
checksley Documentation

Release 0.6.0

Jesús Espino

December 06, 2016

1 Usage	1
2 Validators	3
2.1 Core validators	3
2.2 Extend validators	5
2.3 L10N validators	6
3 API	9

Usage

Validators

To use the validators you have to define in the fields a data attribute based on the validator name (Example: `data-required="True"`).

2.1 Core validators

`type()`

The value of the field must fit the data-type.

Arguments

- `data-type` (*string*) – Type of validator

`email`

The field value must be an email.

`url`

The field value must be an url.

`urlstrict`

The field value must be an url (must have the protocol defined).

`digits`

The field value must be a digit.

`dateIso`

The field value must be a date in iso format.

`alphanum`

The field value must be an alphanumeric value.

`phone`

The field value must be a US phone number.

`notnull()`

The field must have a value.

Arguments

- `data-notnull` (*boolean*) –

`notblank()`

The field must have a not blank (spaces, tabs...) value.

Arguments

- **data-notblank** (*boolean*) –

required ()

The field must have a value.

Arguments

- **data-required** (*boolean*) –

regexp ()

The field value must match the data-regexp.

Arguments

- **data-regexp** (*string*) – Regular expression to verify

min ()

The field value must be greater than data-min.

Arguments

- **data-min** (*number*) – Min value

max ()

The field value must be smaller than data-max.

Arguments

- **data-max** (*number*) – Max value

range ()

The field value must be between data-range[0] and data-range[1].

Arguments

- **data-range** (*list*) – List of [min, max]

minlength ()

The field value must be longer than data-minlength.

Arguments

- **data-minlength** (*integer*) – Min length

maxlength ()

The field value must be shorter than data-maxlength.

Arguments

- **data-maxlength** (*integer*) – Max length

rangelength ()

The field value must have a length between data-rangelength[0] and data-rangelength[1].

Arguments

- **data-rangelength** (*list*) – List of [minlength, maxlength]

mincheck ()

Synonym of *max* ()

maxcheck ()

Synonym of *min* ()

rangecheck ()

Synonym of *range* ()

equalto ()

2.2 Extend validators

minwords ()

The field value must have more than data-minwords words.

Arguments

- **data-minwords** (*integer*) – Min words

maxwords ()

The field value must have less than data-maxwords words.

Arguments

- **data-maxwords** (*integer*) – Max words

rangewords ()

The field value must have words between data-rangewords[0] and data-rangewords[1].

Arguments

- **data-rangewords** (*list*) – List of [minwords, maxwords]

greaterthanvalue ()

The field value must be greater than the value of data-greaterthan.

Arguments

- **data-greaterthan** (*selector*) – Min value.

lessthanvalue ()

The field value must be smaller than the value of data-lessthan.

Arguments

- **data-lessthan** (*selector*) – Max value.

beforedatevalue ()

The field value must date be a date before the date in data-beforedate.

Arguments

- **data-beforedate** (*selector*) – Max date.

afterdatevalue ()

The field value must date be a date after the date in data-afterdate.

Arguments

- **data-afterdate** (*selector*) – Min date.

greaterthan ()

The field value must be greater than the field selected by data-greaterthan.

Arguments

- **data-greaterthan** (*selector*) – A jquery selector of other field.

lessthan ()

The field value must be smaller than the field selected by data-lessthan.

Arguments

- **data-lessthan** (*selector*) – A jquery selector of other field.

beforedate ()

The field value must date be a date before the field selected by data-beforedate.

Arguments

- **data-beforedate** (*selector*) – A jquery selector of other field.

afterdate ()

The field value must date be a date after the field selected by data-afterdate.

Arguments

- **data-afterdate** (*selector*) – A jquery selector of other field.

inlist ()

The field value must be in the list of valid values.

Arguments

- **data-inlist** (*string*) – List of valid values
- **data-inlistDelimiter** (*string*) – Delimiter to split the data-inlist string in valid values.

luhn ()

The field value must pass the luhn algorithm (Validates credit card numbers, as well as some other kinds of account numbers).

Arguments

- **data-luhn** (*boolean*) –

americandate ()

The field value must be a valid american date.

Arguments

- **data-americandate** (*boolean*) –

2.3 L10N validators

2.3.1 ES

es_dni ()

The field value must be a valid Spanish DNI.

