int`)
Use whitespace in returned data to format it.

`JSON_UNESCAPED_SLASHES` (`int`)
Don't escape `/`.

`JSON_UNESCAPED_UNICODE` (`int`)
Encode multibyte Unicode characters literally (default is to escape as \uXXXX).

`JSON_PARTIAL_OUTPUT_ON_ERROR` (`int`)
Substitute some unencodable values instead of failing.

`JSON_PRESERVE_ZERO_FRACTION` (`int`)
Ensures that ENTITY values are always encoded as a float value.

`JSON_UNESCAPED_LINE_TERMINATORS` (`int`)
The line terminators are kept unescaped when `JSON_UNESCAPED_UNICODE` is supplied. It uses the same behaviour as it was before PHP 7.1 without this constant. Available as of PHP 7.1.0.

The following constants can be combined to form options for `json_decode` and `json_encode`.

`JSON_INVALID_UTF8_IGNORE` (`int`)
Ignore invalid UTF-8 characters. Available as of PHP 7.2.0.

`JSON_INVALID_UTF8_SUBSTITUTE` (`int`)
Convert invalid UTF-8 characters to \0xfffd (Unicode Character 'REPLACEMENT CHARACTER') Available as of PHP 7.2.0.

`JSON_THROW_ON_ERROR` (`int`)
Throws `JsonException` if an error occurs instead of setting the global error state that is retrieved with `json_last_error` and `json_last_error_msg`. `JSON_PARTIAL_OUTPUT_ON_ERROR` takes precedence over `JSON_THROW_ON_ERROR`. Available as of PHP 7.3.0.

□□□ #2
□ □□□ □□□ 2026-10-31 17:51:39 UTC
□ □□□ □□□ 2026-10-31 17:51:40 UTC