

Rules Reference

Rules Reference

Anonymous · 09/09/2026

Rules Reference

Cast Rules

Rule	Description	Example
boolean, bool	Convert value to boolean	'cast' => 'boolean'
date	Convert value to Carbon date	'cast' => 'date'
datetime	Convert value to Carbon datetime	'cast' => 'datetime'
float, decimal	Convert value to float	'cast' => 'float'
integer, int	Convert value to integer	'cast' => 'integer'
string, str	Convert value to string	'cast' => 'string'
timestamp	Convert value to Unix timestamp	'cast' => 'timestamp'

Transform Rules

Rule	Description	Example
base64_decode	Decode base64 string	'transform' => ['base64_decode']
base64_encode	Encode value to base64	'transform' => ['base64_encode']
json_decode	Decode JSON string	'transform' => ['json_decode'], 'transform' => ['json_decode:array']
json_encode	Encode value to JSON	'transform' => ['json_encode'], 'transform' => ['json_encode:pretty']
json_to_array	Convert JSON string to array	'transform' => ['json_to_array']
json_to_object	Convert JSON string to object	'transform' => ['json_to_object']
round	Round numeric value	'transform' => ['round:2']
lowercase, lower	Convert string to lowercase	'transform' => ['lowercase']

Rule	Description	Example
slug	Convert string to URL friendly format	'transform' => ['slug']
transliterate	Convert characters to ASCII	'transform' => ['transliterate']
uppercase, upper	Convert string to uppercase	'transform' => ['uppercase']

Sanitize Rules

Rule	Description	Example
addslashes	Add slashes before quotes	'sanitize' => ['addslashes']
htmleentities	Convert characters to HTML entities	'sanitize' => ['htmleentities']
htmlspecialchars	Convert special characters to HTML entities	'sanitize' => ['htmlspecialchars']
regex_keep	Keep only characters matching pattern	'sanitize' => ['regex_keep:/[0-9]/']
regex_remove	Remove characters matching pattern	'sanitize' => ['regex_remove:/[^a-z]/']
remove_chars	Remove specific characters	'sanitize' => ['remove_chars:@#\$',']
remove_non_printable	Remove non-printable characters	'sanitize' => ['remove_non_printable']
remove_prefix	Remove string prefix	'sanitize' => ['remove_prefix:http']
remove_suffix	Remove string suffix	'sanitize' => ['remove_suffix:.txt']
spaces	Normalize multiple spaces to single	'sanitize' => ['spaces']
strip_tags	Remove HTML and PHP tags	'sanitize' => ['strip_tags'], 'sanitize' => ['strip_tags:<p> ']
substring	Extract part of string	'sanitize' => ['substring:5'], 'sanitize' => ['substring:0,10']
trim	Remove whitespace from ends	'sanitize' => ['trim']

Validate Rules

Array Validation

Rule	Description	Example
<code>in, choices</code>	Check if value is in list	<code>'validate' => ['in:a,b,c']</code>
<code>max_items</code>	Check array maximum length	<code>'validate' => ['max_items:5']</code>
<code>min_items</code>	Check array minimum length	<code>'validate' => ['min_items:1']</code>
<code>notempty</code>	Check if array is not empty	<code>'validate' => ['notempty']</code>
<code>unique</code>	Check if array values are unique	<code>'validate' => ['unique']</code>

Date Validation

Rule	Description	Example
<code>after</code>	Check if date is after another	<code>'validate' => ['after:2024-01-01']</code>
<code>after_or_equal</code>	Check if date is after or equal	<code>'validate' => ['after_or_equal:2024-01-01']</code>
<code>before</code>	Check if date is before another	<code>'validate' => ['before:2024-12-31']</code>
<code>before_or_equal</code>	Check if date is before or equal	<code>'validate' => ['before_or_equal:2024-12-31']</code>
<code>between</code>	Check if date is between two dates	<code>'validate' => ['between:2024-01-01,2024-12-31']</code>
<code>date_equals</code>	Check if date equals another	<code>'validate' => ['date_equals:2024-01-01']</code>
<code>date_format</code>	Check if date matches format	<code>'validate' => ['date_format:Y-m-d']</code>
<code>weekday</code>	Check if date is weekday	<code>'validate' => ['weekday']</code>
<code>weekend</code>	Check if date is weekend	<code>'validate' => ['weekend']</code>