Arguments

- **data-es_dni** (*boolean*) –

es_cif ()

The field value must be a valid Spanish CIF.

Arguments

- **data-es_cif** (*boolean*) –

es_postalcode ()

The field value must be a valid Spanish postal code.

Arguments

- **data-es_postalcode** (*boolean*) –

es_ssn ()

The field value must be a valid Spanish social security number.

Arguments

- **data-es_ssn** (*boolean*) –

es_ccc ()

The field value must be a valid Spanish bank account number (Codigo Cuenta Cliente).

Arguments

- **data-es_ccc** (*boolean*) –

2.3.2 US

us_region ()

The field value must be a valid USA region.

Arguments

- **data-us_region** (*boolean*) –

us_postalcode ()

The field value must be a valid USA postal code.

Arguments

- **data-us_postalcode** (*boolean*) –

class Checksley (*jq*)

The main Checksley class.

Arguments

- **jq** (*object*) – JQuery or Zepto instance.

updateDefaults (*options*)

Update the default checksley configuration with the options object.

Arguments

- **options** (*object*) – Options object to overwrite the defaults

updateValidators (*validators*)

Updated (or add) checksley validators.

Params object validators Validators object to overwrite the checksley validators

updateMessages (*lang, messages*)

Updated (or add) checksley messages of a language.

Params string lang Language code (optional)

Params object messages Messages object to overwrite the language messages.

injectPlugin ()

Inject the plugin on the jQuery or Zepto.

setLang (*lang*)

Set the current language.

Arguments

- **lang** (*string*) – Language code

detectLang ()

Try to detect the language from the html.

getMessage (*key, lang*)

Get a message from a language.

Arguments

- **key** (*string*) – The message key
- **lang** (*string*) – The language code (optional)

class Form (*elm*, *options*={})

initialize ()

Initialize the form initializing the fields, binding the events and binding the data fields.

bindData ()

Set the data checksley attribute to the form storing the *Form* () object.

unbindData ()

Unset the data checksley attribute to the form storing the *Form* () object.

initializeFields ()

Initialize the Field objects for all the fields of the form.

setErrors ()

Add to html the errors of this form as custom error messages.

Arguments

- **errors** (*object*) – An object of key/value with field name as key and error message as value.

validate ()

Validate all the fields of the form.

Returns boolean

bindEvents ()

Connect validation to the form events.

unbindEvents ()

Disconnect validation of the form events.

removeErrors ()

Remove error messages.

destroy ()

Destroy the form.

reset ()

Reset the status of the form.

class Field (*elm*, *options*={})

bindData ()

Unset the data checksley-field attribute to the form storing the *Field* () object.

unbindData ()

Unset the data checksley-field attribute to the form storing the *Field* () object.

focus ()

Set the focus in the field.

eventValidate (*event*)

Handle events that trigger the validation.

Arguments

- **event** (*Event*) –

unbindEvents ()

Disconnect validation of the field events.

bindEvents ()

Connect validation to the field events.

errorClassTarget ()

TODO: Review this method Return the target of the error class.

resetHtml5Constraints ()

Reload the constraints of the field based on his html5 type attribute.

resetConstraints ()

Reload the constraints of the field based on his data attributes.

hasConstraints ()

Check if the field has any constraint/validation.

Returns boolean

validate (showErrors)

Apply general validators and call *applyValidators ()*.

Arguments

- **showErrors** (*boolean*) – Enable error showing on this validation.

applyValidators (showErrors)

Apply specific field validators.

Arguments

- **showErrors** (*boolean*) – Enable error showing on this validation.

handleClasses (valid)

Add/remove classes to the field based on valid param.

Arguments

- **valid** (*boolean*) – validation status

manageError (name, constraint)

Obtain and add to the html the error message for a validation and a constraint.

Arguments

- **name** (*string*) – Validator name.
- **constraint** (*object*) – Constraint object.

setErrors (messages)

Add to html the errors of this field as custom error messages.

Arguments

- **messages** (*object*) – A string or an array of string with the error messages.

makeErrorElement (constraintName, message)

Build a li element with the message as content, and with the classes checksley-<constraintName> and <constraintName>.

Arguments

- **constraintName** (*string*) – The constraint thats generate the error.
- **message** (*string*) – The error message.

Returns Element

addError (errorElement)

Arguments

- **errorElement** (*Element*) – The li element with the error to add.

Add the errorElement to the error container of the field.

reset ()

Reset the status of the field.

removeErrors ()

Remove the field errors.

getValue ()

Get the current value of the field.

Returns string

errorContainerId ()

Get the error container id.

Returns string

errorContainerClass ()

Get the error container class.

Returns string

getErrorContainer ()

Return the field error container (create one if not exists).

destroy ()

Destroy the field.

setForm (*form*)

Set the form of the field.

Arguments

- **form** (*Form*) –

class **FieldMultiple** (*elm, options*)

Subclass of Field.

getSiblings ()

Get the other fields in the multiframe group.

getValue ()

Get the value based on the multiple field type (radio or checkbox).

Returns string

unbindEvents ()

Disconnect validation of the field events.

bindEvents ()

Connect validation to the field events.

class **ComposedField** (*elm, options*)

Subclass of Field.

getComponents ()

Get a list of jQuery objects that compound the composed field based on the data-composed attribute value.

Returns [element]

getValue()

Get the value based on the composition of the `getComposed()` returned fields joined with the `data-composed-joiner` attribute.

Returns string

unbindEvents()

Disconnect validation of the field events.

bindEvents()

Connect validation to the field events.

- `genindex`

A

addError() (built-in function), 11
afterdate() (built-in function), 6
afterdatevalue() (built-in function), 5
alphanum (global variable or constant), 3
americandate() (built-in function), 6
applyValidators() (built-in function), 11

B

beforedate() (built-in function), 5
beforedatevalue() (built-in function), 5
bindData() (built-in function), 10
bindEvents() (built-in function), 10, 12, 13

C

Checksley() (class), 9
ComposedField() (class), 12

D

dateIso (global variable or constant), 3
destroy() (built-in function), 10, 12
detectLang() (built-in function), 9
digits (global variable or constant), 3

E

email (global variable or constant), 3
equalto() (built-in function), 4
errorClassTarget() (built-in function), 11
errorContainerClass() (built-in function), 12
errorContainerId() (built-in function), 12
es_ccc() (built-in function), 7
es_cif() (built-in function), 6
es_dni() (built-in function), 6
es_postalcode() (built-in function), 6
es_ssn() (built-in function), 6
eventValidate() (built-in function), 10

F

Field() (class), 10
FieldMultiple() (class), 12

focus() (built-in function), 10
Form() (class), 9

G

getComponents() (built-in function), 12
getErrorContainer() (built-in function), 12
getMessage() (built-in function), 9
getSiblings() (built-in function), 12
getValue() (built-in function), 12
greaterthan() (built-in function), 5
greaterthanvalue() (built-in function), 5

H

handleClasses() (built-in function), 11
hasConstraints() (built-in function), 11

I

initialize() (built-in function), 10
initializeFields() (built-in function), 10
injectPlugin() (built-in function), 9
inlist() (built-in function), 6

L

lessthan() (built-in function), 5
lessthanvalue() (built-in function), 5
luhn() (built-in function), 6

M

makeErrorElement() (built-in function), 11
manageError() (built-in function), 11
max() (built-in function), 4
maxcheck() (built-in function), 4
maxlength() (built-in function), 4
maxwords() (built-in function), 5
min() (built-in function), 4
mincheck() (built-in function), 4
minlength() (built-in function), 4
minwords() (built-in function), 5

N

notblank() (built-in function), 3

nonnull() (built-in function), 3

P

phone (global variable or constant), 3

R

range() (built-in function), 4
rangecheck() (built-in function), 4
rangelength() (built-in function), 4
rangewords() (built-in function), 5
regexp() (built-in function), 4
removeErrors() (built-in function), 10, 12
required() (built-in function), 4
reset() (built-in function), 10, 12
resetConstraints() (built-in function), 11
resetHtml5Constraints() (built-in function), 11

S

setErrors() (built-in function), 10, 11
setForm() (built-in function), 12
setLang() (built-in function), 9

T

type() (built-in function), 3

U

unbindData() (built-in function), 10
unbindEvents() (built-in function), 10, 12, 13
updateDefaults() (built-in function), 9
updateMessages() (built-in function), 9
updateValidators() (built-in function), 9
url (global variable or constant), 3
urlstrict (global variable or constant), 3
us_postalcode() (built-in function), 7
us_region() (built-in function), 7

V

validate() (built-in function), 10, 11