File Validation

Rule	Description	Example
<code>file</code>	Validate file with size limit	<code>'validate' => ['file:2M']</code>
<code>image</code>	Validate image file with size limit	<code>'validate' => ['image:2M']</code>
<code>mimetype</code>	Validate file mime type	<code>'validate' => ['mimetype:application/pdf,image/jpeg']</code>

Financial Validation

Rule	Description	Example
bic	Validate BIC/SWIFT code	'validate' => ['bic']
card_number	Validate credit card number	'validate' => ['card_number']
iban	Validate IBAN	'validate' => ['iban']

I18n Validation

Rule	Description	Example
country	Validate country code	'validate' => ['country']
currency	Validate currency code	'validate' => ['currency']
language	Validate language code	'validate' => ['language']
locale	Validate locale code	'validate' => ['locale']
timezone	Validate timezone identifier	'validate' => ['timezone']

ID Validation

Rule	Description	Example
uuid	Validate UUID string	'validate' => ['uuid']

Internet Validation

Rule	Description	Example
email	Validate email address	'validate' => ['email']
hostname	Validate hostname	'validate' => ['hostname']
ip	Validate IP address	'validate' => ['ip'], 'validate' => ['ip:v4'], 'validate' => ['ip:v6']
url	Validate URL	'validate' => ['url']

Numeric Validation

Rule	Description	Example
decimal, float, real	Validate decimal number	'validate' => ['decimal:2']
digits_range	Validate number of digits range	'validate' => ['digits_range:3,5']
digits	Validate exact number of digits	'validate' => ['digits:4']
gt	Greater than	'validate' => ['gt:10']

Rule	Description	Example
<code>gte, min</code>	Greater than or equal	<code>'validate' => ['gte:10']</code>
<code>integer, int</code>	Validate integer	<code>'validate' => ['integer']</code>
<code>lt</code>	Less than	<code>'validate' => ['lt:10']</code>
<code>lte, max</code>	Less than or equal	<code>'validate' => ['lte:10']</code>
<code>multiple_of</code>	Check if number is multiple	<code>'validate' => ['multiple_of:5']</code>
<code>numeric</code>	Validate numeric value	<code>'validate' => ['numeric']</code>
<code>range</code>	Validate number in range	<code>'validate' => ['range:1,100']</code>

String Validation

Rule	Description	Example
<code>alpha</code>	Only alphabetic characters	<code>'validate' => ['alpha']</code>
<code>alpha_dash</code>	Alphanumeric with dashes/underscores	<code>'validate' => ['alpha_dash']</code>
<code>alpha_num</code>	Only alphanumeric characters	<code>'validate' => ['alpha_num']</code>
<code>base64</code>	Validate base64 string	<code>'validate' => ['base64']</code>
<code>ends_with</code>	String ends with value	<code>'validate' => ['ends_with:Test']</code>
<code>json</code>	Validate JSON string	<code>'validate' => ['json']</code>
<code>length</code>	Exact string length	<code>'validate' => ['length:10']</code>
<code>max_length</code>	Maximum string length	<code>'validate' => ['max_length:255']</code>
<code>min_length</code>	Minimum string length	<code>'validate' => ['min_length:3']</code>
<code>not_regex</code>	Not match regular expression	<code>'validate' => ['not_regex:/[0-9]/']</code>
<code>no_whitespace</code>	No whitespace in string	<code>'validate' => ['no_whitespace']</code>
<code>regex</code>	Match regular expression	<code>'validate' => ['regex:/^[A-Z]+\$/']</code>
<code>required</code>	Value is required	<code>'validate' => ['required']</code>
<code>slug</code>	Validate URL friendly string	<code>'validate' => ['slug']</code>
<code>starts_with</code>	String starts with value	<code>'validate' => ['starts_with:Test']</code>

Last updated on 09/09/2026